Contextualization, User Modeling and Personalization in the Social Web

From Social Tagging via Context to Cross-System User Modeling and Personalization

Von der Fakultät für Elektrotechnik und Informatik
der Gottfried Wilhelm Leibniz Universität Hannover
zur Erlangung des Grades
Doktor der Naturwissenschaften
Dr. rer. nat.
genehmigte Dissertation
von

Fabian Abel

geboren am 8. Juli 1980 in Hannover, Deutschland

2011

Kommission:

Referentin:	Prof. Dr. Nicola Henze
Korreferent:	Prof. Dr. Wolfgang Nejdl
Korreferentin:	Prof. Dr. Cristina Baroglio
Tag der Promotion:	20. April 2011

Abstract

Social Web stands for the culture of participation and collaboration on the Web. Structures emerge from social interactions: social tagging enables a community of users to assign freely chosen keywords to Web resources. The structure that evolves from social tagging is called folksonomy and recent research has shown that the exploitation of folksonomy structures is beneficial to information systems.

In this thesis we propose models that better capture usage context of social tagging and develop two folksonomy systems that allow for the deduction of contextual information from tagging activities. We introduce a suite of ranking algorithms that exploit contextual information embedded in folksonomy structures and prove that these context-sensitive ranking algorithms significantly improve search in Social Web systems. We setup a framework of user modeling and personalization methods for the Social Web and evaluate this framework in the scope of personalized search and social recommender systems. Extensive evaluation reveals that our context-based user modeling techniques have significant impact on the personalization quality and clearly improve regular user modeling approaches. Finally, we analyze the nature of user profiles distributed on the Social Web, implement a service that supports cross-system user modeling and investigate the impact of cross-system user modeling methods on personalization. In different experiments we prove that our cross-system user modeling strategies solve cold-start problems in social recommender systems and that intelligent re-use of external profile information improves the recommendation quality also beyond the cold-start.

Keywords: user modeling, personalization, social web

Abstract

Das Social Web beschreibt eine Kultur der Partizipation, in der Internetbenutzer durch ihre Beiträge selbst zu einem wichtigen Bestandteil des World Wide Web werden. Im Social Web entstehen Strukturen durch soziale Interaktionen. So werden beim *Social Tagging* Web Ressourcen von einer Gruppe von Benutzern gemeinsam beschlagwortet. Das Resultat dieses emergenten Prozesses sind sogenannte Folksonomien, die Benutzer, Web Ressourcen und Schlagwörter (Tags) miteinander in Relation setzen. Verwandte Arbeiten haben gezeigt, dass Folksonomien vorteilhaft in Informationssystemen genutzt werden können, um etwa Suche zu verbessern oder benutzerspezifische Empfehlungen zu generieren.

In dieser Arbeit werden Modelle und Methoden eingeführt, die den Kontext von Social Tagging besser erfassen. Diese Methoden werden in zwei Onlinesystemen demonstriert, die wir im Rahmen dieser Arbeit entwickelt haben. Ferner stellen wir eine Reihe von Ranking Algorithmen vor, die Kontextinformation dazu verwenden um Elemente entsprechend anwendungs- und benutzerspezifischen Relevanzkriterien zu ordnen. Unsere Experimente zeigen, dass diese kontextsensitiven Algorithmen Suche in Social Tagging Systemen signifikant verbessern. Zudem stellen wir Methoden zur kontextbasierten Benutzermodellierung vor und zeigen, dass unsere Methoden erfolgreich für die Personalisierung von Social Web Systemen eingesetzt werden können. Unsere kontextbasierten Ansätze führen im Vergleich zu herkömmlichen Benutzermodellierungsstrategien zu einer signifikanten Verbesserung von personalisierter Suche und Empfehlungsfunktionalität. Schließlich untersuchen wir wie Benutzermodellierung im Social Web über Systemgrenzen hinaus umgesetzt werden kann. Hierzu analysieren wir die Charakteristiken von Profildaten, die über verschiedene Social Web Systeme verteilt sind, implementieren ein Framework zur Unterstützung von systemübergreifender Benutzermodellierung und erforschen welchen Einfluss systemübergreifende Benutzermodellierung auf Personalisierung in Social Web Systemen hat. Unsere Ergebnisse beweisen, dass unsere Benutzermodellierungsstrategien Kaltstartprobleme in Systemen lösen, die an den Benutzer angepasste Empfehlungen bereitstellen wollen, und ferner Personalisierung über den Kaltstart hinaus signifikant verbessern.

Schlagworte: Benutzermodellierung, Personalisierung, Social Web

Foreword

In the last years I published the building blocks of this thesis in several workshops, conferences, journals and book chapters relevant to the research area of information systems. Here, I list the most important publications that directly contribute to my thesis.

Basic principles and models that build the basis for our algorithms are best described in the following publications.

- The Benefit of additional Semantics in Folksonomy Systems. By F. Abel. In *Proceedings of the 2nd PhD Workshop on Information and Knowledge Management (PIKM '08)*, ACM, 2008 [1].
- Social Semantic Web at work: annotating and grouping Social Media content. By F. Abel, N. Henze, and D. Krause. In S. H. Jose Cordeiro and J. Filipe, editors, *Web Information Systems and Technologies, Lecture Notes in Business Information Processing*, volume 18, Springer, 2009 [25].
- Semantic Enhancement of Social Tagging Systems. By F. Abel, N. Henze, D. Krause, and M. Kriesell. In Vladan Devedzic, Dragan Gasevic, editors, *Annals of Information Systems – Web 2.0 & Semantic Web*, volume 6, 2009 [28].
- Multi-faceted Tagging in TagMe!. By F. Abel, R. Kawase, D. Krause, and P. Siehndel. In *8th International Semantic Web Conference (ISWC '09)*, Springer, 2009 [35].

We implemented these principles and and approaches to user and context modeling in different systems. We developed GroupMe!, a social bookmarking system that enables users to visually organize their bookmarks in groups, and TagMe!, a tagging and exploration front-end for Flickr images. Further, we implemented the so-called Grapple User Modeling Framework (GUMF), which allows for user modeling across system boundaries, and the Mypes service, which is part of GUMF and provides functionality for aggregating and aligning user data distributed across the Social Web. These *tools* have, for example, been presented in the subsequent research articles.

- GroupMe! – Where Semantic Web meets Web 2.0. By F. Abel, M. Frank, N. Henze, D. Krause, D. Plappert, and P. Siehndel. In *6th International Semantic Web Conference (ISWC '07)*, Springer, 2007 [10].
- A Novel Approach to Social Tagging: GroupMe!. By F. Abel, N. Henze, and D. Krause. In *4th International Conference on Web Information Systems and*

Technologies (WEBIST), INSTICC Press, 2008 [22].

- GroupMe! - Where Information meets. By F. Abel, N. Henze, and D. Krause. In *Proceedings of the 17th International Conference on World Wide Web (WWW '08)*, ACM, 2008 [21].
- GroupMe! - Combining ideas of Wikis, Social Bookmarking, and Blogging. By F. Abel, M. Frank, N. Henze, D. Krause, and P. Siehndel. In *2nd International Conference on Weblogs and Social Media (ICWSM 2008)*, AAAI Press, 2008 [12].
- The Art of multi-faceted Tagging – interweaving spatial annotations, categories, meaningful URIs and tags. By F. Abel, R. Kawase, D. Krause, P. Siehndel, and N. Ullmann. In *6th International Conference on Web Information Systems and Technologies (WEBIST '10)*, INSTICC Press, 2010 [36].
- Mashing up user data in the Grapple User Modeling Framework. By F. Abel, D. Heckmann, E. Herder, J. Hidders, G.-J. Houben, D. Krause, E. Leonardi, and K. van der Slujis. In *Workshop on Adaptivity and User Modeling in Interactive Systems (ABIS '09)*, 2009 [14].

The systems and tools we implemented served as playground to experiment with the algorithms, which we outline in this thesis. For example, we introduce several algorithms that exploit contextual information embedded in social tagging structures and apply these algorithms for search and ranking in tagging systems. An overview of these algorithms and corresponding evaluations regarding *search and ranking in social tagging systems* is given in the following papers.

- On the effect of group structures on ranking strategies in folksonomies. By F. Abel, N. Henze, D. Krause, and M. Kriesell. In R. Baeza-Yates and I. King, editors, *Weaving Services and People on the World Wide Web*, Springer, 2009 [27].
- Ranking in Folksonomy Systems: can context help? By F. Abel, N. Henze, and D. Krause. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM '08)*, ACM, 2008 [23].
- Context-aware ranking algorithms in folksonomies. By F. Abel, N. Henze, and D. Krause. In *Proceedings of the Fifth International Conference on Web Information Systems and Technologies (WEBIST '09)*, INSTICC Press, 2009 [24].
- Optimizing search and ranking in folksonomy systems by exploiting context information. By F. Abel, N. Henze, and D. Krause. *Lecture Notes in Business Information Processing*, volume 45(2), Springer, 2010 [26].
- The impact of multifaceted tagging on learning tag relations and search. By F. Abel, N. Henze, R. Kawase, and D. Krause. In *Extended Semantic Web Conference (ESWC '10)*, Springer, 2010 [19].

We further apply the proposed context and user modeling strategies in combination

with our ranking algorithms to allow for personalization in Social Web systems. Therefore, we introduce and evaluate several methods that support *personalized search and recommender systems*.

- Context-based ranking in folksonomies. By F. Abel, M. Baldoni, C. Baroglio, N. Henze, D. Krause, and V. Patti. In *Proceedings of the 20th ACM Conference on Hypertext and Hypermedia (Hypertext '09)*, ACM, 2009 [4].
- Leveraging search and content exploration by exploiting context in folksonomy systems. By F. Abel, M. Baldoni, C. Baroglio, N. Henze, R. Kawase, D. Krause, and V. Patti. In *New Review of Hypermedia and Multimedia: Web Science*, Taylor & Francis, 2010 [4].
- Exploiting additional Context for Graph-based Tag Recommendations in Folksonomy Systems. By F. Abel, N. Henze, and D. Krause. In *International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT '08)*. ACM, 2008 [20].

As the principles and tools, which we developed as part of this thesis, also increase interoperability across systems, we investigate *cross-system user modeling strategies* in the Social Web.

- Interweaving public user profiles on the Web. By F. Abel, N. Henze, E. Herder, and D. Krause. In *Proceedings of 18th International Conference on User Modeling, Adaptation, and Personalization (UMAP '10)*, Springer, 2010 [17].
- Building Blocks for User Modeling with data from the Social Web. By F. Abel, N. Henze, E. Herder, G.-J. Houben, D. Krause, and E. Leonardi. In *International Workshop on Architectures and Building Blocks of Web-Based User-Adaptive Systems (WABBWUAS '10)*, CEUR, 2010 [16].
- Linkage, Aggregation, Alignment and Enrichment of public user Profiles with Mypes. By F. Abel, N. Henze, E. Herder, and D. Krause. In *International Conference on Semantic Systems (I-Semantics '10)*, ACM, 2010 [18].
- A framework for flexible user profile mashups. By F. Abel, D. Heckmann, E. Herder, J. Hidders, G.-J. Houben, D. Krause, E. Leonardi, and K. van der Slujis. In *International Workshop on Adaptation and Personalization for Web 2.0 at UMAP '09*, CEUR, 2009 [13].
- A flexible rule-based method for interlinking, integrating, and enriching user data. By E. Leonardi, F. Abel, D. Heckmann, E. Herder, J. Hidders, and G.-J. Houben. In *Proceedings of 10th International Conference on Web Engineering (ICWE '10)*, Springer, 2010 [152].

During my Ph.D. work I was also concerned with side topics and corresponding research that emerged from the core work on this thesis. For example, we integrated the tools and systems, which we developed in this thesis, also in other platforms to analyze their

impact on social sharing of learning resources [38, 37], organizing news media [143] as well as on collaborative search [33]. We experimented with rule-based approaches for recommender systems [6, 7] and personalized search, where we exploited preferences explicitly specified by the people [32, 135]. We worked on user modeling in the Semantic Web [29] and proposed vocabularies such as the *Grapple User Profile Format* (Grapple statements) [15]. Further, we developed an access control mechanism for RDF stores (*AC4RDF*) [8] for protecting sensitive user profile data and implemented a corresponding interface that allows for the specification of access control rules [9].

In the area of user modeling and personalization on the Social Web we furthermore established three international workshops where we discussed these topics with researchers from the intelligent user interfaces, Semantic Web and user modeling & personalization communities.

- Workshop on User Data Interoperability in the Social Web (UDISW '10) [2] co-located with International Conference on Intelligent User Interfaces (IUI '10), Hong Kong, China.
- Workshop on Linking of User Profiles and Applications in the Social Semantic Web (LUPAS '10) [30] co-located with Extended Semantic Web Conference (ESWC '10), Heraklion, Greece.
- Workshop on Architectures and Building Blocks of Web-Based User-Adaptive Systems (WABBWUAS '10) [31] co-located with International Conference on User Modeling, Adaptation and Personalization (UMAP '10), Hawaii, USA.

Systems and tools we developed are available online and can be used by researchers, application developers as well as by the general public.

GroupMe! The social tagging system GroupMe! enables users to create collections of bookmarks. GroupMe! also attracted attention by industry when it was presented at the world's largest computer exposition CeBIT 2008 in Hannover, Germany. Website: `http://groupme.org`

TagMe! The Flickr tagging and exploration front-end TagMe! introduces novel paradigms to social tagging such as "tagging of tag assignments". Website: `http://tagme.groupme.org`

GUMF We developed the Grapple User Modeling Framework (GUMF) so that application developers can immediately benefit from the context and user modeling approaches presented in this thesis. Website: `http://gumf.groupme.org`

Mypes Interlinkage, aggregation and semantic enrichment of user data distributed across Social Web systems like Flickr, Facebook, or Delicious is offered by the Mypes service. Website: `http://mypes.groupme.org`

Further, we designed *Radiotube.tv*, which connects Last.fm and YouTube to provide personalized music video recommendations and enables researchers to plug-in and evaluate

folksonomy-based user modeling and recommender strategies. The datasets produced in the above systems are made available to the research community via APIs and can be obtained upon request. Additional information on this Ph.D. thesis is available online at `http://fabianabel.de/phd/`.

Acknowledgements

First, I would like to thank Prof. Dr. Nicola Henze for convincing me to do a Ph.D. and for her brilliant support and guidance during the last years. As my mentor she taught me how to transfer ideas into proper research and how to drive research projects. She gave me many opportunities to further develop myself. She supported my teaching activities, student mentoring, proposal writing and gave me a lot of freedom to develop ideas.

I am glad that I had the privilege to do my Ph.D. at L3S Research Center and I would like to thank Prof. Dr. Wolfgang Nejdl for both supporting my Ph.D. work and establishing this unique, creative, international research environment, in which collaboration with great, talented colleagues became a wonderful experience that impacted both my professional as well as my personal life very positively. I enjoyed the seminars, research workshops, reading group discussions, colloquia, info lunches, research meetings, project meetings, coding events, Ph.D. meetings, etc. and I am proud that I was part of the L3S team.

I thank Prof. Dr. Cristina Baroglio for her continuous support and gorgeous collaboration during the last four years. The research visits in Turin have been very important for this thesis and thanks to the amazing hospitality of Cristina Baroglio, Matteo Baldoni and Viviana Patti these stays became an unforgettable experience.

I am happy that I had the opportunity to work together with such great colleagues at L3S (postdocs, administrative staff, technical staff, Ph.D. students, professors, student assistants, interns). In particular, I would like to thank Daniel Krause, with whom I shared the room in the last three years and eight months. Much of the work reported in this thesis was done in collaboration with Daniel. I am glad that we met each other and I am grateful that we are friends. I thank Eelco Herder for his mentoring and guidance in the Grapple project and for making this project as well as writing papers fun tasks to work on. His creative research ideas – like the idea of organizing a BBQ event based on wish lists from Amazon.com – inspired research presented in this thesis. Much of the Ph.D. work has been done in collaboration with Geert-Jan Houben, Erwin Leonardi and Jan Hidders, whom I thank for their excellent work and for inviting me to work together with them in Delft.

I thank Ricardo Kawase for his inspiring ideas, his hands-on mentality and for all work we did together in context of the TagMe! project. For their coding support, for example in the GroupMe! and TagMe! project, I thank Nicole Ullmann, Mischa Frank, Patrick

Siehndel and Philipp Bähre. I always enjoyed working together with you and I was impressed by your motivation and commitment to the projects we tackled together.

I thank Daniel Olmedilla, Juri Luca De Coi, Arne Kösling, and Philipp Kärger who particularly helped me at the beginning of my Ph.D. to dive into interesting research work. I thank Ernesto Diaz-Aviles, Marco Fisichella and Ralf Krestel for inspiring discussions and for sharing their research ideas with me. For the interesting and enjoyable work on LearnWeb2.0 I thank Ivana Marenzi and Sergej Zerr. I thank Tereza Iofciu, Kerstin Bischoff and Peter Fankhauser for involving me into their research activities on tag-based fingerprints. I also thank Dimitrios Skoutas and Christian Kohlschütter for collaboration and intensive discussions in the context of SYNC3. Further, I thank all L3S colleagues for being such a great team and making the last years such a great experience.

After school my plan was actually to study physics. I thank my grandparents for motivating me to study computer science. I thank my mother and father for their financial support during my studies as well as their mental support during my Ph.D. work. Last but not least I thank Lydia for taking part in my Ph.D. life.

Contents

1 Introduction

In March 1989 Tim Berners-Lee proposed the development of a global hypertext system to improve knowledge management at CERN, the European Organization for Nuclear Research [54]. While the proposal initially attracted little attention, it was approved in 1990 by CERN manager Mike Sendall so that Berners-Lee was allowed to start the development of the first visual browser for the World Wide Web [75]. Therewith the so-called *Memex* envisioned by Vannevar Bush in 1945 that allows for storage, indexing and retrieval of documents and enables people to make and follow links between documents [72] was no longer a rather conceptional idea but became tangible.

Nowadays the Web has more than 100 billion documents and more than one billion people are using the Web [116]. Information retrieval in such large scale information system is a non-trivial task [165]. Berners-Lee et al. therefore shape the vision of the Semantic Web, "in which information is given a well-defined, better enabling computers and people to work in cooperation" [60]. The Semantic Web is, from a pragmatic point of view, a framework of standards specified by the World Wide Web Consortium (W3C) that allows data to be shared and reused on the Web. However, with the advent of Web 2.0 users more and more participate in the evolution of the Web [175] and the understanding of social interactions on the Web becomes crucial for the design of future Web applications [116]. Hence, a paradigm shift from a rather machine-centered view of the Web towards a more user- and community-centered view is postulated by various researchers [44, 106, 116]. The term "Social Web" expresses this paradigm shift.

Social media systems like YouTube, Flickr, or Delicious, which enable people to publish and share videos, images and bookmarks respectively, as well as social networking services like Facebook or LinkedIn further promote the notion of the Social Web. These systems successfully harness social interactions and benefit from emerging structures on the Social Web. For example, social tagging allows people to organize Web resources with freely chosen terms rather than pre-defined taxonomies [102]. User-generated tagging structures, so-called *folksonomies* [161], evolve over time like *desire lines* [166] and allow for efficient retrieval of Web resources [51].

With the advent of social tagging, research on folksonomy systems started exploring the design of these systems [42, 103, 158], investigated search and ranking algorithms for folksonomies [51, 124], and developed recommender systems that support users in the tagging process [73, 128, 199]. Therefore, most research activities model a folksonomy essentially as a set of user-tag-resource triples, so-called tag assignments, which specify that a certain user assigned a specific tag to a given resource [169, 214]. An inherent

problem of these folksonomy models is that the semantics of tag assignments are not well-defined, for example, tags can be ambiguous or different tags might actually mean the same thing. Moreover, traditional folksonomy models [124] abstract from the usage context in which tagging activities have been performed. Hence, from a given tag assignment it is difficult to deduce the actual intention of the user, for example: was the tag assigned to facilitate future retrieval or does the tag rather express some opinion [61]?

In this thesis we investigate whether contextual information is beneficial for information retrieval in folksonomy systems. By *context* we mean (1) information that is attached to the tag assignments like URIs that specify the meaning of the tag assignment [178] and (2) information about the entities referenced by the tag assignment such as profile information about the user who performed the tag assignment [167]. We present approaches for inferring contextual information from user activities, introduce models for embedding context information into folksonomies and design algorithms that take advantage from these advanced models. We evaluate our algorithms with respect to non-personalized as well as personalized information retrieval tasks.

Personalization becomes more and more important as the amount of Web resources is continually growing which makes the retrieval of relevant information difficult [165]. Systems that aim for personalization require information about their users so that they can adapt their functionality to the specific requirements of a user [127]. In the Social Web and folksonomy systems particularly, tagging activities form a valuable source for deducing user interests [155]. Tag-based user profiles [97, 167] have already been applied to support social tagging itself by means of tag recommendations [73, 185, 148, 199]. However, many research questions regarding personalization in folksonomy systems have not been answered yet, for example: how can search and content exploration in folksonomy systems be personalized; which user modeling strategies are appropriate for specific personalization tasks and settings; and is contextual information attached to the tag assignments beneficial for personalization? Answers to these questions will be given in this thesis.

In order to provide personalized services to users, systems have to overcome the so-called *cold-start problem*. For example, it is difficult to provide personalized recommendations to a new user, who just registered and are thus rather unknown to the system yet [196]. With increasing interoperability between systems, the Social Web provides new possibilities to overcome such obstacles. Standardizations of APIs (e.g. OpenSocial [173]) and authentication and authorization protocols (e.g. OpenID [183], OAuth [111]), as well as by (Semantic) Web standards such as RDF [140] and specific vocabularies such as FOAF [67] or SIOC[64] facilitate the process of connecting distributed user profiles. Given these developments, it becomes crucial to investigate the nature of these distributed profiles, propose methods for modeling users across system boundaries and evaluate the benefits of linking user profiles in context of today's Social Web scenery. As part of this thesis we will thus investigate user modeling strategies that exploit profile information distributed across the Social Web and research the impact of these strategies on personalization and cold-start recommendations particularly.

In summary, this thesis contributes to research in the following areas.

Context Modeling in Folksonomy systems. We propose models that better capture usage context of social tagging and develop two tagging systems that allow for the deduction of contextual information from tagging activities.

Search and Ranking in Folksonomy systems. We introduce ranking algorithms that exploit contextual information embedded in folksonomy structures and prove their advantages for information retrieval in several experiments and different settings.

User Modeling and Personalization in Social Web systems. We setup a framework of user modeling and personalization techniques for Social Web systems and evaluate the benefits of this framework with respect to different personalization tasks.

Cross-system User Modeling in the Social Web. We analyze the nature of profiles distributed on the Social Web and evaluate the impact of cross-system user modeling methods on personalization.

A detailed overview on the main research questions answered in this thesis will be given in Section 2.3.

1.1 Structure and Methodology

The main contributions of this thesis are described in Chapters 3-6. Chapter 3 will introduce models as well as corresponding systems where we implemented these models. Algorithms that exploit context and user models will be evaluated with respect to information retrieval (Chapter 4), personalized information retrieval (Chapter 5) and cross-system personalization (Chapter 6). Each of these chapters will start with an introduction, which motivates the corresponding research questions by referring to related work, and will conclude with a summary of main findings and contributions.

Chapter 2 introduces the realm of information retrieval on the Social Web and folksonomy systems particularly. We summarize existing models and ranking algorithms such as FolkRank [124] or HITS [139] that are important for the understanding of our approaches. Further, we summarize related work on search, ranking, user modeling and personalization within the scope of Social Web and derive the main research questions that will be answered in this thesis (see Section 2.3).

In **Chapter 3** we propose strategies for deducing contextual information form social tagging processes. We introduce a generic context folksonomy model that integrates such information. Further, we describe two folksonomy systems we developed where we implement this model and demonstrate strategies for inferring the semantics of tagging: GroupMe! [10] is a social bookmarking system for organizing Web resources in collections and TagMe! [35] is a tagging and exploration interface for pictures. Both systems feature new approaches to social tagging. In Section 3.3 and Section 3.4 we outline these features

and present results from usage analyses.

Algorithms that exploit folksonomies as well as embedded context information are presented in **Chapter 4**. We enhance existing ranking algorithms such as FolkRank [124] so that they can exploit additional semantics provided by the context folksonomy model defined in Chapter 3 and present novel algorithms such as GRank [1] or SocialHITS [4]. Further, we evaluate the performance of these context-sensitive ranking algorithms with respect to search in folksonomy systems. We conduct experiments on different datasets and prove that the consideration of contextual information such as the usage context in which a tag assignment was performed or a URI that specifies the semantic meaning of a given tag assignment improve search and ranking performance significantly.

Chapter 5 provides detailed insights on personalization in folksonomy systems. We propose a set of user modeling strategies and methods that use these models in combination with the ranking algorithms introduced in Chapter 4 for personalization. Overall, these models and methods form a personalization framework for the Social Web which we apply in context of recommender systems and search personalization. Evaluations on different datasets show significant benefits of our framework and explain the performance of our strategies.

In **Chapter 6** we extend the personalization framework with user modeling strategies that profile users across system boundaries on the Social Web. We therefore present a service that features aggregation, linkage, alignment, and enrichment of distributed user profiles. A large-scale analysis explains the characteristics of profile data distributed on the Social Web and justifies our cross-system user modeling strategies. Finally, experiments on personalized tag and resource recommendations prove significant benefits of modeling users *in context* of their Social Web activities.

Chapter 7 concludes this thesis by summarizing our main findings and contributions and answering the research questions raised in Chapter 2. Further, we outline future work made possible by the findings of this thesis and discuss open research challenges.

2 Background: From Social Tagging to Personalization

In this chapter we introduce general background regarding Social Web, social tagging, tagging systems and (personalized) information retrieval in tagging systems. While this chapter gives rather a broad overview and details some selected models and algorithms that will be applied, extended and evaluated in the following chapters, specific information on related work is given in the corresponding sections of the subsequent chapters.

2.1 Introduction to the Social Web and Social Tagging

With the advent of Web 2.0, the role of users on the Web shifted more and more from consumers to contributors so that nowadays *users add value* to the Web [175]. Resource sharing systems such as YouTube, Flickr or Delicious enable casual end-users to easily publish videos, photos and photos respectively. Tagging has become a valuable feature for organizing such resources. In the following we confine the notion of the Social Web and discuss its relation with the Semantic Web before we sketch social tagging and folksonomies, which are structures that emerge from social tagging. Finally, we discuss folksonomy models and folksonomy-based user models, which we utilize and extend in this thesis.

2.1.1 Social Web and Semantic Web

Social and Semantic Web relate to complementary aspects of the Web. While the Social Web refers to the increased user participation on the Web, the Semantic Web initiative[1], which is lead by the World Wide Web Consortium (W3C), aims to provide a "framework that allows data to be shared and reused across application, enterprise, and community boundaries" [117]. In their well-known article, published in the *Scientific American* in 2001, Berners-Lee et al. define the role of the Semantic Web as follows [60].

> The Semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation.

[1] `http://www.w3.org/2001/sw/`

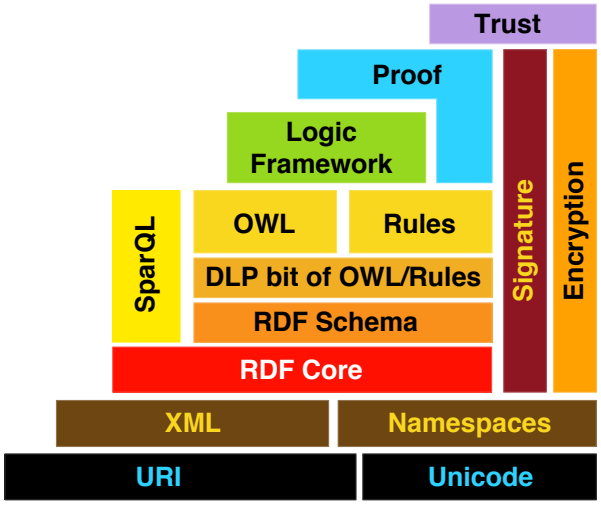
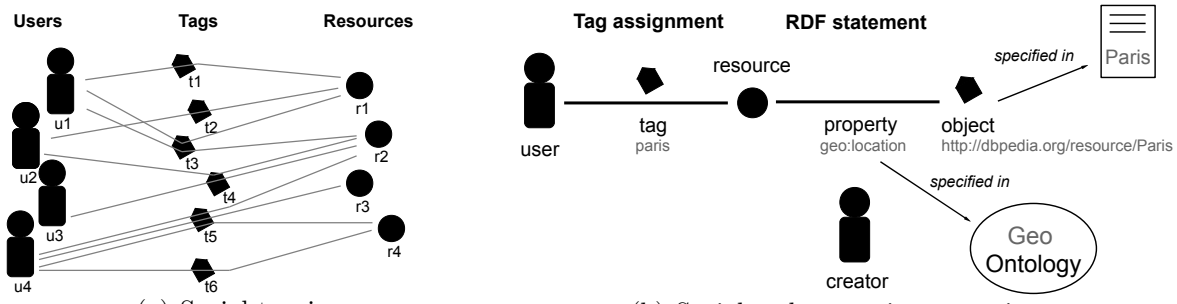


Figure 2.1: Semantic Web architecture as presented by Tim Berners-Lee in 2005 [55].

An important goal of the Semantic Web is to increase interoperability between Web systems by describing information on the Web in a semantically meaningful way. Therefore, the W3C Semantic Web activity defines a set of standards, which are arranged in a layered architecture [119] as displayed in Figure 2.1. Resources on the Semantic Web are identified via a Uniform Resource Identifier (URI) [59] and are described using the Resource Description Framework (RDF) [140]. RDF specifies the data model of the Semantic Web by means of subject-predicate-object triples, so-called RDF statements, which characterize some property (predicate) of a resource (subject) with some value (object). RDF descriptions, i.e. a set of RDF statements, can be serialized, for example, in RDF/XML [52] or Notation3 [58] syntax and can moreover be queried using an RDF query language such as SPARQL [180]—given that the RDF statements are stored in an RDF repository such as Sesame [70]. RDF Schema [65] as well as the Web Ontology Language (OWL) [87] allow for the specification of ontologies, which are according to Gruber "explicit, formal specifications of shared conceptualizations" [107].

The *Friend-Of-A-Friend* ontology (FOAF) [67], for example, allows for describing people and documents as well as relationships among them. By applying the *foaf:knows* property people can link to other people and thus explicitly specify their social network of people they know. FOAF descriptions and other RDF descriptions may be distributed across the Web and are possibly shared between Semantic Web applications so that appropriate trust mechanisms become important (cf. trust layer in Figure 2.1). For example, signatures in combination with encryption techniques can be applied to validate provenance of data and securely share RDF data between Semantic Web applications [203].

While the Semantic Web supports data sharing from a technical angle, Web 2.0 patterns endorse data sharing from a system's design point of view. O'Reilly advices developers of Web applications to re-use data produced in other applications and to foster user



(a) Social tagging (b) Social and semantic annotations

Figure 2.2: Tagging: (a) *social* tagging refers to situations, in which a group of users is annotating resources with tags, and (b) a comparison of tag assignments with RDF statements reveals that tag assignments lack well-defined semantics.

participation. Therefore he articulates, amongst others, the following Web 2.0 design patterns [175].

Cooperate, Don't Control. Web 2.0 applications are built of a network of cooperating data services. Therefore, offer Web services interfaces and content syndication, and re-use the data services of others [...].

Users Add Value. The key to competitive advantage in Internet applications is the extent to which users add their own data to that which you [the application developer] provide. Therefore, do not restrict your "architecture of participation" to software development. Involve your users both implicitly and explicitly in adding value to your application.

The Social Web reflects that more and more Web systems accomplish an *architecture of participation*, which involves participation of end-users. Resource sharing systems like Flickr or YouTube depend on their users, who contribute pictures and videos, because the main purpose of these systems relies in sharing user-contributed content. Social tagging supports resource sharing within these systems [121]: "social resource sharing systems are Web-based systems that allow users to upload their resources, and to label them with arbitrary words, so-called tags". For example, in Flickr a user may publishes pictures from her latest travel to France, which she annotates with keywords such as "france", "paris" or "beautiful-nature". These tags will help the user to retrieve certain images in the future and therewith support her personal information management [115]. Further, other users will be enabled to find the pictures if they utilize the corresponding tags to search for Flickr pictures [157, 153].

Social tagging describes a setting, in which a group of users is annotating a set of resources with tags (see Figure 2.2(a)). While there exists systems such as Google Mail, which exploit tagging for personal information management only, tagging becomes a *social* activity if a group of people is annotating a set of resources collaboratively [103]. Over time, structures emerge from social tagging. For example, the community of users

may agree on certain tags for describing specific (types of) resources as depicted in Figure 2.2(a): different users (u_2 and u_3) assigned tag t_4 to resource r_2, whereas t_3 was only applied by user u_1. However, the semantics of tags are not explicitly defined, for example, the semantic relation between the tags t_3 and t_4, which are both assigned to r_2, is not clear – even though both tags are syntactically different they could semantically have the same meaning.

In comparison with semantic annotations, which describe resources by means of RDF statements and thus adhere to Semantic Web standards, social annotations lack semantics regarding different dimensions as illustrated in Figure 2.2(b). Tags are assigned to a resource without specifying to which kind of property they refer to. A tag may describe the content of a resource, contextual information such as *when* or *where* the resource was created or it could express the user's opinion regarding the resource [61, 103]. RDF statements, by contrast, explicitly specify the property of the resource that is described by the object. The semantic meaning of such a property can moreover be explicitly defined within an ontology. Further, tags itself lack of well-defined semantics: tags are strings while the object of an RDF statement can be a typed literal or an RDF resource, which possibly itself has a semantic description that explains the meaning of the resource (see Figure 2.2(b)).

2.1.2 Emergence of Folksonomies from Social Tagging

The success of tagging can be explained by Ross Mayfield's *Power Law of Participation*[2]: tagging requires only low efforts from the users so that many users are motivated to contribute. Social tagging does not require pre-defined taxonomies, but vocabularies used for organizing resources in tagging systems rather emerge like *desire lines* [166]. The structures that emerge from social tagging are called folksonomies. The term *folksonomy* was first introduced by Thomas Vander Wal [161] and depicts the structures that evolve over time when users (the *folks*) annotate resources with freely chosen keywords.

Folksonomies relate users, tags and resources based on the tag assignments that are performed by the user community. As illustrated in Figure 2.2, tag assignments are triples that state which user assigned which tag to which resource. Hence, a folksonomy can thus be considered as a collection of tag assignments and *folksonomy systems* are those systems that allow for the evolution of folksonomies.

Today, there exist many diverse folksonomy systems in various domains. For example, Last.fm enables users to annotate music, bookmarks can be tagged in systems such as Delicious, BibSonomy supports social tagging of research articles, Amazon enables their customers to tag products, and Google Mail users can organize their emails via freely chosen labels. Marlow et al. developed a *tagging system design taxonomy* that allows for the classification of folksonomy systems [158]. In particular, the authors propose the following dimensions.

[2] `http://ross.typepad.com/blog/2006/04/power_law_of_pa.html`

Tagging support. When users annotate resources some systems support them with tag suggestions. For example, Delicious recommends tags to the user that are possibly appropriate for the given bookmark while in the so-called *ESP game* [209] users have to agree on adequate tags without the support of tag recommendations and moreover without the ability to view tags that are already assigned to the given resource (cf. *blind* vs. *viewable* tagging).

Aggregation model. The aggregation model describes whether (different) users are allowed to assign the same tag more than once to a particular resource. For example, Flickr does not allow for duplicated tags (*set*) whereas in Delicious the same tag can be attached multiple times to the same resource by different users (*bag*).

Object type. Marlow et al. distinguish between two main types of objects: textual and non-textual. The type of resources shared in today's social tagging systems ranges from traditional Web pages (bookmarks) to entities such as persons or events (cf. tagging in LinkedIn). An important characteristic of a tagging system is the system's approach for representing the resource during the tagging process. For example, it is important whether a picture is represented just via some textual description (e.g. tagging images in Delicious) or via some none-textual representation so that the tagger can actually see the content of the image (e.g. tagging images in Flickr).

Source of material. The source of the resources that are tagged by the users also differs between the systems. In traditional resource sharing systems such as YouTube or Flickr, resources are contributed by the users of the system (*user-contributed*). In social platforms such as Last.fm or the ESP game on the contrary, the system itself contributes the resources while the user masses are *just* utilized to structure these resources (*system*) and social bookmarking services like Delicious or StumbleUpon enable users to tag any resource available on the Web (*global*).

Tagging rights. Tagging rights prescribe *who* is allowed to annotate resources. Usually these rights are influenced by the *source of material* as well. For example, in Flickr users upload their (personal) pictures and can decide by themselves whether other users (e.g., *friends* or *all other users*) are allowed to tag these pictures (*permission-based*). By contrast, in Delicious all users are allowed to tag all resources (*free-for-all*) and in GMail users are only allowed to annotate their own resources (*self-tagging*).

Social connectivity. Some resource sharing systems enable users to connect with other users by means of *friend connections* (*connected*) or *groups* the users can join. Cha et al. showed that this social connectivity supports information propagation [80] and can thus foster the convergence of a folksonomy.

Resource connectivity. By nature, folksonomy systems connect resources via tags as well as via the users who assign tags to the resources. In addition, some tagging systems provide functionality to connect resources explicitly: Flickr allows users to

organize pictures in photo albums or to add them to thematic collections (*groups*). Upcoming[3], which is a social tagging system for sharing events such as concerts or conferences, allows users to add *links* between the resources via attributes, for example, events having the same location are automatically connected so that users can explore similar resources. Just like the social connectivity, the resource connectivity might also foster the alignment of the underlying folksonomy, because users are better enabled to inspect what kind of tags have been assigned to similar resources as if the resources would be isolated (*no explicit connection*).

User incentives. Users might tag for different reasons. Marlow et al. differentiate between (i) future retrieval, (ii) contribution and sharing, (iii) attracting attention (iv) play and competition, (v) self presentation, and (vi) opinion expression [158]. Ames and Naaman further structure these incentives into a functional and social dimension [42]. Regarding the functional dimension, users tag either for the purpose of *organization* (e.g., contribution and sharing) or *communication* (e.g., self presentation). And regarding social tagging incentives, Ames and Naaman distinguish between tagging activities that are performed rather for the tagger herself (*self*) such as facilitating personal future retrieval and activities that are motivated by social aspects (*social*) such as attracting attention.

The user incentives can also be deduced from the type of tags the people use. Golder and Huberman [103] introduced a classification of tags. Bischoff et al. refined this classification and proposed eight main categories [61]: topic, time, location, type, author/owner, opinions/qualities, usage context and self reference. Hence, tags such as "really-cool" or "annoying" would be categorized as *opinion tags* and the motivation of the user to add such tags might be *social signaling*, i.e. the user possibly would like to express her opinion and communicate this opinion to other users. Thom-Santelli et al. moreover identify social tagging roles and label users, whose tagging motivation relies in social signaling, as *evangelists* [206]. Further they identify roles such as *community-seeker*, who utilize tags to find and get in contact with people from a certain community, or *community-builder*, who establish and re-use tags applied by a certain community.

The characteristics of a social tagging system influence the evolution of the underlying folksonomy and consequently also impact algorithms that exploit the folksonomy structures. Hence, some of the design decisions regarding folksonomy-based ranking algorithms (see below) are influenced by the above tagging characteristics.

Folksonomy Models

Folksonomies can be divided into *broad* folksonomies, which allow different users to assign the same tag to the same resource, and *narrow* folksonomies, in which the same tag can be assigned to a resource only once [160]. Formal models of a folksonomy are, for example, presented by Halpin et al. [110] or Mika [169] and are based on bindings

[3] `http://upcoming.yahoo.com`

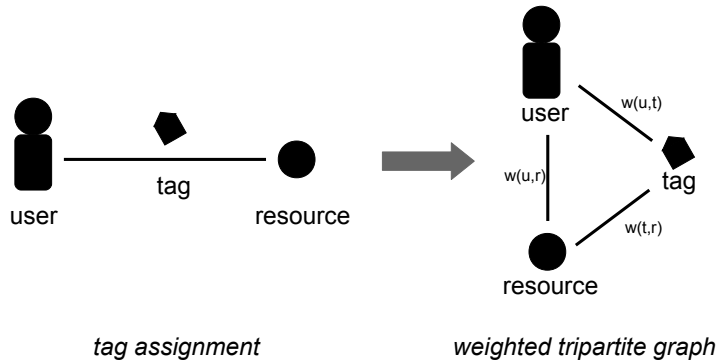


Figure 2.3: Transformation of tag assignment into a weighted, tripartite graph (w(u,t) denotes the weight of the edge between a user and a tag, etc.).

between users, tags, and resources. Hotho et al. define a folksonomy as depicted in Definition 2.1 [124].

Definition 2.1 (Folksonomy) *A* folksonomy *is a quadruple* $\mathbb{F} := (U, T, R, Y)$*, where:*

- U*,* T*,* R *are finite sets of instances of* users*,* tags*, and* resources*, respectively, and*
- Y *defines a relation, the* tag assignment*, between these sets, that is,* $Y \subseteq U \times T \times R$*.*

Wu et al. moreover attribute timestamps to tag assignments to specify when a tag assignment was performed by a user [214] and Hotho et al. also embed relations between tags (super-sub-concept relationships) into the formal folksonomy model, because such relations can explicitly be specified by users of BibSonomy[4], a social bookmarking system developed by the authors [121].

A folksonomy can be interpreted as a 3-uniform hypergraph [53] where each edge corresponds to a tag assignment so that $\mathbb{G} = (V, E)$, where $V = U \cup T \cup R$ is the set of vertices and $E = \{\{u, t, r\} | (u, t, r) \in Y\}$ is the set of hyperedges (cf. [124]). Further, a folksonomy can be transformed into a tripartite undirected graph, which is denoted as *folksonomy graph* $\mathbb{G}_\mathbb{F}$.

Definition 2.2 (Folksonomy Graph) $\mathbb{G}_\mathbb{F} = (V_\mathbb{F}, E_\mathbb{F})$ *is an undirected weighted tripartite graph that models a given folksonomy* $\mathbb{F}$*, where:*

- $V_\mathbb{F} = U \cup T \cup R$ *is the set of nodes,*
- $E_\mathbb{F} = \{\{u, t\}, \{t, r\}, \{u, r\} | (u,\ t, r) \in Y\}\}$ *is the set of edges, and*
- *a weight* w *is associated with each edge* $e \in E_\mathbb{F}$*.*

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

Figure 2.3 illustrates the transformation of tag assignments into the tripartite folksonomy graph. The weight associated with an edge $\{u,t\}$, $\{t,r\}$, and $\{u,r\}$ usually corresponds to the co-occurrence frequency of the corresponding nodes within the set of tag assignments Y (cf. [124, 51]). For example, $w(t,r) = |\{u \in U : (u,t,r) \in Y\}|$ corresponds to the number of users that assigned tag t to resource r. The ability to model a folksonomy as a (hyper-)graph implies that a folksonomy can be represented by matrices (cf. *matrix associated with G* in [53], for example, by an adjacency matrix A where $A[i,j]$ denotes the weight of an edge $\{i,j\}$. In Section2.2 and Section 4.2 we present algorithms that exploit such matrix representations.

User Modeling in Folksonomies

User modeling describes the process of deriving knowledge about people where the kind of knowledge depends on the particular domain [187]. There exist different approaches to user modeling such as *stereotyping* [188], which applies so-called stereotypes to construct a user profile, or *overlay modeling* [104], where user profiles overlay some reference model. An overview on user modeling techniques is for example given in [141, 142]. These approaches can be distinguished with respect to different dimensions such as the temporal space (e.g., long-term vs. short-term user characteristics) or information source (e.g. is information rather explicitly provided by the user or is it deduced from the user behavior?) [187]. Creating user profiles from tagging activities of the users can be considered as a rather implicit way of obtaining user feedback. A straightforward approach to model users in folksonomies is to model them by means of their *personomy*, which represents the tagging activities a particular user performed (see Definition 2.3) [124].

Definition 2.3 (Personomy) *The personomy $\mathbb{P}_u = (T_u, R_u, I_u)$ of a given user $u \in U$ is the restriction of $\mathbb{F}$ to u, where:*

- *T_u and R_u are finite sets of* tags *and* resources *respectively that are referenced from tag assignments performed by the user u and*
- *I_u defines a relation between these sets: $I_u := \{(t,r) \in T_u \times R_u | (u,t,r) \in Y\}$.*

Such personomies can be exploited to create tag-based profiles. Firan et al. exploit such personomy structures to create tag-based and resource-based user profiles which are sets of weighted tags and resources respectively [97]. A naive approach to determine the weights associated with tags is to count how often a user u applied a given tag t [167]: $w_u(t) = |\{r \in R_u : (t,r) \in I_u\}|$. Michlmayr and Cayzer further introduce a tag-based user modeling approach, *Add-A-Tag*, that considers also the temporal evolution of tag-based profiles [168]. Add-A-Tag applies ant colony optimization techniques [89]: the weights of relations between users and tags decrease over time when a user has not used a tag for a long time.

2.1.3 Enhancing the Semantics of Folksonomies

A disadvantage of today's folksonomy systems is that they are designed for humans and do not comply with the vision of the Semantic Web [60]. Although many of these systems feed back data to the web, interoperability is still not supported sufficiently because application programming interfaces are proprietary. Semantic Web standards such as RDF [140] in combination with vocabularies such as FOAF [67], the *Friend-Of-A-Friend* ontology for describing people and documents and specifying relations among these entities, or SIOC [64], an ontology for interlinking social communities on the Web, are used seldomly, for example, regarding vocabulary standards many systems are limited to RSS [211] and do not export their data in semantically more meaningful ways.

Revyu [113], a social tagging system for sharing reviews, sets a good example as it adheres to the principles of Linked Data [56] and therewith enables software agents to navigate through its folksonomy data corpus. The Linked Data initiative aims to connect distributed data on the Web and promotes four basic design principles [56, 62]:

1. Use URIs as names for things.
2. Use HTTP URIs so that people can look up those names.
3. When someone looks up a URI, provide useful information, using the standards (RDF [140], SPARQL [180]).
4. Include links to other URIs so that they can discover more things.

The above rules support interoperability as the meaning of concepts is clearly defined via resolvable HTTP URIs which applications can look up to obtain a description of the corresponding concept. However, in social tagging systems resources are described via tags where the semantic meaning of tags is not clearly defined, because the same tag may have different meanings or different tags may refer to the same thing. The MOAT (*Meaning Of A Tag*) framework [178] can be applied to solve this problem by means of a collaborative approach, in which users manually map tags to ontology concepts by selecting appropriate URIs that define the intended meaning of a tag. MOAT requires a knowledge repository like DBpedia [47], the RDF representation of the Wikipedia encyclopedia, Geonames[5], a geographical database with more than 2.5 million places, or Sindice [208], a search engine for the Semantic Web, to look up appropriate URIs that will be suggested to the user during the tagging process. Passant et al. also extend the so-called *tag ontology* [171] by a *tagMeaning* property so that the semantics of tag assignments can be made available as RDF, for example:

```
<rdf:RDF xmlns="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:tag="http://www.holygoat.co.uk/owl/redwood/0.1/tags/"
    xmlns:moat="http://moat-project.org/ns#"
    xmlns:foaf="http://xmlns.com/foaf/0.1/"
```

[5] http://geonames.org

```
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">

<tag:RestrictedTagging>
  <foaf:maker rdf:resource="http://fabianabel.de/foaf.rdf#fabian"/>
  <tag:associatedTag rdf:resource="http://tags.moat-project.org/tag/apple"/>
  <tag:taggedResource rdf:resource="http://www.apple.com/ipad/"/>
  <moat:tagMeaning rdf:resource="http://dbpedia.org/resource/Apple_Inc."/>
</tag:RestrictedTagging>

</rdf:RDF>
```

The above tag assignment specifies that the user (*http://fabianabel.de/foaf.rdf#fabian*) annotated a resource (*http://www.apple.com/ipad/*) with the tag *"apple"*. The DBpedia URI clearly specifies that the tag assignment refers to the company but rather not to the fruit. In LODr [177], MOAT enables users to revise and semantically enrich the tag assignments they performed at platforms such as Flickr or Delicious. The social bookmarking system Faviki [170] also follows the MOAT approach and allows for *semantic tagging* by enabling users to attach URIs to their bookmarks. The idea of semantic annotations for Web sites is not new and has been studied in systems like Annotea [133]. However, leveraging semantic annotations with social annotations contributed by the masses is a new trend. In accordance to this, Ankolekar et al. [44] postulate a paradigm shift from a rather machine-centered view of the Semantic Web towards a more user- and community-centered approach. Gruber refers to these Social Web initiatives that make use of Semantic Web technologies as *Social Semantic Web* [105] and claims that this would allow for advanced applications like as tag search across multiple sites or combining tags with structured queries.

Further, there exists research on learning relations and ontological structures from social annotations and folksonomies particularly. Hotho et al. [122] show that association rule mining [41] can be applied to learn sub-super-concept relations from folksonomies. For example, if two tags t_1 and t_2 often co-occur at same resources and t_1 is used significantly more often than t_2 in the folksonomy then there is a high chance that t_1 is some sort of super-concept of t_2. Similar approaches that exploit tag co-occurrences are proposed, for example, by Mika [169] or Brooks and Montanez [71]. Balby Marinho et al. [50] follow these approaches and denote a taxonomy that is constructed from a folksonomy as *collabulary*. And Körner et al. show that there is a causal relation between individual tagging practices and emergent semantics in folksonomies [146].

2.2 Information Retrieval in the Social Web

In this section we overview research related with ranking in the Web and in Social Web systems systems. We recap ranking algorithms important for Web search such as PageRank [176] or HITS [139], discuss ranking algorithms that exploit structures

resulting from social tagging and summarize approaches for personalizing the retrieval process in social tagging systems.

2.2.1 Ranking in the Web

The Web consists of millions of Web resources so that the retrieval of relevant resources is a non-trivial task. Ranking supports the retrieval process and is an important feature in various applications such as Web search or enterprise search [149]. Web pages are often formatted using the Hypertext Markup Language (HTML) and are connected via hyperlinks [57]. Given these links, the Web can be modeled as graph, in which each node corresponds to a Web page and a directed edge is used to represent a hyperlink from one page to another. This network of interlinked nodes allows for the application of link analysis techniques, which have been applied already in the 1950s. For example, Katz proposed a method for computing a *status index*, representing the reputation of an entity, by analyzing *who* and *how many other entities* referred to the entity [136]. At the end of the 1990s Brin et al. and Kleinberg developed the first link analysis algorithms for supporting Web information retrieval: PageRank [176] and the Hyperlink-Induced Topic Search (HITS) algorithm [139].

PageRank

The PageRank algorithm calculates ranking for each crawled Web page and is one of the key features of the Google search engine [68]. The ranking of a Web page represents its importance within a set of Web pages. These pages and their connections are modeled by means of a graph $G = (V, E)$, where V is the set of nodes representing the Web pages and E is the set of directed edges that represent the links between the Web pages. An edge (q, p) is contained in E ($(q, p) \in E$) if there exists a link from q to p. The PageRank algorithm analyzes the quality of incoming links to determine the ranking of a page p: the higher the rank of pages q that link to page p ($q : (q, p) \in E$) the higher the rank of p. In particular, the PageRank of a Web page p is defined as follows [176].

$$PageRank(p) = d \cdot \sum_{q:(q,p)\in E} \frac{PageRank(q)}{|\{(q, p')|(q, p') \in E\}|} + \frac{1-d}{|V|} \tag{2.1}$$

Hence, the PageRank of p is the sum of PageRank scores of pages q, which link to p, multiplied by the probability of following the link from q to p, which is modeled by the probability of randomly selecting the link (q, p) from q's outgoing links (q, p') ($\{(q, p')|(q, p') \in E\}$). The sum of (incoming) PageRank values is further multiplied with a residual probability $d \in [0, 1]$, where $1 - d$ models the probability that a user visits a Web page without following a link so that $1-d/|V|$ corresponds to the probability that a user randomly jumps to a page p (Page et al. suggest to set $d = 0.85$ [176]).

The PageRank formula has an intuitive basis in random walks on graphs. It models the behavior of a *random surfer* on the Web graph [176, 68]: the random surfer continuously clicks on links at random without having any priority regarding which link to follow. The probability of selecting an outgoing link (q, p') at page q thus corresponds to the reciprocal number of q's outgoing links. Periodically, the random surfer becomes tired of following links, but jumps to a randomly chosen page. In Equation 2.1, the probability of a random jump is $1-d$ and the probability of jumping to a page $p \in V$ follows a uniform distribution. The Personalized PageRank [176] allows also for other distributions and foresees the consideration of user preferences: instead of randomly jumping to any page $p \in E$, a Web page is selected according to the user's preferences.

The PageRank algorithm has been further developed by other researchers as well. For example, Kamvar et al. [134] and Eiron et al. [91] tackled the issue of *dangling links*, i.e. links to pages without any outgoing links, for which it is not clear how the PageRank scores should be propagated—Page et al. suggest to simply remove these edges before computing PageRank [176]. Broder et al. [69] and Kohlschütter et al. [144] worked on the efficient (parallel) computation of PageRank. Haveliwala [112] introduced a topic-sensitive version of PageRank, where ranking scores are computed within the context of the main categories used in the Open Directory Project[6] (ODP). Baeza-Yates and Davis proposed WLRank (Weighted Links Rank), a PageRank variant that utilizes linking features such as anchor text length or the relative position of a link within a page to adjust the weights of links [48].

HITS

Kleinberg's Hyperlink-Induced Topic Search (HITS) algorithm [139] allows for the detection of hub and authority entities in hyperlinked network structures. A *hub* describes an entity that links to many high quality authority entities and an *authority* denotes an entity, which is linked by many high quality hub entities. Hence, the HITS algorithm is based on a mutually reinforcing relationship between hubs and authorities. Therefore, the operations that update the authority weight $x^{\langle p \rangle}$ and hub weight $y^{\langle p \rangle}$ of an entity p are defined by the operations A and H [139].

$$A : x^{\langle p \rangle} \leftarrow \sum_{q:(q,p)\in E} y^{\langle q \rangle} \tag{2.2}$$

$$H : y^{\langle p \rangle} \leftarrow \sum_{q:(p,q)\in E} x^{\langle q \rangle} \tag{2.3}$$

Here, E denotes the set of directed edges within the given graph G. The core algorithm of HITS, which detects the authorities and hubs in a given graph G, performs k iterations

[6] `http://dmoz.org`

in order to update $x^{\langle p \rangle}$ and $y^{\langle p \rangle}$ for each entity (node) within G. The core iteration is defined as follows [139].

Definition 2.4 (HITS iteration) *The core HITS iteration applies Equation 2.2 and Equation 2.3 to a given graph G.*

function *iterate(G, k)*
G: a graph containing n linked entities
Let x and y be vectors containing the authority and hub weights.
Set x_0 *and* y_0 *to* $(\frac{1}{n}, \frac{1}{n}, \frac{1}{n}, ...) \in \mathbb{R}^n$
for $i = 1, 2, ..., k$ ***do:***
$x'_i \leftarrow$ *apply A to* (x_{i-1}, y_{i-1})
$y'_i \leftarrow$ *apply H to* (x'_i, y_{i-1})
$x_i \leftarrow ||x'_i||_1$
$y_i \leftarrow ||y'_i||_1$
end
return (x_k, y_k)

The graph G that is passed to the core iteration of HITS has to be a directed graph. In general, G is a partial Web graph consisting of linked resources that are possibly relevant to a certain topic (cf. [139]).

2.2.2 Ranking in Folksonomy Systems

For folksonomy systems, one can apply traditional ranking approaches that, for example, represent resources by means of vector space models [192] where each dimension corresponds to a tag and the value for each dimension is computed via some weighting scheme. For example, Gemmell et al. [100] apply *TFxIDF* weighting, i.e. the weight associated with a tag t for a given resource r corresponds to the *term frequency* (TF), which refers to the number of users that assigned tag t to the given resource, multiplied by the *inverse document frequency* (IDF), which measures the importance of t in the folksonomy.

In Section 2.1 we saw that social tagging induces structures, so-called folksonomies, which can be modeled as graphs (folksonomy graph, see Definition 2.2). In the following paragraphs we will outline graph-based ranking algorithms for folksonomies that follow ranking strategies such as PageRank [176] or HITS [139] (see above) and that will be used as baseline ranking strategies in our experiments on search and personalization in the subsequent chapters.

FolkRank

The FolkRank algorithm [124] operates on the folksonomy model specified in Definition 2.1. The core idea of the FolkRank algorithm is to transform the hypergraph formed by the traditional tag assignments into an undirected, weighted tripartite graph $\mathbb{G}_{\mathbb{F}} = (V_{\mathbb{F}}, E_{\mathbb{F}})$, which serves as input for an adaption of PageRank [176]. At this, the set of nodes is $V_{\mathbb{F}} = U \cup T \cup R$ and the set of edges is given via $E_{\mathbb{F}} = \{\{u,t\},\{t,r\},\{u,r\}|(u,t,r) \in Y\}\}$ (cf. Definition 2.1). The weight w of each edge is determined according to its frequency within the set of tag assignments, i.e. $w(u,t) = |\{r \in R : (u,t,r) \in Y\}|$ is the number of resources the user u tagged with keyword t. Accordingly, $w(t,r)$ counts the number of users who annotated resource r with tag t, and $w(u,r)$ determines the number of tags a user u assigned to a resource r. With $\mathbb{G}_{\mathbb{F}}$ represented by the real matrix A, which is obtained from the adjacency matrix by normalizing each row to have 1-norm equal to 1, and starting with any vector $\vec{w}$ of non-negative reals, the adapted PageRank iterates as follows:

$$\vec{w} \leftarrow dA\vec{w} + (1-d)\vec{p}. \tag{2.4}$$

The adapted PageRank utilizes vector $\vec{p}$ as a preference vector, fulfilling the condition $||\vec{w}||_1 = ||\vec{p}||_1$. Its influence can be adjusted by $d \in [0,1]$. Based on this, FolkRank is defined as follows [124].

Definition 2.5 (FolkRank) *The* FolkRank algorithm *computes a topic-specific ranking in folksonomies by executing the following steps:*

1. *$\vec{p}$ specifies the preference in a topic (e.g. preference for a given tag).*
2. *$\vec{w}_0$ is the result of applying the adapted PageRank with $d = 1$.*
3. *$\vec{w}_1$ is the result of applying the adapted PageRank with some $d < 1$.*
4. *$\vec{w} = \vec{w}_1 - \vec{w}_0$ is the final weight vector. $\vec{w}[x]$ denotes the* FolkRank *of $x \in V$.*

Hence, FolkRank applies the adapted PageRank (see Equation 2.4) twice, first with $d = 1$ and second with $d < 1$. The final vector, $\vec{w} = \vec{w}_{d<1} - \vec{w}_{d=1}$, contains the *FolkRank* of each folksonomy entity. In our experiments we will make use of FolkRank and, unless otherwise noted, set $d = 0.7$ as suggested by Hotho et al. [124].

SocialPageRank

The SocialPageRank algorithm [51] is motivated by the observation that there is a strong interdependency between the popularity of users, tags, and resources within a folksonomy. For example, resources become popular when they are annotated by many

users with popular tags, while tags, on the other hand, become popular when many users attach them to popular resources.

SocialPageRank constructs the folksonomy graph $\mathbb{G}_{\mathbb{F}}$ similarly to FolkRank. However, $\mathbb{G}_{\mathbb{F}}$ is modeled within three different adjacency matrices. A_{TR} models the edges between tags and resources. The weight $w(t,r)$ is computed as done in the FolkRank algorithm (cf. Section 2.2.2): $w(t,r) = |\{u \in U : (u,t,r) \in Y\}|$. The matrices A_{RU} and A_{UT} describe the edges between resources and users, and users and tags respectively. $w(r,u)$ and $w(u,t)$ are again determined correspondingly. The SocialPageRank algorithm results in a vector $\vec{r}$ whose items indicate the *social PageRank* of a resource.

Definition 2.6 (SocialPageRank) *The* SocialPageRank algorithm *(see [51]) computes a ranking of resources in folksonomies by executing the following steps:*

1. ***Input:*** *Association matrices A_{TR}, A_{RU}, A_{UT}, and a randomly chosen SocialPageRank vector $\vec{r}_0$.*
2. ***until*** *$\vec{r}_i$ converges* ***do****:*
   a) $\vec{u}_i = A_{RU}^T \cdot \vec{r}_i$
   b) $\vec{t}_i = A_{UT}^T \cdot \vec{u}_i$
   c) $\vec{r'}_i = A_{TR}^T \cdot \vec{t}_i$
   d) $\vec{t'}_i = A_{TR} \cdot \vec{r'}_i$
   e) $\vec{u'}_i = A_{UT} \cdot \vec{t'}_i$
   f) $\vec{r}_{i+1} = A_{RU} \cdot \vec{u'}_i$
3. ***Output:*** *SocialPageRank vector $\vec{r}$.*

SocialPageRank and FolkRank both base on the PageRank algorithm. Regarding the underlying *random surfer model* of PageRank [176], a remarkable difference between the algorithms relies on the types of links that can be followed by the "random surfer". SocialPageRank restricts the "random surfer" to paths in the form of resource-user-tag-resource-tag-user, whereas FolkRank is more flexible and allows e.g. also paths like resource-tag-resource.

SocialSimRank

The SocialSimRank algorithm [51] computes the similarity between two tags of a folksonomy. SocialSimRank adapts the idea of SimRank [129] and states that similar tags are usually assigned to similar resources. Definition 2.7 outlines the SocialSimRank algorithm as proposed in [51].

Definition 2.7 (SocialSimRank) *The* SocialSimRank *algorithm computes a ranking of tags in folksonomies by executing the following steps:*

1. ***Input:*** *Association matrix A_{TR}, tag similarity matrix S_T^0, and resource similarity matrix S_R^0*
2. ***Init:*** $S_T^0(t_i, t_j) = 1$ *for each* $t_i = t_j$, *otherwise 0*
   $S_R^0(r_i, r_j) = 1$ *for each* $t_i = t_j$, *otherwise 0*
3. ***until*** S_T *converges* ***do****:*
   for each *annotation pair* (t_i, t_j) ***do****:*
   $$S_T^{k+1}(t_i, t_j) = \frac{C_T}{|R(t_i)||R(t_j)|} \cdot \sum_{m \in R(a_i)} \sum_{n \in R(a_j)} \frac{min(A_{TR}(t_i, m), A_{TR}(t_j, n))}{max(A_{TR}(t_i, m), A_{TR}(t_j, n))} S_R^k(m, n)$$
   for each *resource pair* (r_i, r_j) ***do****:*
   $$S_R^{k+1}(r_i, r_j) = \frac{C_R}{|T(r_i)||T(r_j)|} \cdot \sum_{m \in T(r_i)} \sum_{n \in T(r_j)} \frac{min(A_{TR}(m, r_i), A_{TR}(n, r_j))}{max(A_{TR}(m, r_i), A_{TR}(n, r_j))} S_T^k(m, n)$$
4. ***Output:*** *SocialSimRank matrix S_T*

SocialSimRank utilizes the association matrix A_{TR} for tags and resources, which is also part of the SocialPageRank algorithm. The weight $w(t, r)$, which is needed to fill the matrix, is computed as done in the FolkRank algorithm (cf. Section 2.2.2): $w(t, r) = |\{u \in U : (u, t, r) \in Y\}|$. $A_{TR}(t_i, r_j)$ therewith corresponds to the number of users, who have annotated resource r_j with tag t_i. $R(t_i)$ is the set of resources that are tagged with t_i and $T(r_j)$ correspondingly defines the set of tags that are assigned to r_j. C_T and C_R are constant damping factors, which allow to adjust the similarity propagation of tags and resources respectively. In our experiments we set C_T and C_R to 0.7 as done in [51].

Tag-based HITS algorithm

Kleinberg's HITS algorithm [139] (see above) allows for the detection of hub and authority entities in directed network structures. The above folksonomy-based ranking algorithms, by contrast, operate on the undirected folksonomy graph ($\mathbb{G}_{\mathbb{F}}$, see Definition 2.2). Tag assignments do not explicitly prescribe a direction. Hence, the challenge of applying HITS to folksonomies is to transform a folksonomy into a directed graph.

Wu et al. [213] propose the following strategy to construct directed edges from the set of tag assignments: for each tag assignment $(u, t, r) \in Y$ two edges *"$u \rightarrow t$"* and *"$t \rightarrow r$"* will be constructed. Figure 2.4 illustrates this transformation. The resulting link structure implies that hubs are restricted to be users while the authority role is bound to resources. In our evaluations we will denote this strategy as *naive HITS* algorithm. Further, we will introduce *SocialHITS* [4], a HITS-based algorithm which does not limit the role of hubs and authorities to certain folksonomy entity types, but makes it possible to detect authoritative users as well.

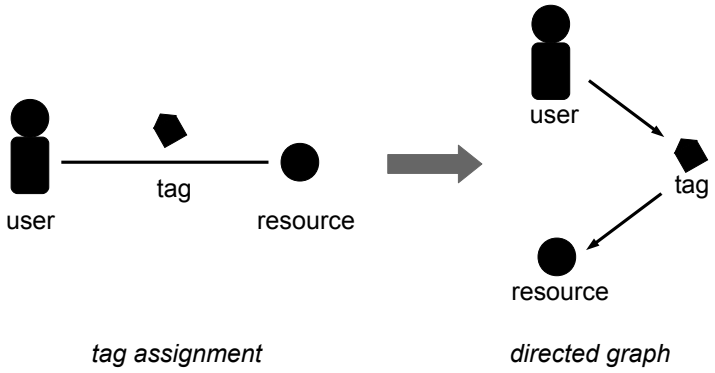


Figure 2.4: Transformation of tag assignment into a directed graph.

2.2.3 Personalization in Folksonomy Systems

In the field of personalization in folksonomy systems we overview research on recommender systems that exploit folksonomy structures and give insights on personalized search.

Folksonomy-based Recommender Systems

Recommender systems support the users in finding or selecting relevant items. Folksonomy-based recommender systems exploit the folksonomy structure and particularly the set of tag assignments (cf. Definition 2.1). Given the basic folksonomy model, there are three main types of items that are of interest for folksonomy-based recommender systems: users, tags, and resources. Recent research in this field mainly focussed on tag recommendations and competitions have been established that aim for optimizing tag recommendations [120, 92].

Supporting users during the tagging process is an important step towards easy-to-use applications. Consequently, different approaches have been studied to find best tag recommendations for resources. The tag recommendation challenge can be specified in different ways. For example, *personalized* tag recommendation algorithms as proposed by Rendle et al. [185, 186] or Yin et al. [218] aim to predict tags a given user will assign to a given resource, while *non-personalized* tag recommendation strategies as investigated by Heymann et al. [118] or Krestel et al. [148] aim to predict appropriate tags to resources independently from the personal preferences of the users. Further, tag recommendation methods can be classified by the kind of features they are analyzing, for example: recommending items by analyzing content [151, 40, 73]), by analyzing tag co-occurrences [199]), or graph-based approaches which analyze the folksonomy graph [128] (cf. Definition 2.2)).

Jäschke et al. evaluate a FolkRank-based tag recommender [128] and compare the performance of their approach to recommendation strategies that apply traditional collab-

orative filtering techniques [193]. The obtained results are measured by a leave-one-out strategy [159]: a complete *post* of a user (i.e. all tags that a certain user has given to some resource) is deleted, and the recommendation strategies are evaluated with respect to their ability to find / recommend the left out data (based on Last.fm and BibSonomy datasets). Some constraints are set on the data used for the evaluation to ensure that each user, tag, and resource occurs in at least as many posts as specified by a so-called *p-core* level. Reported results are only for the levels 5 and 10, which means that only the "dense" part of the folksonomy has been considered for evaluation. In Chapter 5 we will introduce advanced graph-based tag recommendation strategies and compare them with the FolkRank-based approach introduced by Jäschke et al. for arbitrary p-core levels. Sigurbjörnsson and van Zwol compare different tag recommendation strategies in Flickr [199]. They propose a tag recommendation method that first identifies—by analyzing global tag co-occurrence statistics—candidate sets of tags for each of the tags that are already assigned to the given resource. By aggregating and ranking these candidate sets the algorithm determines suitable recommendations.

Recommending resources based on tags has not been researched extensively yet. Sen et al. refer to these folksonomy-based resource recommender systems that predict user preferences for resources by inferring their preferences for tags as *tagommenders* [197]. They show that tagommender algorithms perform better than traditional collaborative filtering techniques. However, as also reported by Firan et al. [97], who investigated the benefits of tag-based user profiles for Last.fm music recommendations, ranking resources by means of cosine similarity between items and users performed worse than advanced machine learning algorithms such as SVM [85]. Further, Sen et al. [197] conclude that for predicting explicit user preferences in resources (*ratings*) collaborative filtering such as [156] that directly exploit known user preferences in resources rather than inferring such preferences via interests in tags perform better. Hence, recommending resources by exploiting folksonomies is a non-trivial task and becomes even more complex in situation where new users register to the system so that recommender systems have to overcome the so-called *cold-start problem* [196]. In Chapter 6 we will investigate strategies to solve this problem.

Social networking services like Facebook or LinkedIn feature user-to-user recommendations, e.g. they suggest users to add certain people to their social network. Terveen and McDonald denote this kind of recommendation challenge as *social matching* [205]. Recent studies showed that by exploiting social network information such as the number of friends two users share these recommendations achieve high precisions [81, 108]. Gertner et al. demonstrated that also the usage of other data sources such as shared bookmarks might support the social matching challenge [101]. However, the impact of exploiting folksonomy structures on the social matching problem is not well explored yet. In Chapter 5 we will explore this problem and evaluate different approaches for ranking users according to a given context.

Personalized Search in Folksonomies

The goal of personalized search is to adapt search result rankings to the specific needs of a user. In the context of Web search, two main approaches to personalized search have been studied: (1) modifying the search query issued by the user or (2) processing the search result so that is conforms to the information needs of the user [179]. For example, search queries can be modified by means of query expansion [210] and search result rankings can be generated so that they conform to a given topic the user might be interested in [181]. In the field of folksonomy systems, personalized search has not been studied in detail yet.

Noll and Meinel show that tag-based profiles from the social bookmarking system Delicious can be applied to personalize Web search [172]. Lerman et al. show how tag-based user profiles can be applied to answer ambiguous search queries in Flickr [154]. Xu et al. proposed a framework for personalized search in folksonomies that represents users, queries and resources in a vector space model [192] where each dimension corresponds to a topic. Cosine similarity is measured between the user and the resources as well as the query and the resources to construct a user-specific ranking r_{user} and a query-specific ranking r_{query} respectively. These rankings are then aggregated using the so-called *Borda method* (cf. [90]) to compute the ranking score for a given resource r with respect to a query q issued by a specific user u.

$$r(u, q, r) = \gamma \cdot r_{query}(q, r) + (1 - \gamma) \cdot r_{user}(u, r) \tag{2.5}$$

In their evaluation, Xu et al. compare different weighting schemes such as *TFxIDF* [201] or *BM25* [190] to set the values in the user, resource and query vectors respectively. However, the authors use the folksonomy, which they exploit to construct the tag-based personalized rankings, also as ground truth: a resource is considered as relevant for a given user if the user has annotated this resource. Hence, the improvements over the baseline, which does not exploit folksonomy structures but utilizes categories from the open directory project, have to be validated in particular because recent work showed that the strategy of applying cosine similarity for ranking resources in the context of recommender systems is outperformed by other approaches [197] (cf. above section).

Gemmell et al. also apply cosine similarity in combination with TF and TFxIDF weighting schemes to personalize the search experience in folksonomy systems [100]. They deduce clusters of tags that are used to represent user preferences and resources as well and show that weighting based on term frequencies is outperformed by TFxIDF weighting. Cai and Li moreover create advanced tag-based resource profiles to adapt search results to the tag-based profiles of the users [74].

2.3 Research Questions answered in this Thesis

The Social Web fosters the participation of a large user community in creating and sharing resources on the Web. Social tagging systems such as Delicious, Flickr, or any other systems are an essential part of the Social Web. In this chapter we discussed general background of social tagging and information retrieval in the Social Web. Research in this field is still in its early stages and there are many questions that require to be further investigated. In the following we will briefly summarize some of these open research questions that will be answered in this thesis.

Context Modeling in Folksonomy Systems. Formal folksonomy models have been proposed by related work [124, 169, 214]. The essential structure of these models are tag assignments, i.e. user-tag-resource bindings (cf. Section 2.1.2), and yet there exists no generic approach to also incorporate contextual information related to a tag assignment activity.

- How can contextual information be modeled in folksonomies?
- How can folksonomy systems deduce semantically meaningful contextual information from tagging activities?

In Chapter 3 we will answer these questions and propose a generic folksonomy model that allows to attach context information to tag assignments. Further, we will describe how we implemented this model in two different folksonomy systems and show how these systems deduce valuable semantics from social tagging.

Search and Ranking in Folksonomy Systems. In Section 2.2.2 we outlined existing ranking algorithms that support information retrieval in folksonomy systems. However, algorithms proposed by related work [51, 124, 213] do not consider contextual information embedded into folksonomies like the semantic meaning of the tag assignments or information about the context of the user who performed a tagging activity. Given an advanced folksonomy model, open research questions with respect to information retrieval are thus:

- How to design ranking algorithms that exploit context information available in folksonomies?
- How does the exploitation of context information available in folksonomies impact information retrieval performance?

These questions will be answered in Chapter 4 by introducing various ranking algorithms for folksonomies and by evaluating the proposed ranking algorithms with respect to search in folksonomy systems.

User Modeling and Personalization in Social Web Systems. Recent research suggests to model users in folksonomy systems by means of a so-called personomy [124] or tag-based profiles which describe user preferences in tags [97, 168]. Research on

user modeling on the Web suggests to model both user *and* context [126]. However, the impact of user and context modeling on personalization on the Social Web has not been studied extensively yet.

- How can user and context modeling strategies support personalization in Social Web systems?
- Which type of user and context modeling strategy is the most appropriate for recommender systems and personalized search?

In Chapter 5 we will introduce different user and context modeling strategies and evaluate these strategies with respect to experiments on recommender systems and personalized search.

Cross-system User Modeling in the Social Web. Approaches such as the Meaning Of A Tag (MOAT) approach [178] and other Social Semantic Web activities [44, 105] support interoperability between folksonomy systems. While there exists studies on user modeling in Web-based systems, cross-system user modeling in the Social Web and its impact on personalization has not been researched yet in detail.

- How to model users across system boundaries in the Social Web?
- What are the benefits of cross-system user modeling in the Social Web and how does it impact the performance of social recommender systems?

Answers to these questions will be explored in Chapter 6 where we introduce and evaluate a framework for cross-system user modeling in the Social Web.

3 Design and Implementation of Context Models for Folksonomy Systems

Given background information on folksonomy systems from the previous chapter, we now introduce the context folksonomy model that builds the basis for our ranking algorithms (see Chapter 4). Further, we present two folksonomy systems we developed that adhere to this model. The main contributions of this chapter have been published in [1, 3, 10, 12, 19, 21, 25, 28, 35, 36].

3.1 Introduction: What is Context within the scope of Folksonomies?

With the success of social media systems like Flickr, Delicious, etc. tagging has become en vogue as it allows users to easily organize content with freely chosen keywords (tags) and facilitates sharing of content as well. The structures that evolve like desire lines [166] over time when users (*folks*) annotate resources (like images, websites, etc.) with respect to their own *taxonomy* are called *folksonomies*[161]. Hence, systems that allow for tagging are called folksonomy systems. Formalizing the model of a folksonomy has already been done in [121, 169, 214] by means of a set of user-tag-resource bindings (possibly attached with a timestamp), which can be modeled as hypergraph. Based on such models one can build valuable applications that support information retrieval. For example, there exists research on exploiting folksonomies in order to realize search and ranking algorithms [51, 123], compute recommendations [73, 128, 199], deduce real-world events from the tagging behavior [182], or model users [97, 155, 167]. However, these approaches suffer from the lack of well-defined semantics in folksonomies. For example, the semantic meaning of a tag assignment can be ambiguous. Moreover, not all tags are appropriate for search because tag assignments are possibly performed to rather express an opinion than describing the content of a resource [61].

In this thesis we investigate whether additional context information helps to overcome the problem of missing semantics. Additional context may be formed by extending the traditional model of tag assignments (user-tag-resource bindings) with additional dimensions or by attaching (meta-)data to the tag assignments that describes the par-

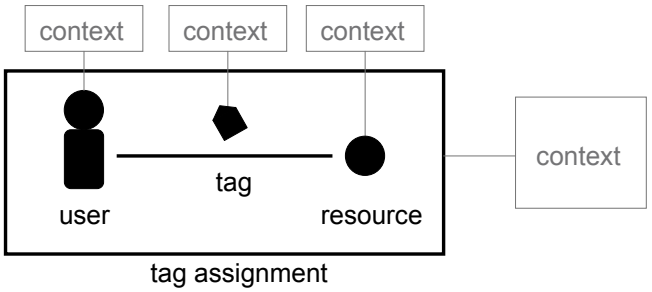


Figure 3.1: The four dimensions of context in folksonomies. Contextual information can refer to the *user*, who performed a tag assignment, to the *tag*, which was applied, to the *resource*, which was annotated, or to the *tag assignment* activity itself.

ticular tagging activity more precisely. For example, a timestamp helps to categorize tag assignments in a temporal manner, the mood the user had when tagging the resource would allow to qualify opinions expressed in a tag assignments and knowledge about the actual intention of the user could be exploited to distinguish ambiguous tags. Additional context results also from descriptions that are attached to the entities of the folksonomy, i.e. to the users, tags, or resources. Background knowledge about the user could, for example, be applied to classify tag assignments in terms of reliability. Figure 3.1 illustrates the four different dimension of context in folksonomies, where contextual information either refers to the folksonomy entities (user, tag, resource) or to the tag assignment itself. In this chapter, we mainly focus on context information that is attached to tag assignments. We introduce a generic context folksonomy model and present two reference systems that implement this model.

From a system's design perspective, another disadvantage of today's folksonomy systems is that they do not comply with the vision of the Semantic Web [60] as they are designed for humans only. Although many of these systems feed back data to the Web, interoperability is still not supported sufficiently because application programming interfaces are proprietary and the use of Semantic Web standards is most often avoided, e.g. regarding vocabulary standards it is, to a large degree, limited to RSS. Revyu [113], by contrast, sets a good example as it adheres to the principles of Linked Data [56] and therewith enables software agents to navigate through its folksonomy data corpus. However, the semantic meaning of tags is not understandable for these agents. The MOAT (*Meaning Of A Tag*) framework [178] can be applied to solve this problem by means of a collaborative approach, in which users manually map tags to ontology concepts by selecting appropriate URIs that define the intended meaning of a tag. Interoperability between folksonomy systems would allow for advanced (mash-up) applications such as recommender systems that exploit distributed folksonomy data (see Chapter 6). In this chapter, we will present two folksonomy systems we developed that bring together Social Web and Semantic Web technologies and thus make the *Social Semantic Web* [105] tangible.

In summary, we will answer the following research questions.

- How can contextual information be modeled in folksonomies?
- How can folksonomy systems deduce semantically meaningful contextual information from tagging activities?
- How can interoperability between folksonomy systems be increased?

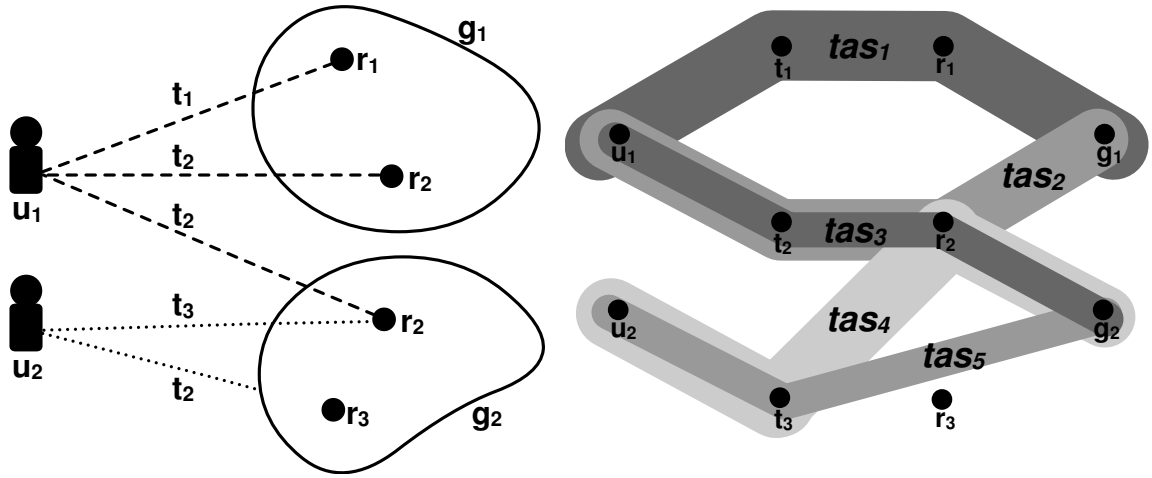
In Section 3.2 we will first discuss folksonomy models that allow for contextual information. In Section 3.3, we will describe two folksonomy systems we developed as part of this thesis. Both systems demonstrate how contextualization can be implemented in folksonomy systems and illustrate the benefits of additional semantics gained by the folksonomy models presented in the previous section. We conclude this chapter with a short discussion and summary of our main contributions.

3.2 Context Folksonomy Model

Some systems imply a folksonomy model that incorporates additional information indicating in which context a tag was assigned to a resource. In particular, such context might be formed by other resources the tagged resource is grouped with. For example, a user might annotate an image with *"paris"* in context of an album that is entitled *"Trip to France"* and contains other images tagged with terms like *"france"* or *"travel"*. The album context can help to clarify the intended semantic meaning of *"paris"* which could refer to the capital of France, to diverse cities in the USA or to people who are called *Paris*. As groups might facilitate the interpretation of tag assignments, we thus introduce the notion of groups to folksonomies.

Definition 3.1 (Group) *A* group *is a finite set of resources.*

A group is a resource as well. Groups can be tagged or arranged in groups as well which might imply hierarchies among resources. Figure 3.2 shows a scenario where tagging of resources is done in context of groups. Users u_1 and u_2 have grouped resources r_{1-3} into g_1 and g_2, and tagged both, resources and groups with keywords t_{1-3}. The tag assignment tas_2 (u_1, t_2, r_2, g_1) in Figure 3.2(a) describes that user u_1 has annotated resource r_2 in context of group g_1 with tag t_2. The group context helps to detect ambiguous tags. For example, in Figure 3.2(a) t_2 is attached to different resources (r_2 and g_2) in context of two different groups. Hence, there is a probability that the meaning of t_2 is ambiguous. However, as g_1 and g_2 have an overlap of resources (r_2 occurs in both groups), there is evidence that the topics of g_1 and g_2 are similar which indicates that the meaning of t_2 is probably the same in both groups. Assume that there is a group which does not contain any of the resources of g_1 and t_2 would be the only tag that occurs in both groups then meaning of t_2 is possibly ambiguous. If users assign tags to a group, which



(a) Tagging scenario

(b) Hypergraph representation

Figure 3.2: Tagging in context of groups: (a) scenario in which two users assign tags to resources in context of different groups and (b) the corresponding hypergraph representation.

is itself not contained in a group, then the group context information is not available ($\rightarrow (u_2, t_2, g_2, \varepsilon)$) and within the hypergraph representation the tag assignment can be interpreted as an edge containing only three vertices ($\rightarrow tas_5$).

Given the notion of groups, we define the corresponding *group context folksonomy* as specified in Definition 3.2 [27].

Definition 3.2 (Group Context Folksonomy) *A* group context folksonomy *is a* 5*-tuple* $\mathbb{F} := (U, T, \breve{R}, G, \breve{Y})$*, where:*

- U*,* T*,* R*,* G *are finite sets that contain instances of users, tags, resources, and groups, respectively,*
- $\breve{R} = R \cup G$ *is the union of the set of resources and the set of groups and*
- $\breve{Y}$ *defines a tag assignment having a* group context*:* $\breve{Y} \subseteq U \times T \times \breve{R} \times (G \cup \{\varepsilon\})$*, where* ε *is a reserved symbol for the* empty group context*, i.e. if there is no group context available.*

Group context folksonomies evolve in systems like GroupMe![1] [10] (see Section 3.3), which allows for tagging of bookmarks in the context of a group of related bookmarks, or Flickr, which enables users to create sets of images they can tag.

[1] `http://groupme.org`

In Definition 3.2, the group context is attached to the tag assignments. We will use this folksonomy model whenever we deal with groups of resources that form the context of a tag assignment. In Definition 3.3 we introduce a more generic folksonomy model that allows us to attach arbitrary type of context to tag assignments [19].

Definition 3.3 (Context Folksonomy) *A* context folksonomy *is a tuple* $\mathbb{F} := (U, T, R, Y, C, Z)$*, where:*

- U*,* T*,* R*,* C *are finite sets of instances of* users*,* tags*,* resources*, and* context information *respectively,*
- Y *defines a relation, the* tag assignment *that is,* $Y \subseteq U \times T \times R$ *and*
- Z *defines a relation, the* context assignment *that is* $Z \subseteq Y \times C$

Given the context folksonomy model, it is possible to attach any kind of context to tag assignments. For example, the model allows for tagging tag assignments. TagMe![2] [36] (see Section 3.4), a tagging and exploration front-end for Flickr pictures, introduces three types of context:

- spatial information describing to which part of a resource a tag assignment belongs to,
- categories for organizing tag assignments, and
- URIs that describe the semantic meaning of a tag assignment.

Such context information is simply assigned to a tag assignment by the relation Z. For example, given a tag assignment $tas_1 = (u_1, t_1, r_1)$, one can make the meaning of t_1 more explicit by attaching the unique resource identifier uri_1 which defines the meaning of t_1 with respect to tag assignment tas_1, i.e. $(tas_1, uri_i) \in Z$. Following Definition 3.2 where a group context is a resource and can therewith be tagged as well, we also allow for tagging context so that: $|R \cap C| \geq 0$.

A group context folksonomy can easily be transformed into a context folksonomy as follows.

1. Add each group $g \in \breve{R}$ to C.
2. For each tag assignment $tas_{group} = (u, t, r, g) \in \breve{Y}$:
   a) Add $tas_{context} = (u, t, r)$ to Y.
   b) If $g \neq \epsilon$: create a new context assignment $(tas_{context}, g)$.

In the the subsequent section we will introduce two folksonomy systems where we implemented the context folksonomy model. We will refer to Definition 3.2 if we operate

[2] `http://tagme.groupme.org`

on context that is embedded into the folksonomy by means of group structures (groups of resources) and Definition 3.3 if we operate on other kind of context.

3.3 GroupMe! – Enhancing Social Bookmarking with Context

GroupMe![3] [10] extends the idea of social bookmarking systems with the ability to create groups of multimedia Web resources. Therefore, it provides an enjoyable interface, which enables the creation of groups via *drag & drop* operations. Resources within GroupMe! groups are visualized according to their media type so that users can grasp content without visiting each resource separately. GroupMe! groups form new sources of information as they bundle content, which is, according to the group creator, relevant for the topic of a group. GroupMe! groups are not only accessible for humans, but also for third-party applications because GroupMe! captures user interactions as RDF, i.e. whenever a user adds a resource to a group, annotates a resource/group, etc. GroupMe! produces RDF.

Figure 3.3 shows a screenshot of the GroupMe! system that illustrates how users can create GroupMe! groups via easy drag-and-drop operations. In the example, a user is creating a GroupMe! group entitled "Trip to Hypertext '09, Turin", in which she collects diverse Web resources useful for organizing her trip to the so-called Hypertext conference in Turin. She added already bookmarks referring to the conference website, some video and pictures showing sights of Turin, a map of Turin's city center, an RSS feed reporting about the current weather conditions in Turin, the contact details of a professor she would like to visit during her stay in Turin, etc. Tags can be attached to the individual resources as well as to the entire group. GroupMe! visualizes the grouped resources according to their media types so that users can obtain important information at a glance. For example, the videos can be watched immediately without navigating to the actual Web site, the RSS feed automatically lists the latest items and the contact details (email, phone number) of the linked contact are already previewed. End-users can easily create new groups via drag-and-drop (see Figure 3.3) or via the GroupMe! *bookmarklet*.

Group Builder. GroupMe! integrates different services like Google or Flickr that enable users to discover and search for resources they may want to add to their groups. Figure 3.3 demonstrates how a user drags an image gathered from Flickr into her group. Drag-and-drop operations also allow to arrange resources within a group, i.e. to position and resize resources. We applied these features also to the so-called *news story creator* of SYNC3[4] to enable journalists and bloggers to create and connect news story artifacts [143].

[3] `http://groupme.org`
[4] `http://sync3.eu`

Figure 3.3: Screenshot of GroupMe! application: a user drags a photo from the right-hand side Flickr search bar into the GroupMe! group. Available online: `http://groupme.org/GroupMe/group/3355`

Browser Button. While browsing the Web users can click on the GroupMe! browser button (*bookmarklet*) to add resources, they are interested in, to a group. When clicking the button users are directed to an input form where they can select the group(s) and specify tags they want to assign to the resource.

In addition to these features, there exist third-party applications that connect to GroupMe!'s API (see below) to allow users the creation of GroupMe! groups. For example, we integrated GroupMe! into the e-learning platform LearnWeb2.0 [38] to support users in organizing their learning resources.

GroupMe! groups are interpreted as regular Web resources and can also be arranged within groups. This enables users to build hierarchies among Web resources and to make use of the information hiding principle – detailed information can be encapsulated into groups. Users that just want to get a rough overview about a topic do not need to visit those groups that contain detailed information.

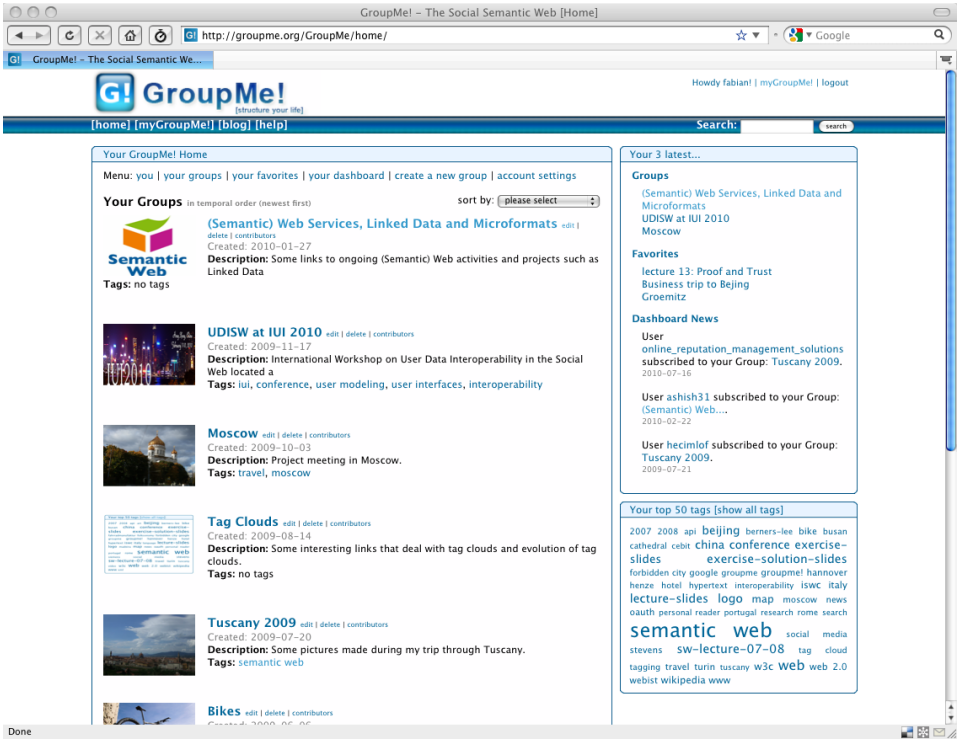
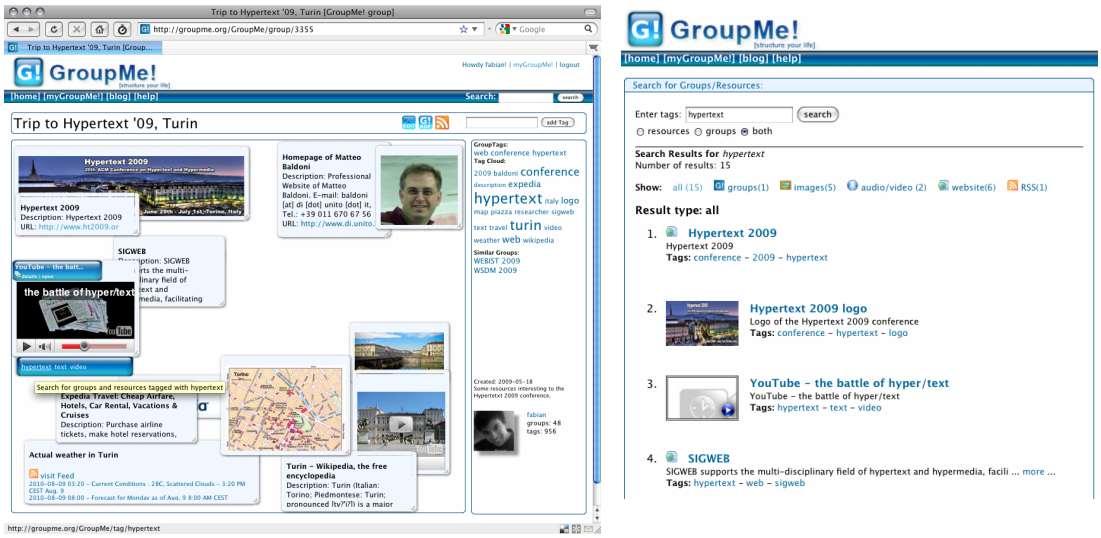


Figure 3.4: Screenshot of the personal GroupMe! page. It lists the groups a user has created, groups she has subscribed to, events that recently occurred in groups of interest (*dashboard*), etc. Via a personal tag cloud the user can navigate to groups and resources she has annotated.

GroupMe! groups are dynamic collections, which may change over time. Other users that also plan to attend the Hypertext conference (see Figure 3.3) are enabled to subscribe to the group and will be notified at their *personal GroupMe! page* (see Figure 3.4) whenever the group is modified, e.g. a new resource is added or removed, new tags have been assigned, etc. Users can also utilize their favored news reader for being notified as each GroupMe! group provides an RSS feed that reports about recent activities related to the group. Thus, GroupMe! can be considered as a lightweight blogging tool where creation of blog entries is done via simple mouse operations instead of writing text [12]. Information content is captured also by the group context, e.g. by adding the website "powerset.com" to a group "Promising Web 2.0 companies" the user denotes what he thinks about the corresponding company.

In order to ease future retrieval GroupMe! allows to tag both resources and groups. The personal GroupMe! page lists tags, that a user has assigned to resources/groups she is interested in: the *user tag cloud*. By clicking on a tag t within the user tag cloud she receives all resources/groups she has annotated with t and has the opportunity to navigate also to other resources related to t. Tag clouds are furthermore computed and displayed for each GroupMe! group (see Figure 3.5(a)). Such group-specific tag clouds help users to get an overview about the topic of a group. Another advantage of group-



(a) GroupMe! group (b) Search

Figure 3.5: Visualization of groups and search results: (a) GroupMe! groups visualize group-specific tag clouds (right sidebar) and tags assigned to the individual resources (bar at the bottom of resources) which allow users to (b) search for related resources.

specific tag clouds is that they enable users to explore the GroupMe! corpus. Clicking on a tag t of a *group tag cloud* invokes a GroupMe! search operation, which results in a list of related resources and groups (see Figure 3.5(b)) – not only those resources which are directly tagged with t, as described in Chapter 4. Starting from a search result list, user can navigate to other resources and groups. In general, all entities in GroupMe! – users, tags, resources, and groups – are clickable and resolvable, which results in an advanced browsing experience, e.g. each group points to similar groups (see top right in Figure 3.5(a)), or resources refer to groups they are contained in.

3.3.1 Tagging in GroupMe!

GroupMe! competes with social tagging systems like BibSonomy, Delicious, or Flickr. Table 3.1 summarizes some characteristics of GroupMe! according to the dimensions in the *tagging system design taxonomy* developed in [158], and compares them with the characteristics of related tagging systems [22].

Tagging rights. GroupMe! allows every user to tag resources and groups (*free-for-all*) as this enables us to gather more tags about a resource and also a higher variety of keywords than in constrained systems. However, Flickr restricts tagging e.g. to the resource owner, friends, or contacts.

Dimension/System	*GroupMe!*	*BibSonomy*	*Delicious*	*Flickr*
Tagging rights	free-for-all	free-for-all	free-for-all	permission-based
Tagging support	blind/viewable	suggested	suggested	viewable
Aggregation model	bag	bag	bag	set
Object type	multimedia	textual	textual	images
Source of material	global	global	global	user-contributed
Social connectivity	links	links, groups	links	links
Resource connectivity	groups	none	none	groups
User incentives	- future retrieval - contribution - sharing - attract attention - self presentation	- future retrieval - contribution - sharing	- future retrieval - contribution - sharing - attract attention	- future retrieval - contribution - sharing - attract attention - self presentation

Table 3.1: GroupMe! tagging design in comparison to other social tagging systems. And user incentives in terms of tagging.

Tagging support. When users annotate resources they are not supported with tag suggestions as this would limit the variety of tags. However, they have the ability to list tags that have already been assigned to a resource in context of the actual group. Tags, that have been assigned in context of other groups – and hence are possibly not appropriate in the actual group context – are not visible to the user when tagging (*blind/viewable*).

Aggregation model. In comparison to Flickr, which does not allow for duplicated tags (*set*), GroupMe! allows different users to assign the same tag to a certain resource (*bag*). This may enable a better evaluation of the importance of the tags.

Object type. GroupMe! is the only system listed in Table 3.1 that supports tagging of resources displayed in a multimedia fashion. Although systems like Delicious enable users to bookmark and tag arbitrary Web resources, they just visualize resources in a textual way. Hence, while tagging e.g. an image in Delicious, users usually do not see the image they tag.

Source of material. Resources that can be annotated and grouped in GroupMe! are globally distributed over the Web, and referenced by their URL. This enables GroupMe! to handle often changing resources like RSS feeds appropriately: whenever a group is accessed, the most recent versions of the contained resources are displayed.

Social connectivity. All systems listed in Table 3.1 allow users to be linked together. GroupMe! does not provide integrated features, but utilizes users' FOAF descriptions in order to identify links between users.

Resource connectivity. Independent of the users' tags, a few resource sharing systems provide other features to connect resources. There are some systems that allow users to organize themselves into groups, and that provide functionality to retrieve resources, which are related to these groups – e.g. BibSonomy or CiteULike[5].

[5] `http://www.citeulike.org`

However, Flickr and GroupMe! are the only tagging systems listed in Table 3.1 that enable users to assign resources to groups explicitly. Such hand-selected groups are highly valued by the users as indicated in our analysis in the subsequent chapters.

User incentives. GroupMe! users have several motivations to annotate resource ranging from simplification of future retrieval to self presentation (e.g. some users tag resources with *holiday* in order to express which locations they have visited).

What makes GroupMe! unique is that (1) users can assign tags to entire groups and (2) resources are always tagged in context of a specific group. Thereby, GroupMe! extends the traditional folksonomy model as described in Section 3.2.

3.3.2 GroupMe! System Architecture

GroupMe! is a modular Web application that adheres to the Model-View-Controller pattern [184]. It is implemented using the J2EE application framework *Spring*[6]. Figure 3.6 illustrates the underlying architecture, which consists of four basic layers:

Aggregation. The aggregation layer provides functionality to search for resources a user wants to add into GroupMe! groups. Currently, GroupMe! supports Google, Flickr, and of course a GroupMe!-internal search, as well as adding resources by specifying their URL manually. *Content Extractors* allow us to process gathered resources in order to extract useful data and metadata, which are converted to RDF using well-known vocabularies.

Model. The core GroupMe! model is—in accordance with the group context folksonomy model (see Definition 3.2)—composed of four main concepts: *User*, *Tag*, *Group*, and *Resource*. In addition, the model covers concepts concerning the users' arrangements of groups, etc. The *Data Access* layer cares about storing model objects. The actual data store back-end is arbitrarily exchangeable. At the moment we are using a MySQL database.

Application logic. The logic layer provides various controllers for modifying the model, exporting RDF, etc. The internal GroupMe! search functionality, which is implemented according to the strategy pattern in order to switch between different search and ranking strategies, is made available via a RESTful API. It enables third parties to benefit from the improved search capabilities (cf. Chapter 4), and to retrieve RDF descriptions about resources – even such resources that were not equipped with RDF descriptions before they were integrated into GroupMe!. To simplify usage of exported RDF data, we further provide a lightweight Java *Client API*, which transforms RDF into GroupMe! model objects. We also launched a generic library for mapping between RDF and Java instances which is called *SemREST* and available via SourceForge: `http://semrest.sourceforge.net/`.

[6] `http://springframework.org`

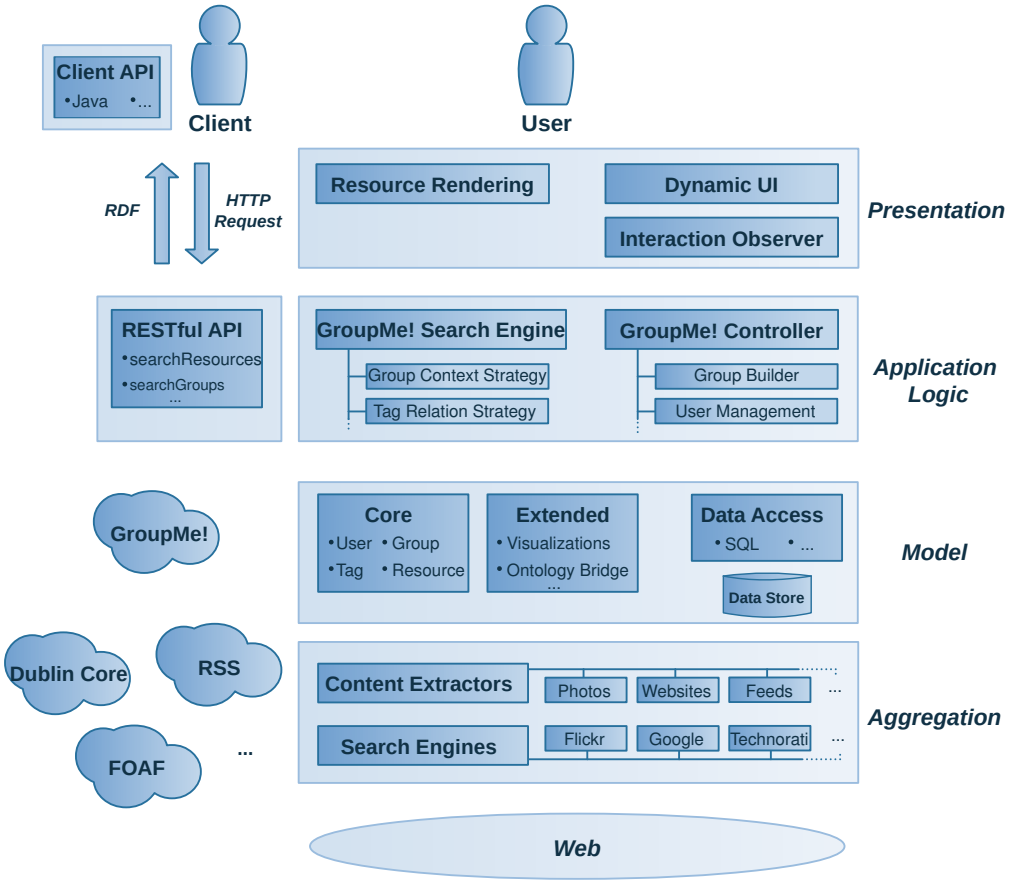


Figure 3.6: Technical overview of the GroupMe! system.

Presentation. The GUI of the GroupMe! application is based on AJAX principles. Therefore, we applied Ajax and JavaScript frameworks like script.aculo.us[7], DWR[8], or Prototype[9]. Such frameworks provide already functionality to drag and drop elements, resize elements, etc. Visualization of groups and resources is highly modular and extensible. Switching between components that render a specific resource or type of resource can be done dynamically, e.g. visualization of group elements is adapted to their media type (see Figure 3.5(a)). We further experimented with user-adaptable visualizations that enable individual users to visualize groups according to their current needs (e.g., timelines, lists) [212].

When creating or modifying groups, each user interaction (e.g. moving and resizing resources) is monitored and immediately communicated to the responsible GroupMe! controller so that e.g. the actual size or position of a resource within a group is stored in the database.

[7] http://script.aculo.us
[8] http://getahead.org/dwr
[9] http://prototypejs.org

GroupMe! can also be considered as an RDF generator that enriches the Web of data with RDF statements as follows.

1. Each user interaction (grouping and tagging) is captured as RDF using several vocabularies, e.g. FOAF [67] and a GroupMe!-specific vocabulary[10] that defines new GroupMe! concepts. External applications can therewith utilize information gained within the GroupMe! system like the information that two resources are grouped together, or a certain tag was assigned to a resource within the context of a group.

2. Whenever a user adds a Web resource into a group, domain dependent content extractors gather useful (meta-) data so that resources can be enriched with semantically well defined descriptions. When e.g. adding a Flickr photo into a group, a *Photo content extractor* translates Flickr-specific descriptions into RDF descriptions using *DCMI element set*[11]. Some content extractors make use of Aperture[12] which facilitates extraction of data and metadata from different information systems and file formats.

3.3.3 Linked Data in GroupMe!

The RDF data generated in GroupMe! is made available according to the principles of Linked Data [56]. Whenever an agent requests RDF—which is done via HTTP content negotiation as described in [194]—then useful information as well as links to related URIs are delivered to the agent so that the agent can navigate through the GroupMe! resources.

The novel group semantics have the advantage that they relate Web resources, which were not related before. When a user adds, for example, a Flickr image (see Figure 3.5(a)) together with a Google map into the same group then both resources are immediately linked to each other. GroupMe! also acquires additional metadata when a resource is dropped into a group. In the given example, the Google Maps resource is equipped with longitude (*wgs84:long*) and latitude (*wgs84:lat*) attributes and hence the Flickr image can now also benefit from such geographical information as it can possibly also be related to the corresponding location.

The GroupMe! ontology [1], which models the additional semantics and the group context folksonomy (see Definition 3.2) in particular, is outlined in Figure 3.7. It integrates existing ontologies and mainly introduces four new concepts.

Group A *Group* is a collection of resources (*skos:Collection*, cf. [66]). As groups can be grouped like any other document, we define *Group* to be a subclass of *foaf:Document*.

[10] `http://groupme.org/rdf/groupme.owl`
[11] `http://dublincore.org/documents/dces/`
[12] `http://aperture.sourceforge.net`

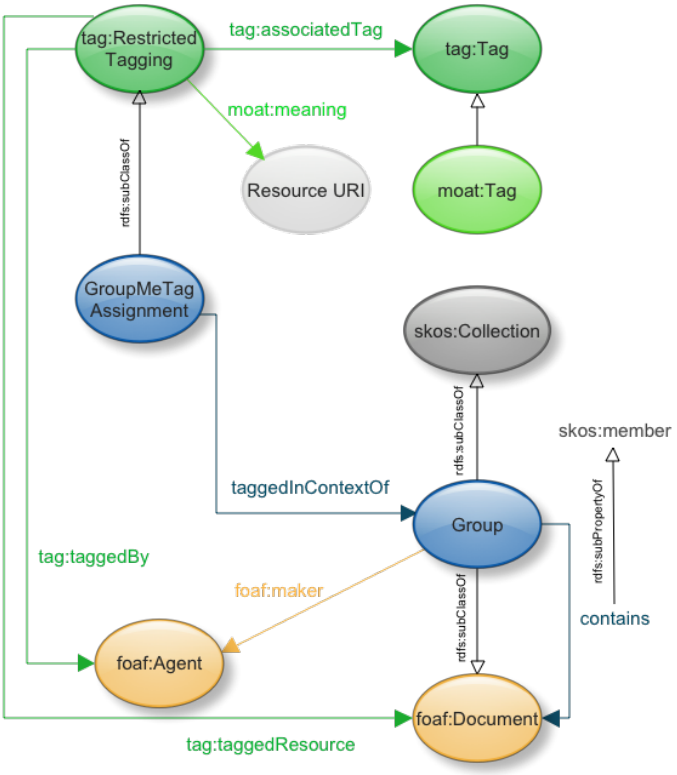


Figure 3.7: GroupMe! ontology.

contains The property *contains* is correspondingly a subproperty of *skos:member* and describes that a resource (*foaf:Document*) is contained in a group.

GroupMeTagAssignment The GroupMe! tag assignment (cf. Definition 3.2) is modeled as subclass of *tag:RestrictedTagging* defined in R. Newman's Tag ontology [171]. We utilize the *RestrictedTagging* with two extensions: (1) the property *moat:meaning* of the MOAT ontology [178], which refers to an ontology concept that defines the intended meaning of the tag, and (2) the *taggedInContextOf* property.

taggedInContextOf The *taggedInContextOf* property refers to the group, in which a user assigned a tag to a certain resource.

Figure 3.8 lists an extract of the RDF that is provided by the GroupMe! group illustrated in Figure 3.5(a). The RDF utilizes the vocabulary introduced by the GroupMe! ontology, e.g. the group links to its resources via *contains* property. It also refers to its creator (*../user/fabian*) and to tags that are directly assigned to the group (e.g. *../tag/hypertext*). The GroupMe! system ensures that RDF descriptions of each folksonomy entity can be accessed by applications that navigate through the linked folksonomy data. The Flickr resource in Figure 3.8 is, e.g., equipped with some provenance metadata (*dc:publisher*, *dc:creator*) and its GroupMe! tag assignments, which are (possibly) enriched with a *moat:meaning* pointing to an ontology concept that unambiguously describes the mean-

a. GroupMe! group:

```
<Group rdf:about="http://groupme.org/GroupMe/group/3355">
 <dc:title>Trip to Hypertext '09, Turin</dc:title>
 <dc:description>
   Some resources important for my trip to Hypertext conference 2009.
 </dc:description>
 <contains rdf:resource="http://groupme.org/GroupMe/resource/2414"/>
 <contains rdf:resource="http://groupme.org/GroupMe/resource/2417"/>
 ...
 <foaf:maker>
  <foaf:Person rdf:about="http://groupme.org/GroupMe/user/fabian">
   <foaf:nick>fabian</foaf:nick>
   <rdfs:seeAlso rdf:resource="http://fabianabel.de/foaf.rdf#me"/>
  </foaf:Person>
 </foaf:maker>
 <tag:taggedWithTag>
  <tag:Tag rdf:about="http://groupme.org/GroupMe/tag/hypertext">
   <tag:name>hypertext</tags:name>
  </tag:Tag>
 </tags:taggedWithTag>
 ...
</Group>
```

b. Flickr image:

```
<foaf:Image rdf:about="http://groupme.org/GroupMe/resource/3695">
 <dc:title>Torino - Piazza Vittorio</dc:title>
 <dc:publisher rdf:datatype="&xsd;anyURI">
   http://flickr.com
 </dc:publisher>
 <dc:contributor rdf:datatype="&xsd;anyURI">
   http://flickr.com/user/7677931@N02
 </dc:contributor>
 <rdfs:seeAlso rdf:resource="http://static.flickr.com/6/5e2a_t.jpg"/>
 ...
 <tag:tag>
  <GroupMeTagAssignment rdf:about="http://groupme.org/GroupMe/tas/5128">
   <tag:associatedTag rdf:resource="http://groupme.org/GroupMe/tag/turin"/>
   <tag:taggedBy rdf:resource="http://groupme.org/GroupMe/user/fabian"/>
   <tag:taggedOn rdf:datatype="&xsd;date">2009-05-18T11:23:40</tag:taggedOn>
   <moat:meaning rdf:resource="http://dbpedia.org/resource/Turin/">
  </GroupMeTagAssignment>
 </tag:tag>
 ...
</foaf:Image>
```

c. Google Maps resource:

```
<foaf:Document rdf:about="http://groupme.org/GroupMe/resource/3699">
 <dc:title>Turin @ Google Maps</dc:title>
 <wsg84:lat rdf:datatype="&xsd;double">45.073036</wsg84:lat>
 <wsg84:long rdf:datatype="&xsd;double">7.809219</wsg84:long>
 ...
</foaf:Document>
```

Figure 3.8: RDF descriptions of Linked Data in the GroupMe! system

ing of the tag in the given context. In [19], we describe how context embedded into the folksonomy helps to derive these URIs automatically.

In Chapter 4 we present algorithms that apply an information propagation model where resources that *meet in GroupMe! groups* [21] benefit from each other. However, GroupMe! does not prescribe how metadata should be propagated among the resources in general. For example, the above mentioned Google Maps resource provides geographical metadata such as longitude/latitude data which may be relevant to the Flickr image (see right in Figure 3.5(a)) as well. Nevertheless, we do not specify any rules which define how the resources can benefit from the metadata of other resources they are grouped with, as we do not understand the *grouping behavior* of the users sufficiently. Applications that access GroupMe! data—for example via the RESTful Semantic Web interface—thus have to decide to which extent metadata of a resource is also appropriate for resources of the same group.

The RESTful Semantic Web interface of GroupMe! [25] follows the *Resource Oriented Architecture* (ROA) [189], which is an architecture that conforms to the REST approach [96]. The API allows other applications to read, add, modify, and delete data by exploiting the main methods of HTTP [95] (GET, POST, PUT, and DELETE).

GET. As mentioned above, the GroupMe! data corpus is made available according to the principles of Linked Data [56]. Applications that request RDF via HTTP GET and HTTP content negotiation (cf. [194]) will be provided with useful information as well as links to related URIs. These URIs enable the applications to navigate through the whole GroupMe! folksonomy. Figure 3.8 lists an extract of the RDF representation that is provided to applications which access the group about Hypertext 2009 conference. The visual representation of that group is displayed in Figure 3.5(a).

POST. The HTTP POST method is used to add new content to the GroupMe! system, e.g. to add a new group to the system, to add resources to groups, or to add annotations to resources or groups. To create a new group an application has to post an RDF resource, which is an instance of *groupme:Group* (cf. GroupMe! ontology explained above), to *http://groupme.org/GroupMe/group*, e.g. the following RDF post would create a new group entitled "REST and Semantic Web".

```
<Group>
 <dc:title>REST and Semantic Web</dc:title>
 <dc:description>
   Information about REST and Semantic Web principles.
 <dc:description>
</Group>
```

GroupMe! cares about the creation of the URI identifying the group and returns it to the sender of the HTTP request, e.g. *http://groupme.org/GroupMe/group/6859*. Groups can be filled with resources by posting resources to the group's URI. Tags and other annotations can be added similarly. Hence, content as listed in Figure 3.8 can not only be created via the graphical user interface (see Figure 3.3)

or via the so-called GroupMe! bookmarklet, but also by posting RDF data to the GroupMe! system.

PUT. If a client application sends an HTTP PUT request to an existing resource then GroupMe! modifies the resource that is identified by the URI according to the RDF data that is sent together with the HTTP request, e.g. the following RDF (as part of an HTTP PUT) would change the title and description of the group "Trip to Hypertext '09, Turin".

```
<Group rdf:about="http://groupme.org/GroupMe/group/3355">
 <dc:title>Traveling to HT '09</dc:title>
 <dc:description>My trip to Hypertext 2009 in Turin, Italy<dc:description>
</Group>
```

DELETE. Deletion of content is done via HTTP DELETE, e.g. in order to remove the tag assignment shown in Figure 3.8, an application has to send the HTTP DELETE request to *http://groupme.org/GroupMe/tas/5128*.

The HTTP methods are therewith utilized in a way that conforms to HTTP and REST as well. In the current implementation of GroupMe! the POST, PUT, and DELETE operations can only be performed by the owner of a group, which is ensured via an authorization token that has to be included in the header of each corresponding HTTP request.

In general, the Semantic RESTful API of GroupMe! is easy to use as it just exploits the semantics of HTTP and the semantics defined in the GroupMe! ontology. The API is already used by other applications. For example, LearnWeb2.0 [38], a platform for exploring and organizing learning resources available on Web 2.0 platforms such as YouTube or Delicious, connects to the RESTful API so that LearnWeb2.0 users can group their learning resources. Further, we developed a GroupMe! client application [174] that enriches GroupMe! tag assignments with URIs describing the semantic meaning of the tag (cf. MOAT approach [178]).

3.3.4 User Acceptance and Usage Patterns

GroupMe! enabled us to deploy and evaluate the algorithms and approaches reported in this thesis in an online setting. End-users could thus immediately benefit from our advanced approaches to search (see Chapter 4) or personalization (see Chapter 5). In this subsection we present general statistics that characterize the usage behavior, while the concrete datasets, on which we run our experiments, are described in the evaluation sections related to GroupMe! (Section 4.3.1, 5.3.2, and 5.4.2).

The data underlying this analysis was collected during the first three years after the system's launch on July 14, 2007. Within this period, GroupMe! had a total of 4234 resources of which 3370 were normal resources and 864 (20.41%) were groups. Altogether, 4929 tag assignments were monitored, with 1.24 tags per resource in average.

The overall evolution of resources and groups is plotted in Figure 3.9. The first abrupt rise at the beginning of March 2008 was caused by CeBIT, the world's largest computer exposition, where we presented GroupMe! to introduce the platform to the industry and public. After three years, the number of active users (more than 650 in July 2010) as well as the number of groups and resources that are published and shared in GroupMe! is still growing continually.

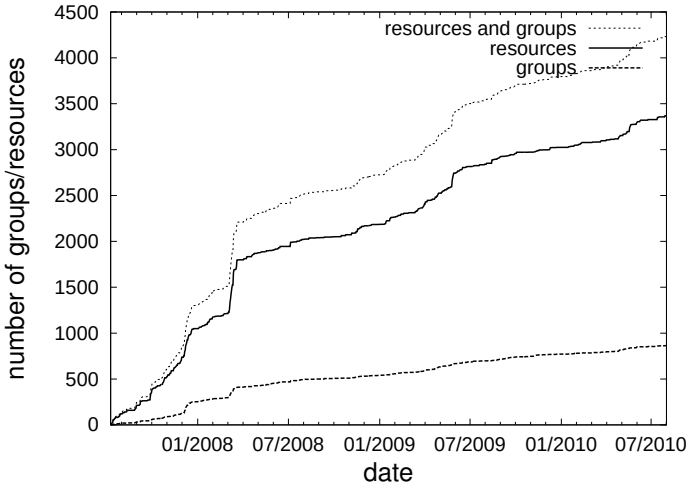


Figure 3.9: Evolution of number of resources/groups.

According to the *tagging system design taxonomy* proposed in [158], GroupMe! is a free-for-all tagging system, which allows users to annotate multimedia content for future retrieval. Hence, GroupMe! allows for broad folksonomies as every user is allowed to tag every resource or group without any restrictions. Tagging a resource r is done when users are situated in the view of a certain group g. Thereby, users are only able to see those tags that have been assigned to r within the context of the group g (same holds for group g). Explicit tag suggestions are not provided by the GroupMe! system. However, the tag cloud of a group and the resource's visualization, which is adapted to the media type of the resource, help the users to reflect on appropriate tags for the resource.

Interestingly, groups were tagged more intensively than ordinary resources. On average, 1.79 tags were assigned to groups, whereas only 1.11 tags were attached to other resources. Thus, groups were tagged 1.6 times more often than traditional resources. Furthermore, only 50.23% of the groups were not annotated with any tag in contrast to 53.06% of the resources. These observations give support for the hypothesis that users adopt the group idea to organize Web resources and that they also invest time in the group construction process.

A typical group in GroupMe! consists of 2 – 8 resources. That we do not observe groups with significantly more members can be explained from the user interface, which gives the users a canvas to place and arrange the Web resources. As the size of this canvas is limited, the on-screen display of the group becomes impractical with too many Web

Type of Resource	AVG Occurrences
images	27.73%
videos	7.16%
rss feeds	2.49%
groups	2.6%
other Web resources	60.02%

Table 3.2: Percentage of resources' media types that are part of GroupMe! groups.

resources. Users collect resources with different media types in their group, as depicted in Table 3.2. Most popular among the media types are images, followed by videos and RSS feeds. Web sites, academic papers, presentation slides, etc. are denoted as *other Web resources* and are not mentioned separately, because to users they appear as simple bookmarks, i.e. their visualization is not adapted to their media type particularly. The possibility to include groups into a group was also used: 2.6% of the grouped resources are GroupMe! groups themselves and 4.9% of the groups contain at least one GroupMe! group.

Given the statistics from Google Analytics[13], it is interesting to see that for more than 35% of the visits (overall more than 120000), users navigated via some image search engine (e.g., *http://images.google.com*) to *http://groupme.org*. This shows how GroupMe! supports search for multimedia resources: by grouping images/videos together with other Web resources such as Wikipedia sites they can benefit from the (metadata) descriptions attached to these resources. For example, 54.4% of the images in GroupMe! are not annotated with tags. However, these images benefit from tags of *neighbor resources*, i.e. resources that are contained in the same group.

3.4 TagMe! – Enhancing Picture Sharing With Context

TagMe![14] [35, 36] is an online image tagging system where users can explore and organize pictures available in Flickr. Figure 3.10 outlines the conceptual architecture of TagMe!, which can basically be considered as an advanced tagging and search interface on top of Flickr. Users can directly import pictures from their own Flickr account or utilize the search interface to retrieve Flickr pictures. If users tag their own images in TagMe! then these tag assignments are propagated to Flickr as well. We developed TagMe! to demonstrate, in contrast to GroupMe! groups, further approaches to contextualization in folksonomy systems and to investigate the benefits of the context folksonomy model (see Definition 3.3) in different settings. TagMe! extends the Flickr tagging functionality with three additional facets.

[13] `http://www.google.com/analytics/`
[14] `http://tagme.groupme.org`

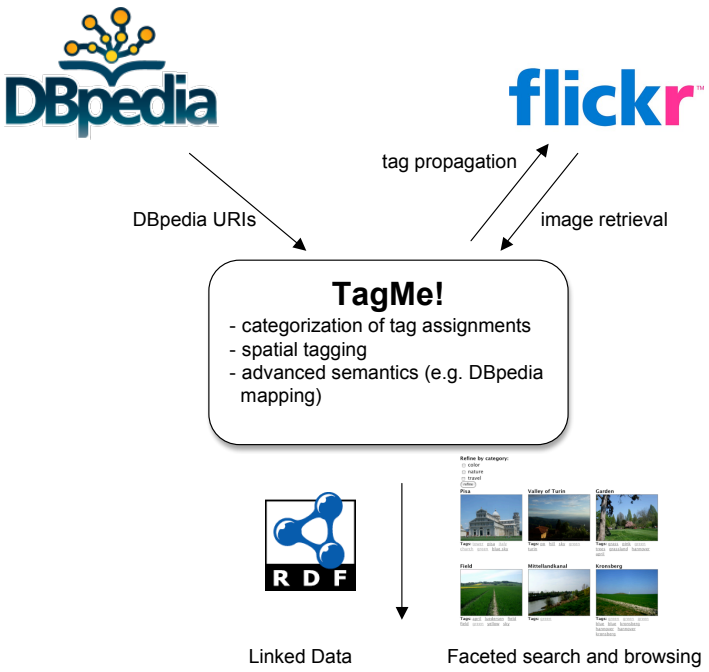


Figure 3.10: Conceptual architecture of TagMe!

Spatial information. TagMe! users are enabled to perform *spatial tag assignments*, i.e. to attach a tag assignment to a specific area of a resource. They can draw a rectangle within the picture (see rectangle within the photo in Figure 3.11) similarly to *notes* in Flickr or annotations in LabelMe [191].

Categories. For each tag assignment users can enter one or more categories that classify the annotation. Categorizing tag assignments can be considered as tagging of tag assignments as the categories can be chosen freely like tags.

URIs. TagMe! maps DBpedia URIs to tag and category assignments by exploiting the DBpedia lookup service[15]. Hence, all tags and categories have well-defined semantics so that applications, which operate on TagMe! data, can clearly understand the meaning of the tag and category assignments.

TagMe! users are motivated to annotate specific areas as each spatial tag assignment has a globally unique URI and is therewith linkable, which allows users to share the link with others so that they can point their friends and other users directly to a specific part of an image [34]. For example, if users follow the link of the spatial tag assignment "opera"[16], shown in Figure 3.11 then they are directed to a page where the corresponding area is highlighted, which might be especially useful in situation where users discuss about specific things within a picture. While the area tags add an enjoyable visible feature for highlighting specific areas of an image and sharing the link to such areas with friends,

[15] http://lookup.dbpedia.org
[16] `http://tagme.groupme.org/TagMe/resource/403/tas/1439`

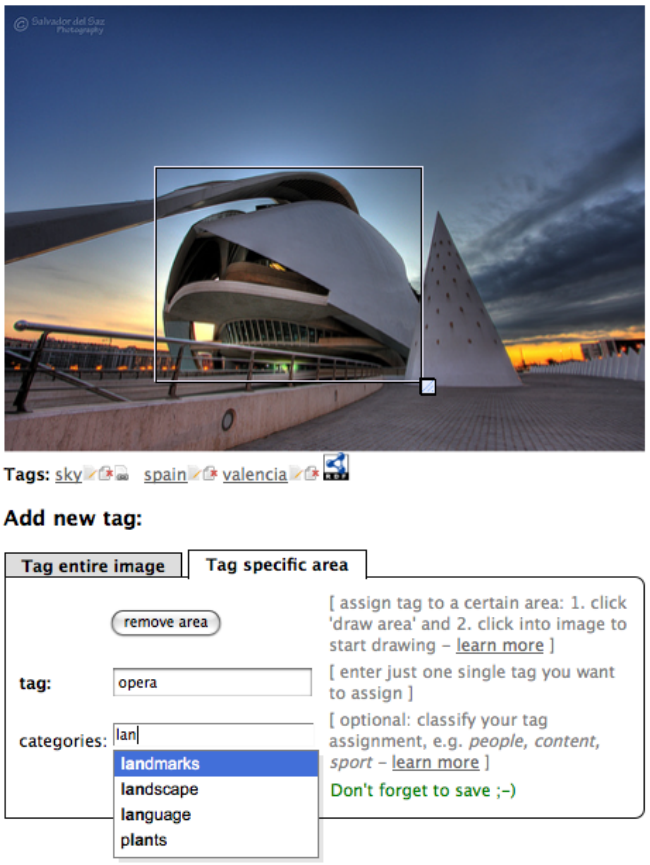


Figure 3.11: User tags an area within an image and categorizes the tag assignment with support of the TagMe! system.

we consider them as highly valuable to improve search by detecting tag correlations or to enhance the identification of similar tags [19]. For example, the size of an area might indicate whether a tag is important for the whole image or just for a specific (possibly small) part of an image.

For each tag assignment the user can enter one or more categories that classify the annotation. While typing in a category, the users get auto-completion suggestions from the pre-existing categories of the user community (see bottom in Figure 3.11). TagMe! users can immediately benefit from the categories as TagMe! provides a faceted search interface that allows to refine tag-based search activities by category (and vice versa) as shown in Figure 3.12.

The (meta-)data created in TagMe! is made available as RDF according to the principles of Linked Data [56], using the MOAT ontology[17] and Tag ontology[18] as primary schemata.

[17] `http://moat-project.org/ns`
[18] `http://www.holygoat.co.uk/projects/tags`

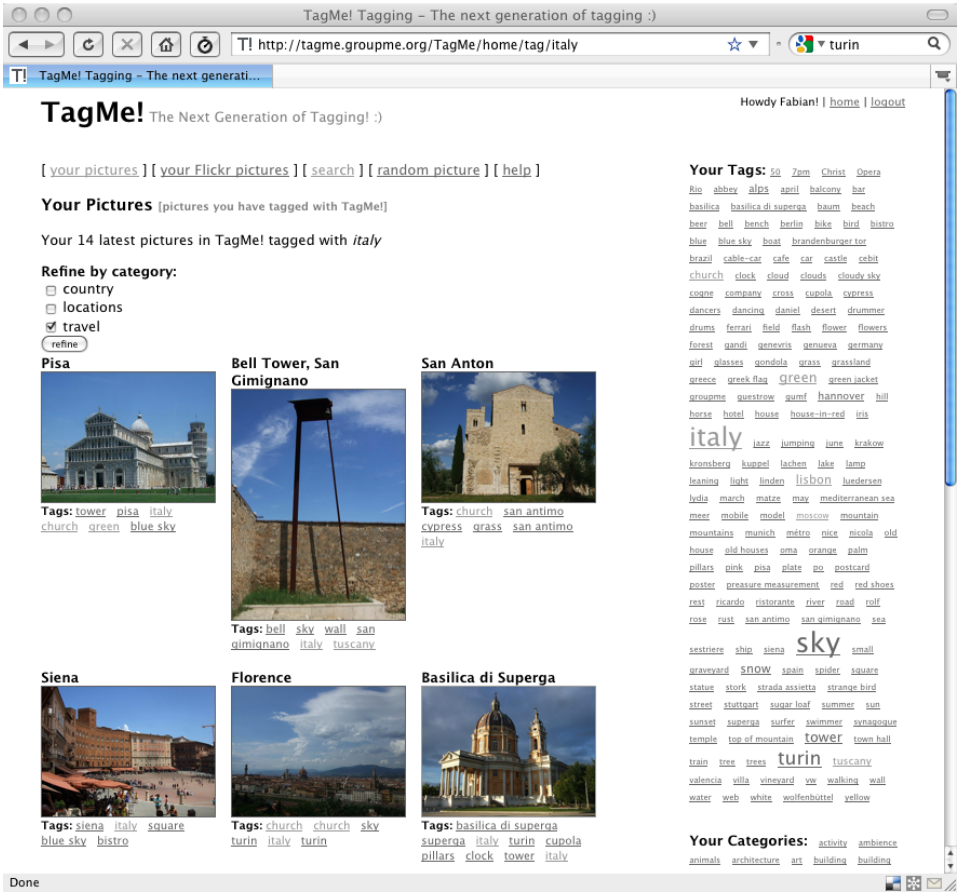


Figure 3.12: Using categories to refine tag-based search.

3.4.1 Tagging in TagMe!

To classify the tagging approach introduced by TagMe!, we compare the tagging and tag-based exploration features of TagMe! from the perspective of the end-users with other tagging systems: Flickr, Delicious, Faviki [170] and LabelMe [191]. Our comparison among the systems is again based on the dimensions of the *tagging system design taxonomy* developed in [158] (cf. Section 3.3.1). For example, we compare the (i) "Tagging rights", (ii) "Tagging support" and (iii) "Aggregation model" of those systems. These characteristics define respectively (i) who can tag, (ii) if the user gets assistance from the system during the tagging process and (iii) whether the system allows users to assign the same tag more than once to a particular resource (aggregation model = bag) or not (aggregation model = set).

We extend the tagging design taxonomy with the additional tagging principles so that we can compare the tagging features provided by TagMe! with features offered by existing tagging systems.

Semantic tagging. We consider tagging as semantic tagging whenever the meaning of

Dimension/System	*Flickr*	*Delicious*	*Faviki*	*LabelMe*	*TagMe!*
Semantic tagging	no	no	yes	no	yes
Spatial tagging	no	no	no	yes	yes
Tag categorization	no	tag bundles	no	no	tas categorization
Tagging support	viewable	suggested	suggested	viewable	suggested
Tagging rights	permission-based	free-for-all	free-for-all	free-for-all	free-for-all
Aggregation model	set	bag	bag	bag	bag

Table 3.3: TagMe! system characteristics in comparison to other social tagging and annotating systems.

a tag is clearly defined, for example, by attaching a URI explaining the meaning of the tag [178].

Spatial tagging. The practice of annotating a specific piece of a resource, e.g., parts of an image or paragraphs in a text.

Tag categorization. A method enabling users to categorize or classify the tags and tag assignments.

Table 3.3 summarizes the characteristics of TagMe! and similar tagging systems according to the taxonomy explained above.

The social bookmarking system Faviki and TagMe! are the only systems listed in Table 3.3 that allow for semantic tagging. Both systems primarily map tag assignments to DBpedia URIs [63]. Faviki requests the end-users to explicitly select the appropriate URIs while TagMe! is doing the mapping automatically. A fundamental restriction of Faviki is that only those tags, which correspond to a meaningful URI, can be assigned to a bookmark. Faviki supports users with a list of URI suggestions from which the users have to select one URI. Delicious and TagMe! provide tagging support by means of auto-completion. Flickr and LabelMe, which is an online annotation tool for images, do not provide tag suggestions but tags already assigned to a resource are *viewable* when adding new tags. In Flickr, users are not allowed to assign the same tag more than once to a particular resource (aggregation model = set) and moreover the owner of a picture has to grant others the permission to tag the picture (tagging rights: permission-based) which results in so-called *narrow folksonomies* [160]. In contrast, the other systems listed in Table 3.3 do not impose these restrictions which allows for *broad folksonomies*.

TagMe! provides two tagging features that are currently not sufficiently implemented in other systems: spatial tagging and tag categorization. Flickr and also MediaWiki[19] platforms enable users to add notes or comments to specific areas within pictures. However, similarly to LabelMe, which allows users to attach keywords to arbitrarily formed shapes within an image, these systems do not provide means for tag-based navigation based on such spatial annotations, i.e. users cannot click on a spatial tag assignment to navigate to other resources that are related to the corresponding tag (and possibly to the area). TagMe! offers tag-based navigation, which is common in tagging systems

[19] `http://www.mediawiki.org`

such as Flickr and Delicious, also for spatial tag assignments. A further innovation of TagMe! is the tag categorization that is performed on the level of tag assignments and can therewith be used to disambiguate the meaning of a particular tag assignment. Delicious, on the contrary, only supports grouping of tags in so-called *tag bundles*. These tag bundles enable users to organize tags but do not help them to disambiguate specific tag assignments. They are moreover seldomly used: Tonkin reports that approx. 10% of the Delicious users have more than five tag bundles [207].

3.4.2 User Acceptance and Usage Patterns

We conducted an analysis to investigate whether users accept the additional tagging features of TagMe! and how they make use of these features. In particular, we target the following questions.

1. How are *categories* used in comparison to tags?
2. How do people make use of spatial tagging, i.e. assigning tags to specific *areas* within an image?
3. How accurate can tags (and categories) be mapped to *DBpedia URIs* describing the meaning of the annotations?

For answering the questions above we analyzed the tagging activities performed within the first month after launching the system. During this time period, 28 people (mainly PhD students in the area of computer science) were using the system for tagging their own Flickr pictures, pictures published by people they know as well as other pictures they were interested in.

Analysis of Category Usage and Benefits

Figure 3.13 shows the evolution of the number of distinct tags and categories. Although categories can be entered freely like tags, they grow much less than tags. Further, only 35 of the 123 distinct categories (e.g., "car" or "sea") have also been used as tags, which means that users seem to use different kinds of concepts for categories and tags respectively.

The TagMe! system supports users in assigning categories by means of auto-completion (see Figure 3.11). During our evaluation we divided the users into two groups: 50% of the users (*group A*) got only those categories as suggestion, which they themselves used before, while the other 50% of the users (*group B*) got categories as suggestions, which were created by themselves or by another user within their group. This small difference in the functionality had a significant impact on the alignment of the categories. The number of distinct categories in group A was growing stronger than in group B: users of group A created, on average, 0.21 new categories per category assignment while the

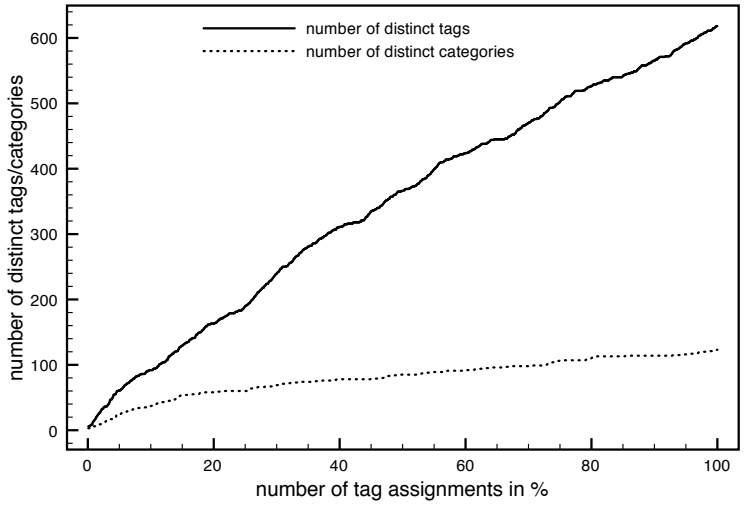


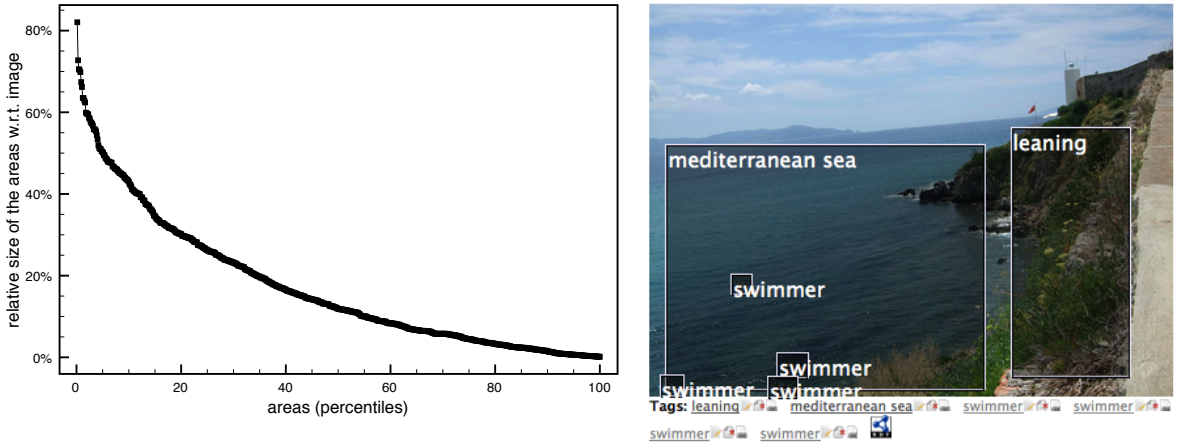
Figure 3.13: Growth of number of distinct tags in comparison to distinct categories.

other user group created introduced just 0.12 new categories per category assignment. Hence, the vocabulary of the categories can be aligned much better if categories, which have been applied by other users, are provided as suggestions as well.

Analysis of Spatial Tagging Information

Overall, we observed that users appreciated the feature of tagging specific areas within an image as 49.5% of the tag assignments were attached to a specific area. We also saw that the usage of *spatial tagging* helps to differentiate categories. For example, some categories have never or very seldomly been used when a specific area of an image was tagged (e.g., "time", "location", or "art") while others have been applied almost only for tagging a specific area (e.g., "people", "animals", or "things").

Figure 3.14(a) depicts the distribution of the size of the areas with respect to the size of the picture they are assigned to. Less than 10% of the spatial annotations cover more than 50% of the picture and more than 20% of the annotations cover less than 5% of the picture. Looking at these annotations in detail, reveals that the corresponding tags describe the main content of the pictures. By contrast, those annotations, which are relatively small and cover less than 5% of the picture, seem to be very specific and rather describe supplemental aspects visible in the pictures than the main content. For example, Figure 3.14(b) shows an image for which users annotated people, who are swimming in the sea. However, these people are hardly visible. Search and ranking algorithms might consider the size of the spatial annotations to adjust the rankings they produce. The photo in Figure 3.14(b) might, for example, be rather be appropriate for people, who are interested in the Mediterranean Sea, than for people, who are interested in pictures of swimmers.



(a) Distribution of area sizes.

(b) Area tagging example.

Figure 3.14: Spatial annotations: (a) the size of the area is specified with respect to the size of the tagged images, i.e. this fraction of an image that is covered by the area tag; (b) example of an image with spatial annotations.

Mapping to DBpedia URIs

For realizing the feature of mapping tags and categories to DBpedia [47] URIs we compared the following two strategies.

- **DBpedia Lookup** The naive lookup strategy invokes the DBpedia lookup service with the tag/category that should be mapped to a URI as search query. DBpedia ranks the returned URIs similarly to PageRank [63] and our naive mapping strategy simply assigns the top-ranked URI to the tag/category in order to define its meaning.
- **DBpedia Lookup + Feedback** The advanced mapping strategy is able to consider feedback while selecting an appropriate DBpedia URI. Whenever a tag/category is assigned, for which already a correctly validated DBpedia URI exists in the TagMe! database then that URI is selected. Otherwise the strategy falls back the naive DBpedia Lookup.

As depicted in Figure 3.15, the mappings of the naive approach result in a precision of 79.92% for mapping tags to DBpedia URIs and 84.94% for mapping categories considering only those tag assignments where a DBpedia URI that describes the meaning properly exists. The consideration of feedback, which is currently managed by the administrators of TagMe!, improves the precisions of the naive DBpedia Lookup clearly to 86.85% and 93.77% respectively, which corresponds to an improvement of 8.7% and 10.4%. As the mapping accuracy for categories is higher than the one for tags, it seems that the identification of meaningful URIs for categories is easier than for tags. Moreover, the precision of the category mappings, which are determined by the strategy that

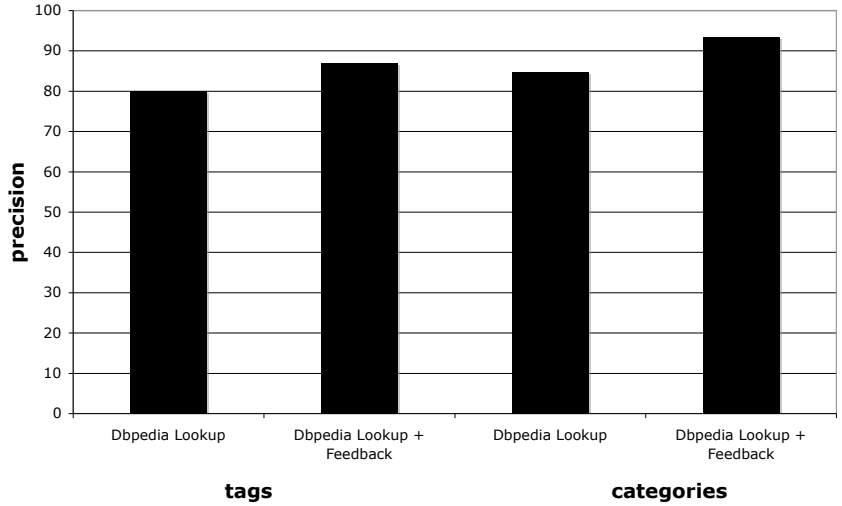


Figure 3.15: Precision of mapping tags and categories to DBpedia URI.

incorporates feedback, will further improve, because—fostered by TagMe!'s category suggestion feature—the number of distinct categories seems to converge (cf. Figure 3.13).

The results of the DBpedia mapping are very encouraging. Further, the mapping strategies itself can be enhanced by also considering the context of the tag/category that should be mapped. For example, for mapping a tag assignment one could select the DBpedia URI, which best fits to the DBpedia URI of the category that is associated to the tag assignment. Overall, the DBpedia mapping reduces the number of distinct tags and categories within TagMe! by 14.1% and 20.9% respectively, which promises a positive impact on the recall when executing tag-based search. For example, while some users assigned the tag "car" to pictures showing cars other users chose "auto" to annotate other pictures that show cars. As both kinds of tag assignments are mapped to "http://dbpedia.org/resource/Automobile", TagMe! can simply search via the DBpedia URI whenever users search via "car" or "auto" to increase recall of the tag-based search operations. In Section 4.2 we will present a ranking algorithm that exploits these characteristics of the meaningful URIs.

Conclusions of Usage Analysis

In summary, our usage analysis reveals already some benefits of the tagging features provided by TagMe!: the contextual information available in the TagMe! folksonomy have a positive impact on identifying correlations between the folksonomy entities (e.g., identifying similar tags). Further, categories and areas enable the extraction of additional semantic relations between tags. As tags are mapped to DBpedia URIs that describe the meaning of a tag assignment, it is possible to deduce rich semantics from the context folksonomy available in TagMe!. Given the results of our study, we can summarize the answers to the questions raised at the beginning of this section as follows.

- The usage of categories differs from the usage of tags: even for those users, who did not benefit from the category suggestions, the number of distinct categories is growing slower than the number of distinct tags.
- The spatial tagging feature was adopted by the users. 49.5% of the tag assignments were enriched with spatial context information. Moreover, we observed that the size of the spatial tag assignments might help to identify those tags that rather describe the main content of an image and those tags that refer to supplemental aspects of an image.
- A naive DBpedia lookup allows us to map tags and categories to ontological concepts (DBpedia URIs) with a high precision of 79.92% (tags) and 84.94% (categories). The consideration of feedback improves the accuracy of the mapping of tags and categories to 86.85% and 93.77% respectively.

This preliminary analysis already delivers insights into potential benefits of having the context information available in the TagMe! folksonomy. In Chapter 4 we will present corresponding algorithms that exploit such contextual information and the evaluation of these algorithms will finally show that categories, URIs and spatial data attached to tag assignments improves search (see Chapter 4).

3.5 Discussion

Traditional social tagging systems such as Flickr, Delicious, Last.fm, or BibSonomy model the users' tagging activities by means of user-tag-resource triples enriched with a timestamp that indicates when a *user* assigned a specific *tag* to a *resource*. In this chapter we introduced a generic context folksonomy model that extends the common folksonomy model [124] and allows for enriching tag assignments with arbitrary context information.

Further, we presented two social tagging systems that have been developed as part of this thesis: GroupMe! and TagMe!. Both systems follow the context folksonomy model defined in Section 3.2. Moreover, these systems demonstrate how semantically meaningful context information can be deduced from tagging activities. GroupMe! introduces group structures to folksonomies: tagging activities are performed in context of a group of resources. TagMe! enables users to attach context information to tag assignments explicitly: spatial information describes to which part of a resource a tag refers and categories allow for classifying tag assignments. Moreover, tag assignments are automatically enriched with URIs that describe the meaning of tags and categories.

GroupMe! and TagMe! foster interoperability between social tagging systems as folksonomy data is described semantically using Semantic Web standards and accessible according to the principles of Linked Data [56]. With GroupMe!'s RESTful Semantic Web interface [25] we furthermore propose how the main HTTP methods [95] (GET,

POST, PUT, and DELETE) should be applied to enable third-party applications to read, add, modify, and delete data in social resource sharing systems.

Our main contributions and findings presented in this chapter can be summarized as follows.

- We propose a generic context folksonomy model [3, 19] (see Section 3.2).
- We developed GroupMe!, a social tagging system for organizing and sharing Web resources in groups [10] (see Section 3.3).
  - The usage analysis showed that the GroupMe! application and the grouping functionality is highly appreciated by end-users.
  - GroupMe! – one of the top five projects at the International Semantic Web challenge 2007 [11] – is a Social Semantic Web showcase application: user-contributed content is accessible and maintainable via RESTful Semantic Web interfaces.
- We developed TagMe!, a tagging and exploration front-end for Flickr, that allows for enriching tag assignments with spatial information, categories and DBpedia URIs [35] (see Section 3.4).
  - The usage analysis revealed that TagMe! users adopt the additional tagging features and that URIs that define the semantic meaning of tag assignments can automatically be identified with high precision.
  - Categories and spatial information allow for the deduction of valuable semantics from tags.
- Both applications demonstrate how social tagging systems can apply the context folksonomy model and generate contextual information and valuable semantics.

In the following chapters we will present different experiments that reveal the benefits of these context models for non-personalized (see Chapter 4) as well as personalized (see Chapter 5) retrieval of information in social tagging systems.

4 Context-based Search and Ranking in Folksonomy Systems

Based on the models defined in Chapter 3 that capture contextual information related to tagging activities, we present ranking algorithms that exploit these context structures. We apply these algorithms for search and ranking and present different experiments where we evaluate and compare the search performance with respect to algorithms presented in related work. The main contributions of this chapter have been published in [1, 3, 4, 19, 23, 26, 27, 28].

4.1 Introduction: Context-based Search and Ranking in Folksonomies

Today, more than 1 billion people are using the Internet[1] and perform several hundreds million keyword-based search queries at search engines such as Google [116]. Ranking is an important technique for Web search engines, because it allows for ordering the documents according to a given query and therewith supports the user in finding relevant documents. Ranking further supports various other applications. For example, recommender systems rank items to identify a list of *top k items* that will be proposed to the user [195] and spam detection approaches such as MailRank [82] or TrustRank [109] apply ranking to detect malicious users (email addresses) and Web pages respectively. Correspondingly, information retrieval applications within the scope of folksonomy systems also benefit from ranking to allow for search [51, 84, 123], recommendations [20, 128, 197, 199] or spam detection [147].

A basic assumption of folksonomy-based search and ranking algorithms is that tags describe the content of resources well. Li et al. prove this assumption by comparing the actual content of Web pages with tags assigned to these sites in the Delicious system [155]. However, Bischoff et al. observe that not all tags can be used for search [61]. Hence, interpreting folksonomy structures in the right way is the core challenge of folksonomy-based search and ranking. Hotho et al. proposed FolkRank (see Section 2.2.2), a ranking algorithm that adapts PageRank [176] to rank folksonomy entities (users, tags, and resources), and presented a qualitative discussion concerning the search and ranking

[1] `http://www.internetworldstats.com/stats.htm` (statistics published on June 30th 2010)

performance of FolkRank [124]. Bao et al. also followed the PageRank paradigm. They developed SocialPageRank (see Section 2.2.2) to rank resources and showed that Web search can be improved by exploiting knowledge embodied in folksonomies [51]. Both algorithms operate on the folksonomy model specified in Definition 2.1 and thus neither FolkRank nor SocialPageRank exploit contextual information.

In this chapter, we will introduce a framework of ranking algorithms that make use of context information available in folksonomies. We verify the utility of this framework of context-sensitive ranking algorithms in different application contexts. In Chapter 5 we apply the algorithms for personalized search and generating recommendations. In the subsequent sections we will focus on non-personalized search and conduct different experiments that reveal—by means of standard information retrieval metrics—that our context-based algorithms improve the search performance of existing approaches significantly. In particular, we specify the search task as follows.

Problem 1 (Search and Ranking) *Given a tag as a query, the task of the search and ranking algorithm is to select resources relevant to the query and put these resources into an order so that the resources, which are most relevant to the query, appear at the very top of the ranking.*

The ranking algorithms thus have to compute a ranking according to the given query which requires the ranking algorithms to be topic-sensitive. Non-topic-sensitive algorithms such as SocialPageRank [51] would not succeed in this task, because they cannot select resources according to a given topic and query in particular. Therefore, we specify a second challenge to be solved by the ranking algorithms: re-ranking search results.

Problem 2 (Re-Ranking Search Results) *Given a base set of possibly relevant resources, the task of the ranking algorithm is to put these resources into an order so that the most relevant resources appear at the very top of the ranking.*

By re-ranking search results one can improve search result rankings. For example, Joachims et al. show that the consideration of click-through data from search engine log files improves Web search [132] and Yan et al. conducted a user study in which they discovered similar improvements for tag-based search [217]. In this chapter, our algorithms perform re-ranking without requiring feedback from search engine log files but exploit the folksonomy structure as well as inherent contextual semantics.

The research questions that we will investigate in this chapter can be summarized as follows.

- How to design ranking algorithms that exploit context information available in folksonomies?
- Which ranking algorithms perform best for search in folksonomies (see Problem 1 and Problem 2)?

- How does the exploitation of context information available in folksonomies impact the search performance?

In Section 4.2 we will first introduce the ranking algorithms we developed in this thesis. In Section 4.3 we will then evaluate the group-sensitive ranking algorithms while the evaluations of the other context-based algorithms are presented in Section 4.4. We conclude with a short summary and discussion of our main findings.

4.2 Context-based Ranking Algorithms

The usage context of social tagging that is explicitly captured in GroupMe! or TagMe! provides additional features that can be exploited by ranking algorithms to enhance search. Further, information about the characteristics of a system, in which tag assignments have been performed, allow for ranking algorithms that consider these specifics. In the following we will introduce ranking algorithms that make use of such contextual information. These algorithms can be categorized according to the folksonomy model they operate on, which is either the traditional folksonomy model without explicit context information (see Subsection 4.2.1), the group context folksonomy model (see Subsection 4.2.2) or the generic context folksonomy model (see Subsection 4.2.3).

4.2.1 Ranking in traditional folksonomies

We first present two algorithms that are based on the traditional folksonomy model. With SocialHITS, we outline how folksonomy structures in combination with knowledge about the folksonomy system design [158] can be interpreted in order to construct directed folksonomy graphs, on which the HITS algorithm can be executed. Further, we present a topic-sensitive alternative to SocialPageRank.

SocialHITS

Kleinberg's HITS algorithm [139] (cf. Section 2.2.2) expects a directed graph G as input. G is a partial Web graph consisting of linked resources that are possibly relevant to a certain topic. For each resource in the graph, the algorithm computes an authority (see Equation 2.2) and hub (see Equation 2.3) score that is based on the incoming and outgoing links of the resource. As the folksonomy graph $\mathbb{G}_{\mathbb{F}}$ is undirected, the challenge of applying HITS to folksonomies is to transform $\mathbb{G}_{\mathbb{F}}$ into a directed graph. Wu et al. propose to create two directed edges from a given tag assignment $(u, t, r) \in Y$ (cf. Folksonomy model, Definition 2.1): *"$u \rightarrow t$"* and *"$t \rightarrow r$"* [213]. However, this restricts the role of hubs and authorities to certain entities. For example, by following this *naive HITS* strategy, resources have no outgoing links so that the hub score of resources becomes 0.

Authorities	
user	a high quality user annotates high quality resources before other users annotate them
tag	is assigned by high quality users
resource	(1) is tagged by high quality users with high quality tags (2) is contained in high quality groups
Hubs	
user	has annotated high quality resources and utilized high quality tags
tag	is assigned to high quality resources
resource	(1) is tagged with tags of high quality resources (2) is contained in groups with high quality resources

Table 4.1: Characteristics of authority/hub users, tags, and resources. The resource characteristics that consider (2) *high quality groups* is only applicable to group context folksonomies (see Definition 3.2.)

We introduce *SocialHITS*, a more sophisticated HITS application that considers the design of the folksonomy system and its user interface in particular when creating the *directed folksonomy graph*. In the GroupMe! system, for example, a resource r_h can be interpreted as a hub of a tag t_a assigned to r_h because each resource displays its tag cloud, whereas in tagging systems that do not show the tags of resources it is not possible to draw that conclusion (cf. *tagging support: "viewable" vs. "blind"* in [158]).

Table 4.1 lists some of the characteristics of users, tags, and resources that indicate when they should be considered as authorities and hubs respectively. Some of these characteristics can be deduced from the traditional folksonomy model (see Definition 2.1) while others require additional context information, e.g. regarding user entities, edges representing some user characteristics can be constructed as follows.

- **hub users** For all resources r a user u has annotated with a tag t we can construct edges *"*$u \rightarrow t$*"* and *"*$u \rightarrow r$*"*. The required information is thus contained in the tag assignments.
- **authority users** According to Table 4.1 an authoritative user u_a can also be characterized by the fact that other users have annotated resources, that u_a has annotated before the other users annotated them. Therefore, the timestamp of tag assignments has to be evaluated so that we can construct an edge *"*$u_h \rightarrow u_a$*"* whenever another user u_h has annotated a resource that was already tagged by u_a.

Having an appropriate strategy for constructing the directed folksonomy graph, which serves as input to the core HITS iteration (see Definition 2.4), SocialHITS can be defined as follows.

Definition 4.1 (SocialHITS) *The* SocialHITS *algorithm computes hub and authority values for arbitrary folksonomy entities.*

1. ***Input:*** *folksonomy $\mathbb{F}$, topic t, search strategy s_t, graph construction strategy s_g, and the number of HITS iterations k to perform*
2. $\mathbb{F}_t \leftarrow$ *apply s_t to $\mathbb{F}$ in order to search for entities and tag assignments relevant to t*
3. $\mathbb{G}_D \leftarrow$ *apply s_g to $\mathbb{F}_t$*
4. $(x_k, y_k) \leftarrow$ *iterate($\mathbb{G}_\mathbb{D}, k$)*
5. ***Output:*** *the vectors x_k and y_k containing the authority and hub values of the entities in $\mathbb{F}_t$*

In our evaluations we experimented with different search strategies s_t to determine the set of possibly relevant resources. For example, in group context folksonomies we applied the so-called GRank algorithm (see Section 4.2.2) for search and utilized the sum of authority and hub score to rank the folksonomy entities.

Personalized SocialPageRank

SocialPageRank [51] (cf. Section 2.2.2) computes a global ranking of resources in folksonomies. With the *Personalized SocialPageRank* algorithm we extend SocialPageRank and transform it into a topic-sensitive ranking algorithm (cf. [112]). Therefore, we introduce the ability of emphasizing weights within the input matrices of SocialPageRank so that preferences can be considered which are possibly adapted to a certain context. For example, $w(t, r)$ is adapted as follows: $w(t, r) = pref(t) \cdot pref(r) \cdot |\{u \in U : (u, t, r) \in Y\}|$, where $pref(\cdot)$ returns the preference score of t and r respectively. The preference function $pref(\cdot)$ is specified in Equation 4.1:

$$pref(x) = \begin{cases} 1, \text{if there is no preference in } x \\ c > 1, \text{if there is a preference in } x \end{cases} \tag{4.1}$$

In contrast to Jeh and Widom who propose to make use of so-called *personalized PageRank vectors* [130], which specify preferences in certain Web resources, we also allow for specifying preferences in resources with respect to certain tags. In our evaluations in Chapter 4 we utilized the Personalized SocialPageRank in order to align the SocialPageRank to the context of a keyword query t_q and specified a preference into t_q using $c = 20$.

4.2.2 Ranking in group context folksonomies

Group context folksonomies (see Definition 3.2) describe in which context a tag assignment was performed where the context is given by a group of resources. In this subsection

we present ranking algorithms that exploit group structures to rank folksonomy entities (users, tags, and resources).

The first set of algorithms is based on FolkRank (see Section 2.2.2), which does not make use of the additional structure available in group context folksonomies (see Definition 3.2). In order to make the FolkRank algorithm aware of the group context gained by folksonomy systems such as GroupMe! (see Section 3.3), we adapt the process of constructing the graph $\mathbb{G}_{\mathbb{F}}$ (see Definition 2.2) from the hypergraph formed by the tag assignments which provide a group context.

GFolkRank

GFolkRank interprets groups as artificial, unique tags. If a user u adds a resource r to a group g then GFolkRank interprets this as a tag assignment (u, t_g, r, ε), where $t_g \in T_G$ is the artificial tag that identifies the group. The folksonomy graph $\mathbb{G}_{\mathbb{F}}$ is extended with additional vertices and edges. The set of vertices is expanded with the set of artificial tags T_G: $V_{\mathbb{G}} = V_{\mathbb{F}} \cup T_G$. Furthermore, the set of edges $E_{\mathbb{F}}$ is augmented by $E_{\mathbb{G}} = E_{\mathbb{F}} \cup \{\{u, t_g\}, \{t_g, r\}, \{u, r\} | u \in U, t_g \in T_G, r \in \breve{R}, u \text{ has added } r \text{ to group } g\}$. The new edges are weighted with a constant value w_c as a resource is usually added only once to a certain group. We suggest to set $w_c = max(|w(t, r)|)$ because we believe that grouping a resource is, in general, more valuable than tagging it. GFolkRank is consequently the FolkRank algorithm (cf. Section 2.2.2), which operates on basis of $\mathbb{G}_{\mathbb{G}} = (V_{\mathbb{G}}, E_{\mathbb{G}})$.

CFolkRank

If users assign a certain tag to resources in context of different groups then the meaning of the tag may differ. CFolkRank attaches the group *context* of tag assignments to the tags. In particular, CFolkRank replaces every tag t with a tag t_{tg}, which indicates that tag t was used in group g. It then transforms all GroupMe! tag assignments into traditional tag assignments. For example, the GroupMe! tag assignment (u_1, t_2, r_2, g_1), presented in Figure 3.2(a), is interpreted as $(u_1, t_{t_2 g_1}, r_2)$ $(= tas_1)$. Assume we also have a tag assignment (u_1, t_2, r_2, g_2) then this would be converted into $(u_2, t_{t_2 g_2}, r_2)$ $(= tas_2)$. Thus, a 3-uniform hypergraph is built, which serves as input for the construction of the folksonomy graph $\mathbb{G}_{\mathbb{C}}$. The construction of $\mathbb{G}_{\mathbb{C}}$ is done as in the normal FolkRank algorithm, described in Section 2.2.2. Detecting equality of tags differs from FolkRank and GFolkRank, e.g. given tas_1 and tas_2 from above, the weight $w(u_1, t_{t_2 g_1})$ is not only determined by tas_1 but also partially by tas_2, although the tag $t_{t_2 g_1}$ in tas_1 is not exactly equal to $t_{t_2 g_2}$ in tas_2. We compute the similarity between two tags $t_{t_x g_y}$ and $t_{t_v g_w}$ and therewith the influence of a tag assignment to a weight as depicted in Table 4.2. Hence, based on tas_1 and tas_2 it is $w(u_1, t_{t_2 g_1}) = 1.4$. As part of the traditional FolkRank iterations, these weights are normalized so that the sum of weights in the corresponding rows in the adjacency matrix is equal to 1.

$\wedge$	$t_x = t_v$	$t_x \neq t_v$
$g_y = g_w$	1.0	0.2
$g_y \neq g_w$	0.4	0

Table 4.2: Example:computing weights with CFolkRank

(G/C)FolkRank and Tag Propagation

In addition to GFolkRank and CFolkRank, we present two further extensions that help to exploit GroupMe! folksonomies. They can be applied to GFolkRank, CFolkRank, and FolkRank as well. The core idea of both extensions is to propagate tags assigned to one resource/group to other resources/groups. Such techniques synthetically increase the amount of input data and do not require to change the algorithms described above substantially.

(G/C)FolkRank$^+$ – Propagation of Group Tags. GroupMe! users annotate groups about 1.75 times more often than common resources [22]. By propagating tags which have been assigned to a group (*group tags*) to its resources we try to counteract this situation. For example in Figure 3.2(a), tag t_2, which is assigned to group g_2, can be propagated to all resources contained in g_2. An obvious benefit of this procedure is that untagged resources like r_3 obtain tag assignments (here: (u_2, t_2, r_3, g_2)). In order to adjust the influence of inherited tag assignments, we weight these assignments by a dampen factor $df \in [0, 1]$. In our evaluations in Chapter 4 we set $df = 0.2$. FolkRank$^+$, GFolkRank$^+$, and CFolkRank$^+$ denote the strategies that make use of group tag propagation.

(G/C)FolkRank^{++} – Propagation of all Tags. Tags can correspondingly be propagated among resources that are contained in the same group. This extension induces propagation of (i) group tags to resources within the group, (ii) resource tags of one resource to other resources within a group, and (iii) resource tags to the group itself. Propagation is damped with factor *df* from above. Note that only tag assignments that have been carried out within the context of the corresponding group are considered for propagation. FolkRank^{++}, GFolkRank^{++}, and CFolkRank^{++} denote the algorithms that propagate all tags.

GRank

With GRank we propose a search and ranking algorithm specialized on group context folksonomies. GRank is specified in Definition 4.2.

Definition 4.2 (GRank) *The* GRank *algorithm computes a ranking for all resources, which are related to a tag t_q with respect to the group structure of group context folksonomies (see Definition 3.2). It executes the following steps:*

1. ***Input:*** *keyword query tag t_q.*
2. $\breve{R}_q = \breve{R}_a \cup \breve{R}_b \cup \breve{R}_c \cup \breve{R}_d$, *where:*
   a) $\breve{R}_a$ *contains resources* $r \in \breve{R}$ *with* $w(t_q, r) > 0$
   b) $\breve{R}_b$ *contains resources* $r \in \breve{R}$, *which are contained in a group* $g \in G$ *with* $w(t_q, g) > 0$
   c) $\breve{R}_c$ *contains resources* $r \in \breve{R}$ *that are contained in a group* $g \in G$, *which contains at least one resource* $r' \in \breve{R}$ *with* $w(t_q, r') > 0$ *and* $r \neq r'$
   d) $\breve{R}_d$ *contains groups* $g \in G$ *containing resources* $r' \in \breve{R}$ *with* $w(t_q, r') > 0$
3. $\vec{w}_{\breve{R}_q}$ *is the ranking vector of size* $|\breve{R}_q|$, *where* $\vec{w}_{\breve{R}_q}(r)$ *returns the* GRank *of resource* $r \in \breve{R}_q$
4. ***for each*** $r \in \breve{R}_q$ ***do:***
   (a) $\vec{w}_{\breve{R}_q}(r) = w(t_q, r) \cdot d_a$
   (b) ***for each*** *group* $g \in G \cap \breve{R}_a$ ***do:***
   $\vec{w}_{\breve{R}_q}(r) \mathrel{+}= w(t_q, g) \cdot d_b$
   (c) ***for each*** $r' \in \breve{R}_a$ *where* r' *is contained in a same group as* r *and* $r \neq r'$ ***do:***
   $\vec{w}_{\breve{R}_q}(r) \mathrel{+}= w(t_q, r') \cdot d_c$
   (d) ***if***$(r \in G)$ ***then:***
   for each $r' \in \breve{R}_a$ *where* r' *is contained in* r ***do:***
   $\vec{w}_{\breve{R}_q}(r) \mathrel{+}= w(t_q, r') \cdot d_d$
5. ***Output:*** *GRank vector* $\vec{w}_{\breve{R}_q}$

$w(t_q, r)$ counts the number of users, who have annotated resource $r \in \breve{R}$ with tag $t_q$in any group. When dealing with multi-keyword queries, GRank accumulates the different GRank vectors. The factors d_a, d_b, d_c, and d_d allow to emphasize the weights gained by (a) directly assigned tags, (b) tags assigned to a group the resource is contained in, (c) tags assigned to neighboring resources, and (d) tags assigned to resources of a group. In our evaluations we saw that direct tag assignment relations (d_a) should be weighted stronger than the other relations and that tags from neighboring resources (d_c) are least

related. Therefore, $d_a = 10$, $d_b = 4$, $d_c = 2$, and $d_d = 4$ are reasonable selections. In Section 4.3 we will furthermore optimize the adjustment of these parameters.

4.2.3 Ranking in context folksonomies

In this section we introduce three ranking algorithms that make use of contextual information provided in context folksonomies (see Definition 3.3). In particular, we present FolkRank-based algorithms that exploit categories, spatial information, and URIs which are attached to tag assignments performed in TagMe! (see Section 3.4).

Category-based FolkRank

The category-based FolkRank algorithm operates on a context folksonomy (see Definition 3.3) where the context is given by categories that are attached to tag assignments. The algorithm relates folksonomy entities via the category assignments and the main hypothesis is that entities sharing the same category are related to each other. Similarly to GFolkRank, the category-based FolkRank introduces an alternative approach for creating the weighted folksonomy graph $\mathbb{G}_{\mathbb{F}}$ (cf. Section 2.2.2). Categories are treated as tags ($c \in T_C$ where $T_C \subseteq T$) so that the set of nodes is extended with T_C: $V_{\mathbb{F}_{new}} = V_{\mathbb{F}} \cup T_C$. For each category assignment $(y, c) \in Z$, new edges are created to connect the given category c with the resource and tag of the tag assignment y: $E_{\mathbb{F}_{new}} = E_{\mathbb{F}} \cup \{\{c, r\}, \{c, t\} | c \in T_C, t \in T, r \in R, ((u, t, r), c) \in Z\}$. The weight of an edge (c, r) corresponds to the frequency the category c is assigned to a tag assignment that refers to r: $w(c, r) = |\{(u, t, r) \in Y : (u, t, r) \in Y, ((u, t, r), c) \in Z\}|$. Weights of (c, t)-edges are accordingly computed by counting the tag assignments that refer to t and are categorized using c.

Area-based FolkRank

While the categories are used to enrich the folksonomy graph with further edges and possibly also with further vertices, the area-based FolkRank merely modifies the weights of edges in $\mathbb{G}_{\mathbb{F}}$ (cf. Section 2.2.2). In particular, it emphasizes the weight of an edge between a tag t and a resource r (i.e. (t, r)-edges) whenever t and r occur within a tag assignment $(u, t, r) \in Y$ to which spatial context information is attached to. The amplification is based on the size of the corresponding area as well as on the distance of the midpoint of the area to the center of the resource (in our experiments we examine pictures).

Size. Our hypothesis is that the larger the size of an area the more important is also the corresponding tag for the given resource, i.e. the larger the area that is attached to $(u, t, r) \in Y$ is the more relevant t is for r. The size of an area is measured

relatively to the size of the resource. For example, if an area is associated to a tag assignment (u, t, r) and the relative size of the area is $s = 0.4$, i.e. the area covers 40% of the resource, then we use s^{-1} to emphasize the weight $w(t, r)$. As different users might attach differently sized areas to (u, t, r), we use the average size $\bar{s}$ of those areas to finally compute the new weight of (t, r)-edges: $w_s(t, r) = \bar{s}^{-1} \cdot w(t, r)$.

Distance. The second hypothesis is that tag assignments which are according to the spatial information relevant to the center of a resource are more important for the resource than tag assignments which are associated to the margin of a resource. The distance d from the center of the area to the center of the resource is also measured relatively and the weight $w(t, r)$ is emphasized with the average distance $\bar{d}$ of the areas attached to $(u, t, r) \in Y$: $w_d(t, r) = \bar{d}^{-1} \cdot w(t, r)$.

Finally, the weight of the edges (t, r) is simply the average of $w_s(t, r)$ and $w_d(t, r)$: $w_{area}(t, r) = 0.5 \cdot w_s(t, r) + 0.5 \cdot w_d(t, r)$.

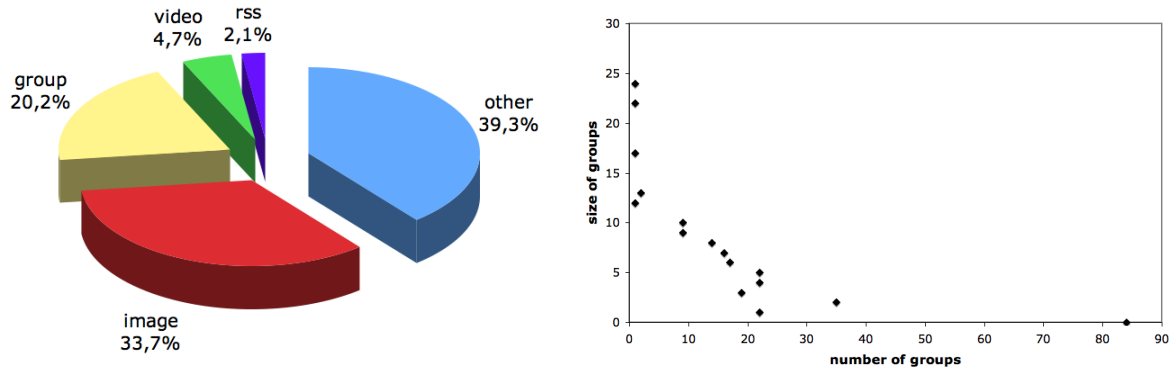
URI-based FolkRank

The URI-based FolkRank operates on meaningful URIs instead of tags. Hence, the construction of the folksonomy graph $\mathbb{G}_{\mathbb{F}} = (V_{\mathbb{F}}, E_{\mathbb{F}})$ is modified as follows. The set of vertices is $V_{\mathbb{F}} = U \cup URI \cup R$, where $URI \subseteq C$ (cf. Definition 3.3) denotes the set of URIs that describe the meaning of the tag assignments. The set of edges is $E_{\mathbb{F}} = \{\{u, uri\}, \{uri, r\}, \{u, r\} | u \in U, uri \in URI, r \in R, ((u, t, r), uri) \in Z\}$ whereas there should only exist exactly one URI assignment $(y, uri) \in Z$ for each tag assignment y. The weights of the edges are computed in the same way as done by the traditional FolkRank algorithm.

The URI-based FolkRank algorithm is resistant against ambiguous tags as well as synonymic tags. For example, given two tag assignments $y_1 = (u_1, t_1, r_1)$ and $y_2 = (u_2, t_2, r_2)$ as well as two context assignments (y_1, uri_1) and (y_2, uri_1), the URI-based FolkRank algorithm would replace the synonymic tags t_1 and t_2 by the unique URI uri_1 that clearly defines the meaning of the tags. It therewith, e.g., relates r_1 and r_2 as it constructs the edges (uri_1, r_1) and (uri_1, r_2). As the TagMe! system utilizes DBpedia URIs to define the meaning of tags, we denote the URI-based FolkRank as *DBpedia FolkRank* in our search experiments in Chapter 4.

4.3 Evaluation of group-sensitive Ranking Algorithms

In this section we evaluate the search performance of the ranking strategies in a group context folksonomy setting. In particular, we evaluate the algorithms on a dataset gained in the GroupMe! system. We conduct search and ranking experiments as well as re-ranking experiments. For the re-ranking experiments we further investigate which strategy is best for detecting the base set of search results to be re-ranked.



(a) Distribution of the media types of the resources (b) Distribution of the number of resources per group

Figure 4.1: GroupMe! characteristics

4.3.1 Dataset Characteristics and Ground Truth

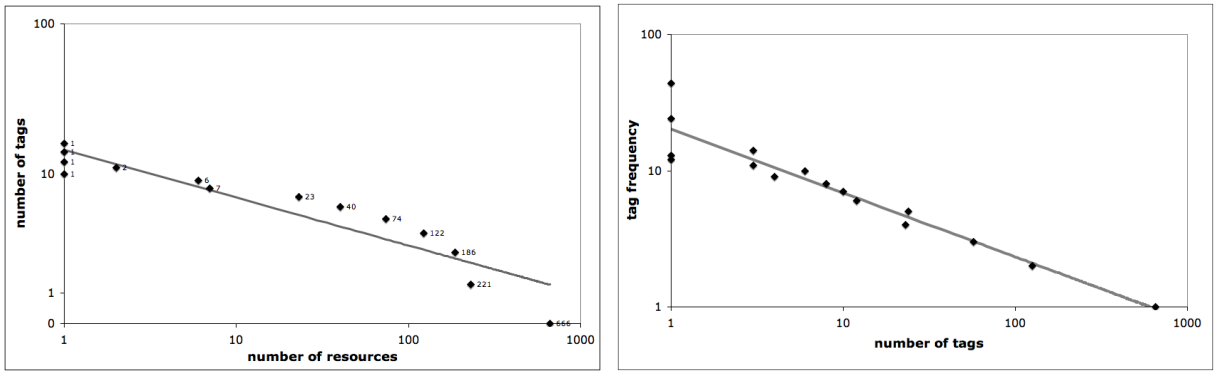
For evaluating the group-sensitive ranking algorithms with respect to the search tasks defined above, we used a snapshot of the folksonomy data gained by the GroupMe! system (see Section 3.3).

Dataset Characteristics

The search and ranking performance was measured on the GroupMe! folksonomy that was created by the community within a period of six months. Overall, the dataset contains 234 users, 974 tags, 1351 resources, 273 groups, and 1758 tag assignments. In the given dataset, 49.3% of the resources do not have any tag assignment.

Figure 4.1(a) shows the distribution of media types among the resources which conforms to the distribution of the complete GroupMe! folksonomy (cf. Section 3.3.4). According to the group context folksonomy model (see Definition 3.2) that is implemented in GroupMe!, groups are treated as resources and can thus also be added to GroupMe! groups. For the given dataset, people made use of this feature quite extensively as 20.2% of the grouped resources are groups themselves. The remaining 79.8% of resources are Web resources, which divide equally into multimedia resources such as videos, images, or RSS feeds (40.5% of all resources) and other Web sites or bookmarks (39.3% of all resources). This balanced distribution among the different media types, which is also present within the groups, is an evidence that users make use of the GroupMe! feature to bundle resources of different media types together.

Having a closer look at the groups, we checked the distribution of the number of resources in a group (see Figure 4.1(b)). 94.89% of the groups contain less than 10 resources. This can be explained by the user interface, which limits the space of a group against the screen size. Therefore, only a limited number of resources can be placed in a group in a



(a) Distribution of the number of tags per resource (b) Distribution of the tag frequency

Figure 4.2: Tagging characteristics

way that all resources are visible and accessible.

The GroupMe! folksonomy follows similar characteristics that also occur in other folksonomy systems such as Flickr [199]. For example, in Figure 4.2(a) we measured the distribution of tag assignments per resource. The maximum number of tags that are assigned to a single resource is 15, which is lower than in other systems. However, on a logarithmic scale the scatterplot shows a distribution that reminds of a *Power Law* distribution [200] which is observed in other folksonomy systems such as Flickr [199] or Delicious [77, 88, 110] as well. For the distribution of tags, we also measured how often a tag was used by the users (see Figure 4.2(b)). The most popular tag is *semantic web*, being used 44 times, the next most frequent tag was used 24 times. Together with the distribution of the number of tags per resource, we deduce that the GroupMe! folksonomy – even it is much smaller than folksonomies evolved in other systems – bears similar characteristics like other common folksonomy systems. In Chapter 5 we will see that results gained on the GroupMe! dataset can be confirmed on a larger Flickr dataset as well.

Ground Truth

Our evaluations are based on 50 hand-selected rankings from which we gained an optimal ranking of search results for different queries. Given 10 query keywords q, which were also used as tags ($q \in T$), and the entire GroupMe! data set, 5 experts independently created rankings for each of the keywords. For a given keyword, the participants had to select and rank these 20 resources which represented from their perspective the most precise top 20 ranking. The agreement of the participants was very high: the overlap ($OSim$) between the hand-selected rankings created by the different experts for a given query was, on average, higher than 90%. By building the average ranking for each keyword, we gained 10 optimal rankings. Among the 10 keywords, there are frequently used tags like *"web"* as well as seldom used ones such as *"beer"*.

4.3.2 Search and Ranking Experiment

In our first set of experiments the ranking algorithms have to execute the *search and ranking task* specified in Problem 1 on the dataset described in the previous section. Hence, the algorithms have to compute select and rank GroupMe! resources for each of the given query for which there exist a ground truth of *optimal rankings*. The success of achieving the goal of this task is measured by comparing the ranking generated by the algorithms with the optimal rankings. For a pairwise comparison of such two rankings there exist metrics that measure the correlation of the ranking lists [93]. In order to measure the quality of rankings with respect to an optimal ranking we thus used the *OSim* and *KSim* metrics as proposed in [112]. $OSim(\tau_1, \tau_2)$ enables us to determine the overlap between the top k entities of two rankings, τ_1 and τ_2.

Definition 4.3 (OSim) *OSim, the* overlapping similarity*, measures the overlap between two ranking lists.*

$$OSim(\tau_1, \tau_2) = \frac{|E_1 \cap E_2|}{k} \tag{4.2}$$

where $E_1, E_2 \subseteq E$ are the sets of entities that are contained in the top k of ranking τ_1 and τ_2 respectively, and $|E_1| = |E_2| = k$.

In general, E is again the set of all folksonomy entities, i.e. given Definition 2.1 it is $E = U \cup T \cup R$. For the problem of search for resources, E is restricted to the set of resources. OSim measures the similarity of two ranked lists without considering the order of the entities within the lists. In contrast, $KSim(\tau_1, \tau_2)$, which is based on Kendall's τ distance measure [137], indicates the degree of pairwise distinct entities, e_u and e_v, within the top k that have the same relative order in both rankings.

Definition 4.4 (KSim) *KSim is the fraction of entities e_u and e_v, within the top k that have the same relative order in both rankings τ_1 and τ_2.*

$$KSim(\tau_1, \tau_2) = \frac{|\{(e_u, e_v) : \tau_1', \tau_2' \text{ agree on order of } (e_u, e_v), e_u \neq e_v\}|}{|E_{\tau_1 \cup \tau_2}| * (|E_{\tau_1 \cup \tau_2}| - 1)} \tag{4.3}$$

$E_{\tau_1 \cup \tau_2}$ is the union of entities of both top k rankings. τ_1' corresponds to ranking τ_1 extended with entities E_1' that are contained in the top k of τ_2 and not contained in τ_1. The order of these entities $e \in E_1'$ is not further specified. τ_2' is constructed correspondingly.

Together, *OSim* and *KSim* are suited to measure the quality of a ranking with respect to an optimal (possibly hand-selected) ranking. The higher OSim and KSim scores are the more successful the ranking algorithms.

Figure 4.3 gives an overview on the measured results for the group-sensitive ranking strategies introduced in Section 4.2.2 with respect to OSim and KSim metrics and compares the performance with these ranking strategies from related work which do not

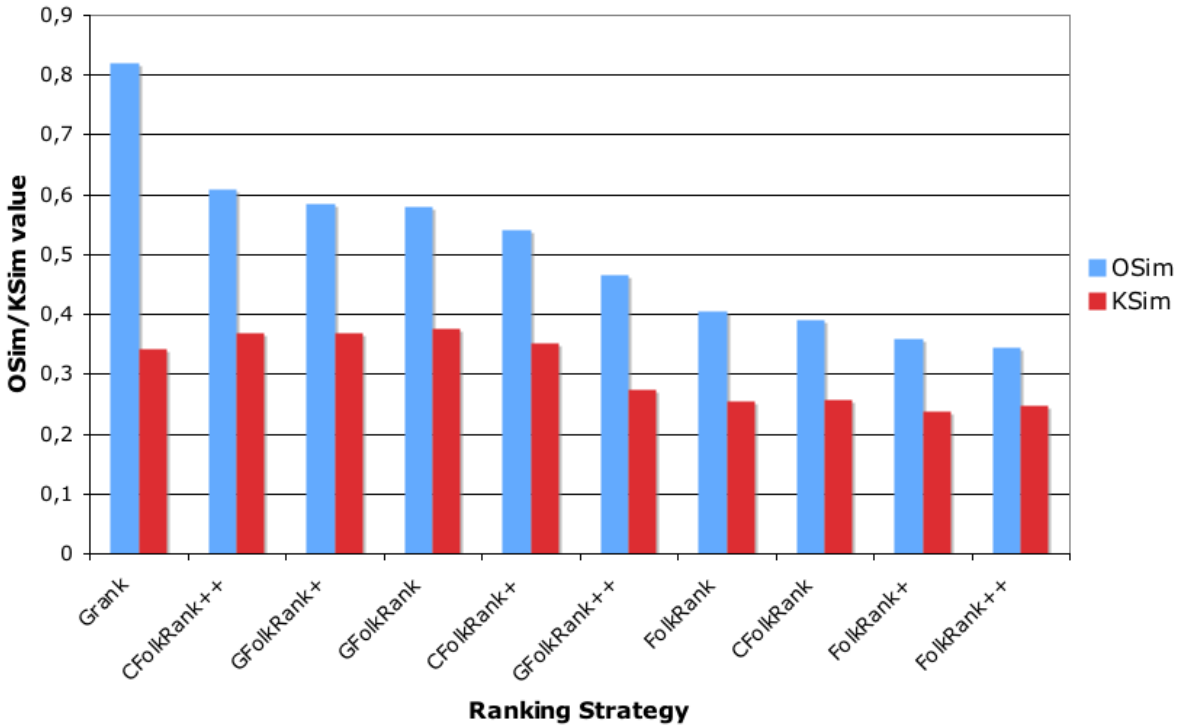


Figure 4.3: Overview of OSim and KSim for different ranking strategies, presented in Section 4.2, ordered by OSim. For tag propagation (cf. Section 4.2.2) we set the dampen factor $d = 0.2$.

exploit contextual information (see Section 2.2). The strategies are ordered according to their OSim performance, whereas both, OSim and KSim values are averaged out of 10 test series (for the 10 different keywords and corresponding hand-selected rankings). In terms of the OSim, GRank clearly outperforms the other ranking algorithms and can be identified as best strategy: it computes rankings, which contain 82% of the resources that also occur in the corresponding hand-selected ranking lists. However, with respect to KSim, GRank is worse than, for example, CFolkRank^{++}, which is the best FolkRank-based strategy in terms of OSim.

FolkRank, which does not exploit the group structure of group context folksonomies (cf. Definition 3.2), is outperformed by most of the group-sensitive ranking algorithms. Further, the extensions of FolkRank, FolkRank^{+} and FolkRank^{++}, which rudimentary exploit GroupMe! folksonomies, do not improve the overlapping similarity of 0.405 but rather degrade the quality of FolkRank. We assume that the approach of propagating tags without modeling the group dimension within the graph, which serves as input for the ranking algorithm, primarily increases the recall but has a negative effect on the precision.

Table 4.3 lists example rankings computed for the keyword query *"socialpagerank"* by different ranking strategies. Furthermore, it lists the corresponding hand-selected, av-

Rank	Hand-selected	FolkRank	GFolkRank	CFolkRank^{++}
1.	Optimizing web search using social annotations	Optimizing web search using social annotations	Optimizing web search using social annotations	Yahoo! research
2.	Exploring social annotations for the sem...	The Semantic Web: Will It All End In Tiers?	HITS	Optimizing web search using social annotations
3.	Personalized PageRank	New *Semantic* Web!	Webpage Ranking *(group)*	Ontologies are us
4.	SimRank	The Semantic Web: An Introduction	SimRank	Bibsonomy
5.	PageRank	The Semantic Web: Scientific American	PageRank	HITS
6.	FolkRank	LEGOLAND	Topic-sensitive PageRank	SimRank
7.	Ontologies are us	eschbach *(group)*	Personalized PageRank	PageRank
8.	Topic-sensitive PageRank	Andreas Eschbach Wikipedia	Yahoo! research	Topic-sensitive PageRank
9.	Webpage Ranking *(group)*	Andreas Eschbach Homepage	FRank	Personalized PageRank
10.	FRank	Andreas Eschbach: Der Nobelpreis	Ontologies are us	FolkRank

Table 4.3: Top 10 rankings computed by different ranking strategies (and selected by hand respectively) for the keyword query *"socialpagerank"*.

eraged ranking, which is based on judgments of five experts. Within the GroupMe! dataset the resource entitled *"Optimizing web search using social annotations"*, a paper which proposes the SocialPageRank algorithm, was the only resource tagged with the keyword "socialpagerank". According to the expert judgments, this resource should appear at the first rank when searching for "socialpagerank". FolkRank and GFolkRank compute rankings that conform to this decision and CFolkRank^{++} at least ranks the resource at rank 2. Starting from the second position the ranking of FolkRank becomes imprecise. As FolkRank does not exploit the group structure, it tries to discover other relevant resources via the users, who annotated the resource, and via the other tags that have been assigned to the resource. The group-based ranking strategies, on the other hand, are able to detect adequate resources via the group containing the resource. In the given example, this group is *"Webpage Ranking"* and the GFolkRank strategy is the only strategy that lists the group also within the top 10.

Figure 4.4 illustrates how the ranking strategies behave when varying the dampen factor for tag propagation. FolkRank and GFolkRank, which are listed as baselines, are not effected by the dampen factor because both strategies do not make use of tag propagation. When varying the dampen factor, the OSim value of GFolkRank^{+} and CFolkRank^{+}, which exploit the tags assigned to GroupMe! groups for propagating these tags to the resources of the groups, is comparatively constant. OSim and KSim of CFolkRank^{++}, which—in addition to group tags—propagates also the tags of a resource to its neighbor resources, continuously degrades when the dampen factor increases. The following example might explain this behavior of CFolkRank^{++}: given a resource r in a group g,

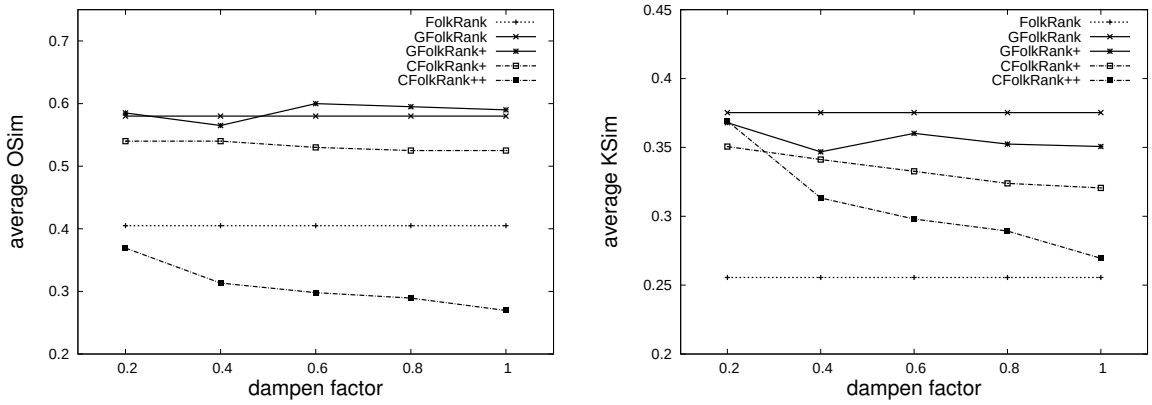


Figure 4.4: Average OSim and KSim (with respect to 10 different top 20 ranking comparisons) for varying dampen factors, which control the influence of propagated tags, and different ranking strategies.

which contains 20 other resources, and r is the only resource, which is tagged with t. Then, propagation of t to g and all resources of g with a dampen factor of 1.0 would conceal the prominent role of resource r in terms of tag t. Hence, ranking the resources of g in an adequate order becomes difficult (see KSim value), and the increased recall complicates the process of identifying resources to put into the top k of the search result ranking.

Result Summary

To give proof on our hypothesis that grouping improves the quality of search, it is necessary to compare the search strategies which explore the grouping context to those search strategies which do not. As benchmark, we have chosen the FolkRank algorithm. All algorithms, FolkRank as well as the group-aware ranking strategies, were tested under the same conditions, i.e. the same set of data, hardware, etc.

We tested our hypothesis with a one-tailed t-Test. The null hypothesis H_0 is that some group-sensitive ranking algorithm is as good as a the normal FolkRank without group-awareness, while H_1 states that some group-sensitive ranking algorithm is better than normal FolkRank. We tested it with a significance level of $\alpha = 0.05$. Tests were performed for the two measures OSim and KSim (see above).

OSim With respect to OSim, GRank is significantly better than all FolkRank-based algorithms. Furthermore, GFolkRank and CFolkRank^{++} are significantly better than FolkRank, FolkRank^{+}, and FolkRank^{++}. GFolkRank is not remarkably influenced by the propagation of tags (GFolkRank^{+} or GFolkRank^{++}). From our test data, we hypothesize that strategy CFolkRank benefits from the propagation of tags while GFolkRank does not. Our actual data did however not give statistically significant results on this.

KSim With respect to KSim, the strategy GFolkRank is significantly better than normal FolkRank, whether or not the latter uses any tag propagation strategy. Also the strategy CFolkRank$^+$, where group tags are propagated, is significantly better than FolkRank.

OSim and KSim GRank is definitely the best strategy with respect to OSim. However, GFolkRank and GFolkRank$^+$ are the only strategies that are significantly better with respect to both measures, OSim and KSim, than the normal FolkRank (whether or not the latter uses any tag propagation strategy).

Hence, the results of our search and ranking evaluations show that the algorithms that exploit group context in folksonomies significantly improve the quality of search in folksonomies. GRank is the most successful search algorithm as it gains the highest OSim and therewith detects a set of highly relevant resources. GFolkRank and GFolkRank$^+$ also gain high results for KSim, which makes them particularly useful for applications that are interested in comparing the relevance of resources relative to one another.

Optimizing GRank

The lightweight GRank algorithm performs best for the search and ranking task (see Problem 1). Given a tag t as query the GRank algorithm ranks a resource r based on four features: (a) the number of users who assigned t to r, (b) the number of user who assigned t to a group where r is contained in, (c) the number of tag assignments where t was used for a resource that is grouped together with r, and (d) the number of tag assignments where t was used for a resource that is contained in r (if r is a group resource). The influence of these features can be adjusted via corresponding parameters d_a, d_b, d_c, and d_d. In our evaluations presented in the previous sections we set $d_a = 10$, $d_b = 4$, $d_c = 2$, and $d_d = 4$ which is founded by the results shown in Figure 4.5(a-d).

Figure 4.5(a) depicts how OSim and KSim vary if d_a is altered from 0 to 20 while d_b, d_c, and d_d are constantly set to 1. The best performance with respect to OSim is outputted for $d_a = 3$ while KSim is maximized for $d_a = 5$ indicating that the first feature (number of user who assigned the query as tag to a resource) should be weighted stronger than the other features. In contrast, the influence of the neighboring resources should be rather small as indicated by Figure 4.5(c) where OSim and KSim are maximized if d_c is closed to zero and therewith smaller than d_a, d_b, and d_d that are equal to 1. Increasing d_c with respect to d_a, d_b, and d_d results in a significant degradation of the OSim and KSim metrics. An examination of the GroupMe! data set explains that observation: GroupMe! groups are often created for a specific task such as travel planing (cf. Section 3.3). Hence, neighboring resources, i.e. resources that are contained in the same GroupMe! group, as well as their tags might be inhomogeneous. For example, a video with travel information might be grouped together with the website of a computer science conference a user plans to attend.

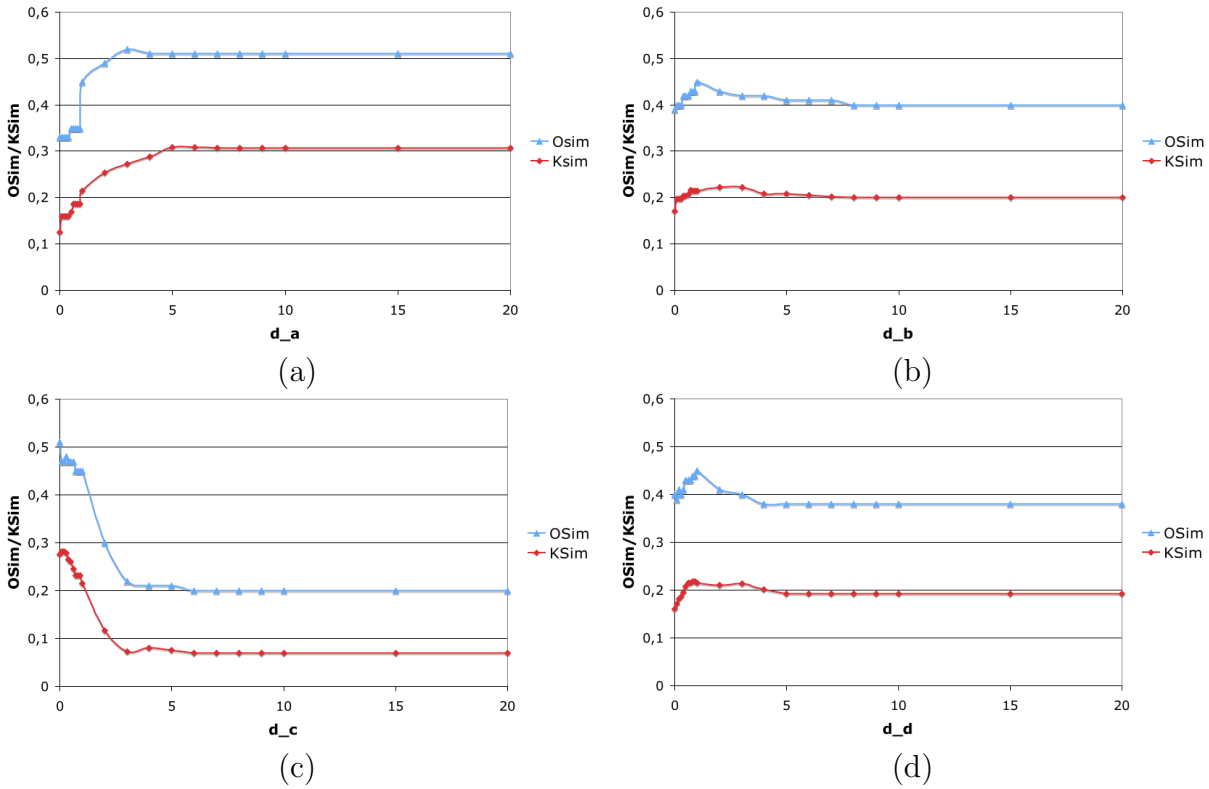


Figure 4.5: Varying parameters d_a, d_b, d_c, and d_d of the GRank algorithm. In Figure (a) the influence of direct tags (d_a) is altered from 0 to 20 while d_b, d_c, and d_d are constantly set to 1. In Figure (b), (c), and (d) the weights d_b, d_c, and d_d are varied correspondingly.

Fig. 4.5(b) reveals that the consideration of tags, which are assigned to a group the resource to be ranked is contained in, is reasonable. Setting d_b as high as d_a, d_c, and d_d produces the best OSim results and increasing d_b to 3 gains the best results regarding KSim. In comparison to tags of neighboring resources that possibly introduce noise, group tags (on average each group has approx. 3 tags) are the more appropriate feature to consider which means that setting $d_b > d_c$ leads to a better OSim/KSim performance. Similarly, $d_d > d_c$ leads to better results as clarified when comparing Fig. 4.5(d) and Fig. 4.5(c).

Using machine learning techniques one can learn and optimize the adjustment of the GRank parameters. For example, the SVM^{rank} algorithm [131], a *Support Vector Machine* approach [85] to the ranking problem, deduces an optimal model (with respect to the ground truth that is used as training set) with the following parameter assignments: $d_a = 1.18$, $d_b = 0.21$, $d_c = 0.14$, and $d_d = 0.26$. In comparison to the setting where all parameters are equally adjusted to 1, the deduced optimal model performs 24.4% better with respect to OSim and even 44.7% better with respect to KSim metrics.

4.3.3 Re-Ranking Experiment

Topic-sensitive ranking strategies can directly be applied to the task of searching for resources, e.g. FolkRank-based algorithms can model the search query within the preference vector (see Equation 2.4 in Section 2.2.2) in order to compute a ranked search result list. In the previous section we evidence that our group-sensitive ranking algorithms like GFolkRank and GRank (see Section 4.2.2) improve the search and ranking quality significantly (one-tailed t-test, significance level $\alpha = 0.05$) compared to FolkRank. Non-topic-sensitive ranking strategies – like SocialPageRank – compute global, static rankings and therewith need a baseline search algorithm, which delivers a base set of possibly relevant resources, which serve as input for the ranking algorithm. In our search and re-ranking experiment we thus evaluate the ranking strategies with respect to the task of *re-ranking a given search result set* (see Problem 2). The set of possibly relevant resources that has to be re-ranked is delivered by a *base set detection* strategy. In the next section we thus first identify an adequate strategy for detecting a good base set of search results.

Base Set Detection

The base set contains all search results, which are finally returned as a search result list, where the order is computed by the ranking algorithm. Hence, it is important to have a search method, which produces a base set containing a high number of relevant resources (high recall) without loosing precision. Table 4.4 compares different base set detection methods with each other.

Base Set Algorithm	*Recall*	*Precision*	*F-measure*
Basic	0.2767	0.9659	0.4301
BasicG	0.5165	0.7815	0.6220
BasicG$^+$	0.8853	0.6120	0.7237

Table 4.4: Comparison of different procedures to determine the basic set of relevant resources. Values are measured with respect to the test set described in Section 4.3.1.

Basic. Returns only those resources, which are directly annotated with the search keyword (cf. $\breve{R}_a$ in Definition 4.2).

BasicG. Returns in addition to *Basic* also resources, that are contained in groups annotated with the query keyword (cf. $\breve{R}_a \cup \breve{R}_b$ in Definition 4.2).

BasicG$^+$. This approach exploits group structures more extensively. It corresponds to our GRank algorithm without ranking the resources (cf. $\breve{R}_q$ in Definition 4.2).

Having a recall of nearly 90%, *BasicG*$^+$ clearly outperforms the other approaches. Though the precision is lower compared to *Basic*, which searches for directly annotated

resources, the *F-measure* – the harmonic mean of precision and recall – certifies the decisive superiority of $BasicG^+$. In our experiments we thus utilize the group-sensitive $BasicG^+$ in order to discover the set of relevant resources to be ranked. All ranking algorithms therewith benefit from the power of $BasicG^+$.

Re-Ranking Methodology

In our experiment we proceeded as follows. For each keyword query of our test set described above and each ranking strategy we perform three steps.

1. Identification of the base set of possibly relevant resources by applying $BasicG^+$ (cf. *Base Set Detection*).
2. Execution of ranking algorithm to rank resources contained in the base set according to their relevance to the query.
3. Comparison of computed ranking with the optimal ranking of the test set by measuring *OSim* and *KSim* (see Section 4.3.2).

Finally, we average the OSim/Ksim values for each ranking strategy.

Result Summary

Figures 4.6(a) and 4.6(b) present the results we obtained by running the experiments as described in the previous section. On average, the base set contains 58.9 resources and the average recall is 0.88 (cf. Table 4.4). The absolute OSim/KSim values are therewith influenced by the base set detection. For example, regarding the Top 20 results in Table 4.6(b), the best possible OSim value achievable by the ranking strategies is 0.92, whereas the worst possible value is 0.27, which is caused by the size and high precision of the base set. OSim and KSim both do not make any assertions about the relevance of the resources contained in the Top k. They measure the overlap of the top k rankings and the relative order of the ranked resources, respectively (see above).

As expected, the strategy, which ranks resources randomly performs worse. However, due to the high quality of the group-sensitive base set detection algorithm, the performance of the random strategy is still acceptable. SocialPageRank is outperformed by the topic-sensitive ranking algorithms. Personalized SocialPageRank, the topic-sensitive version, which we developed in Section 4.2.1, improves the OSim-performance of SocialPageRank by 16% and the KSim-performance by 35%, regarding the top 10 evaluations.

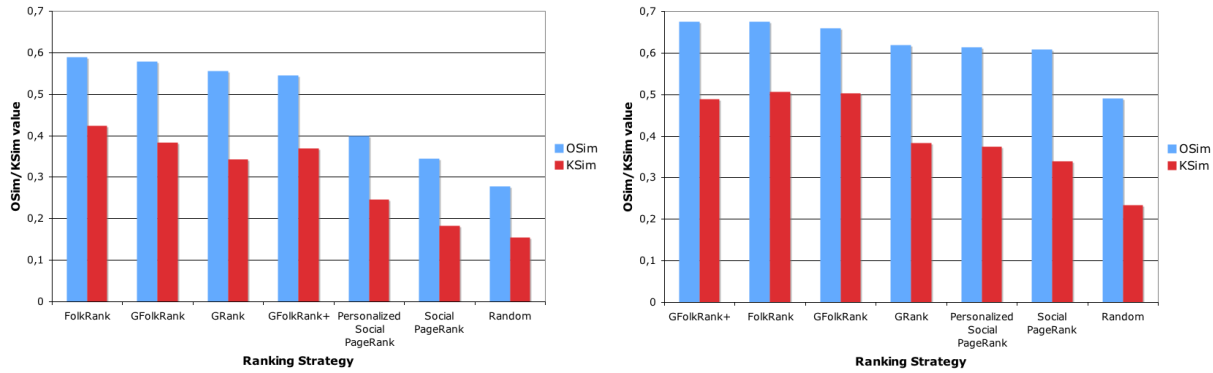
The FolkRank-based strategies perform best, especially when analyzing the measured KSim values. Regarding the performance of SocialPageRank within the scope of the top 10 analysis, FolkRank, GFolkRank, and GFolkRank$^+$ improve KSim by 132%, 110%, and 102% respectively. Here, the results evaluated by the OSim metrics also indicate an increase of the ranking quality, ranging from 58% to 71%.

It is important to clarify that all algorithms profit from the GRank algorithm, which is applied to detect the base set of possibly relevant resources. For example, if the topic-sensitive FolkRank algorithm is used without GRank then the quality would decrease significantly by 17%/13% with respect to OSim/KSim. Moreover, GRank can compete with the FolkRank-based algorithms in re-ranking the set of possibly relevant resources and produces – with respect to OSim and KSim – high quality rankings as well. For example in our top 10 evaluations, GRank performs 65%/89% (OSim/KSim) better than SocialPageRank, whereas FolkRank improves GRank slightly by 5%/25% (OSim/KSim). The promising results of GRank are pleasing particularly because GRank does not require computationally intensive and time-consuming matrix operations as required by the other algorithms.

The group-sensitive ranking strategies do not improve the ranking quality significantly. However, all ranking algorithms listed in Figures 4.6(a) and 4.6(b) benefit from the group-sensitive GRank algorithm, which determines the basic set and which supplies the best (regarding F-measure) set of resources that are relevant to the given query.

4.3.4 Synopsis

In this section we applied the ranking algorithms for search in group context folksonomies (cf. Definition 3.2), i.e. folksonomies where resources can be grouped together. We compared our context-based ranking algorithms (see Section 4.2.2) with algorithms such as FolkRank [124] or SocialPageRank [51] that do not exploit group structures in folksonomies and evaluated the performance of the algorithms with respect to (i) search and ranking (see Problem 1) as well as (ii) re-ranking of search results (see Problem 2).



(a) Top 10 OSim/KSim comparison.

(b) Top 20 OSim/KSim comparison.

Figure 4.6: Top 10 and top 20 OSim/KSim comparison between different ranking strategies. Basic Set is determined via BasicG$^+$ (cf. Section 4.3.3). In (a) the best possible OSim is 0.95 while the worst possible OSim: 0.04. In (b) the best possible OSim is 0.92 while the worst possible OSim is 0.27.

For both tasks the group-sensitive algorithms led to significant improvements (one-tailed t-test, significance level $\alpha = 0.05$) over the baseline ranking algorithms.

Search and Ranking Task. GRank was the best performing algorithm regarding the OSim measure. It performed significantly better than the FolkRank-based algorithms and gained a precision higher than 80% for the top 20 search results ($P@20$). With respect to KSim, which measures the quality of the pairwise order of resources within the ranking, the context-sensitive FolkRank-based approaches performed best. In particular, GFolkRank and GFolkRank$^+$ are significantly better than the FolkRank baseline with respect to both measures, OSim and KSim.

Re-Ranking Search Results Task. The re-ranking task requires a search algorithm for detecting a base set of resources that will be used as input for the ranking algorithms. For the base set detection we saw that the exploitation of group context information improves recall clearly from 0.28 (*return resources directly annotated with the query keyword*) to 0.89 (*exploit group structure like GRank does*). Given this high quality base set detection strategy, the FolkRank-based approaches achieve high precisions.

In summary, we see that group structures improve the quality of search significantly. A comparison of both search tasks further show that conventional ranking algorithms benefit from group-sensitive base set detection algorithms. For example, the search performance of the traditional FolkRank algorithm can be improved from 0.45 to 0.68 with respect to OSim (= $P@20$) when it is applied for re-ranking search results delivered by a group-sensitive algorithm.

4.4 Evaluation of other context-sensitive Ranking Algorithms

The consideration of context information thus has a positive impact on search. To confirm the findings made in the previous section, we conduct further search experiments on a dataset of Flickr images that were annotated using the TagMe! system (cf. Section 3.4). We evaluate and compare the context-sensitive ranking algorithms (see Section 4.2.3) with respect to the search and ranking task defined above (see Problem 1).

4.4.1 Dataset Characteristics and Ground Truth

The subsequent search experiment was conducted on a Flickr dataset that was annotated using the TagMe! system. Hence, the resulting context folksonomy provided three additional types of contextual information: (1) spatial information, (2) categorization of tag assignments, and (3) URIs that describe the semantic meaning of tag assignments.

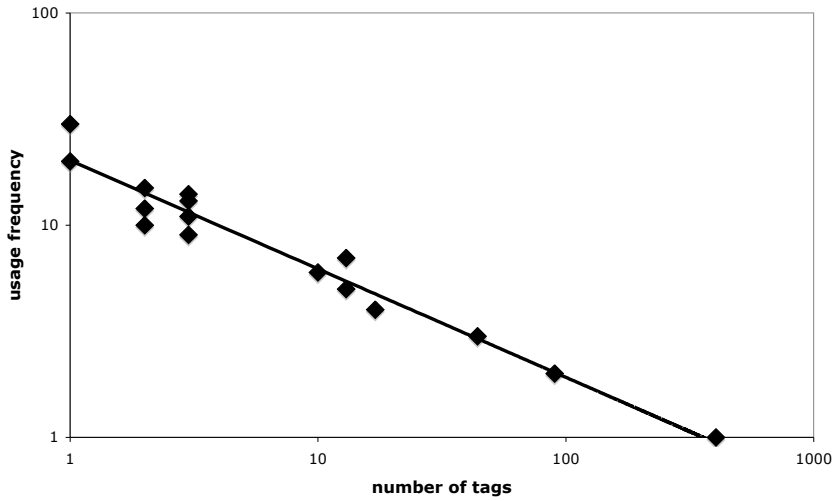


Figure 4.7: Tag usage in the TagMe! data set on a logarithmic scale. Only a few distinct tags have been used frequently while most of the tags are only used once.

Dataset Characteristics

We conducted our experiments on the TagMe! dataset that evolved during the first month after the launch of the system. In this period the users created 1264 tag assignments where 899 tag assignments were also enriched with a category and 657 tag assignments were attached to a specific area of a Flickr picture. As outlined in Section 3.4.2, the number of distinct tags was growing faster than the number of distinct categories. Finally, the TagMe! data set contained 610 distinct tags and 118 distinct categories. The distribution of the usage frequency of tags (see Figure 4.7) shows the same characteristics as detected in the dataset used in Section 4.3: while some tags are used very often, the majority of tags are used just once.

The DBpedia URI assignments that were automatically attached by TagMe! were validated by hand so that the data set on which we performed the experiments did not contain wrong URI assignments. The cleaned data set finally contained 360 distinct DBpedia URIs referenced by tags and 92 DBpedia URIs referenced by categories. For 17% of the tag assignments there did not exist a correct DBpedia URI mappings.

Ground Truth

The relevance assessment for gathering the ground truth was performed by ten users of the TagMe! system. We selected 24 representative tags (according to the usage frequency, cf. Figure 4.7) as keyword queries and asked the participants to rate the relevance of a picture to a given query on a five-point scale: *yes*, *rather yes*, *rather no*, *no*, and *don't know*. Therefore, for each of the queries we obtained all the relevant resources in the TagMe! dataset. On average, for each query there were nearly 30 resources in

the dataset that were rated as relevant (*yes*). However, four of the queries had below 10 relevant (*yes*) resources. For all the 24 tag-based queries a proper DBpedia URI was available in the dataset.

4.4.2 Search and Ranking Experiment

The *ranking task* defined at the beginning of this chapter (see Problem 1) requires the strategies to arrange those resources at the top of the ranking that are most relevant to the given query. We analyzed the ranking algorithms presented in Section 2.2.2 and Section 4.2.3 that are applicable to the context folksonomies as produced in TagMe!: FolkRank, Category-based FolkRank (CategoryFolkRank), Area-based FolkRank (Area-FolkRank), and URI-based FolkRank (DBpediaFolkRank). Each ranking strategy then had to compute a resource ranking for each of the 24 representative keyword queries. We measured the quality of the rankings using the precision and recall metrics which are defined as follows (cf. [49]).

Definition 4.5 (Recall and Precision) *Recall is the fraction of relevant entities* $E_{relevant}$ *which has been retrieved as answer set* E_{answer}*.*

$$Recall = \frac{|E_{answer} \cap E_{relevant}|}{|E_{relevant}|} \tag{4.4}$$

Precision is the fraction of retrieved entities E_{answer} *which is relevant.*

$$Precision = \frac{|E_{answer} \cap E_{relevant}|}{|E_{answer}|} \tag{4.5}$$

Most ranking algorithms introduced in Section 4.2 allow for arbitrary folksonomy entities (users, tags, and resources). With respect to the traditional folksonomy model (see Definition 2.1) recall and precision can thus be measured with respect to entities $e \in E$ where E is the union of users, tags and resources: $E = U \cup T \cup R$. However, for the *search task* specified in Problem 1 we only evaluate the performance of ranking resources: $E_{relevant} \subseteq R$. Further, we use $Precision@k$ ($P@k$) for characterizing the precision of top k ranking lists, i.e. the accuracy of the entities listed among the first k entities of a ranking. For example, $P@10$ refers to the precision within the entities ranked within the top 10 and $P@10 = 0.7$ means that 7 of the 10 top items are relevant. For our experiment we considered an item as *relevant* iff the average user judgement is at least "yes".

In addition to the FolkRank-based approaches we also consider a ranking algorithm (denoted as "F+C+A+D") that combines all four ranking strategies: given the list of weighted resources as computed by the different algorithms it utilizes the average ranking weight to rank the resources.

Following our experiments presented in Section 4.3, we tested the statistical significance of our results with a two-tailed t-Test with a significance level of $\alpha = 0.05$. The null

hypothesis H0 is that some strategy s_1 is as good as another strategy s_2, while H1 states that s_1 is better than s_2.

Result Summary

Figure 4.8 shows the precisions (P@10 and P@20) of the different ranking strategies. Those algorithms that make use of context information embedded in the folksonomy perform better than the traditional FolkRank algorithm that considers only the tag assignments without any additional context. Between DBpediaFolkRank and FolkRank there seems to be no remarkable performance difference. However, as noted above, the DBpediaFolkRank is operating on 215 fewer tag assignments than the other algorithms as for these tag assignments there exists no corresponding DBpedia URI. It is thus remarkable that DBpediaFolkRank still performs slightly better than FolkRank. The CategoryFolkRank presents good results especially with respect to the precision within the top 20 (P@20). Hence, the hypothesis raised in Section 3.4 seems to hold: category assignments can be used to relate resources. By exploiting the category context, the algorithm detects relevant resources that are not directly related via tag assignments to the given query. The AreaFolkRank algorithm, which exploits the size and position of spatial information attached to the tag assignments, is—with respect to P@10—the best algorithm among the core ranking strategies (P@10 = 52.9%). However, there is no significant difference between the FolkRank and the Area-, Category-, and DBpedia-based FolkRank.

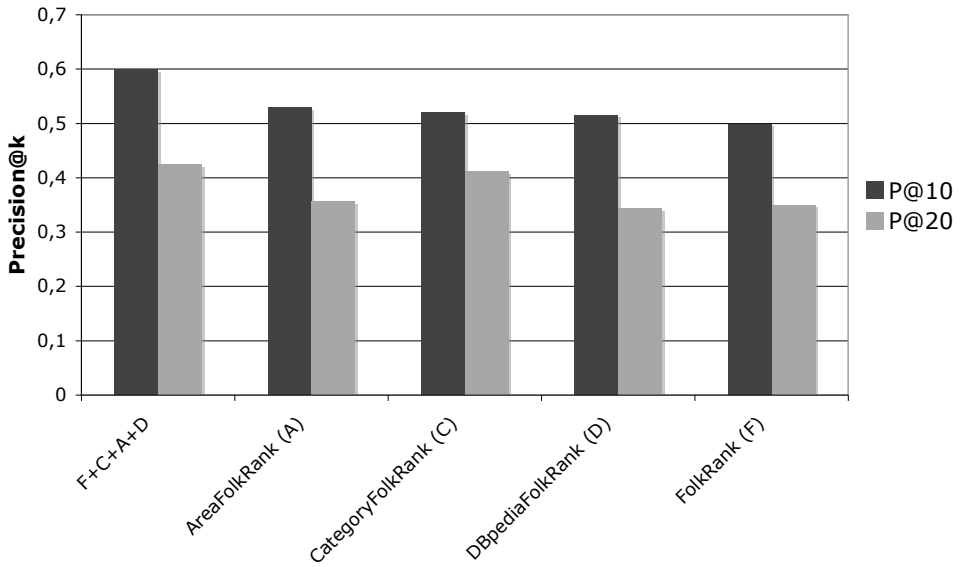


Figure 4.8: Precisions of FolkRank-based search algorithms.

The strategy "F+C+A+D", which combines all four core ranking strategies (i.e., FolkRank, CategoryFolkRank, AreaFolkRank, and DBpediaFolkRank), is the most successful strategy. It performs significantly better than the FolkRank algorithm regarding the P@10

and P@20 metrics. The combined strategy improves the precision of FolkRank by 20.0% and 21.4% with respect to the precision within the top 10 and top 20 respectively.

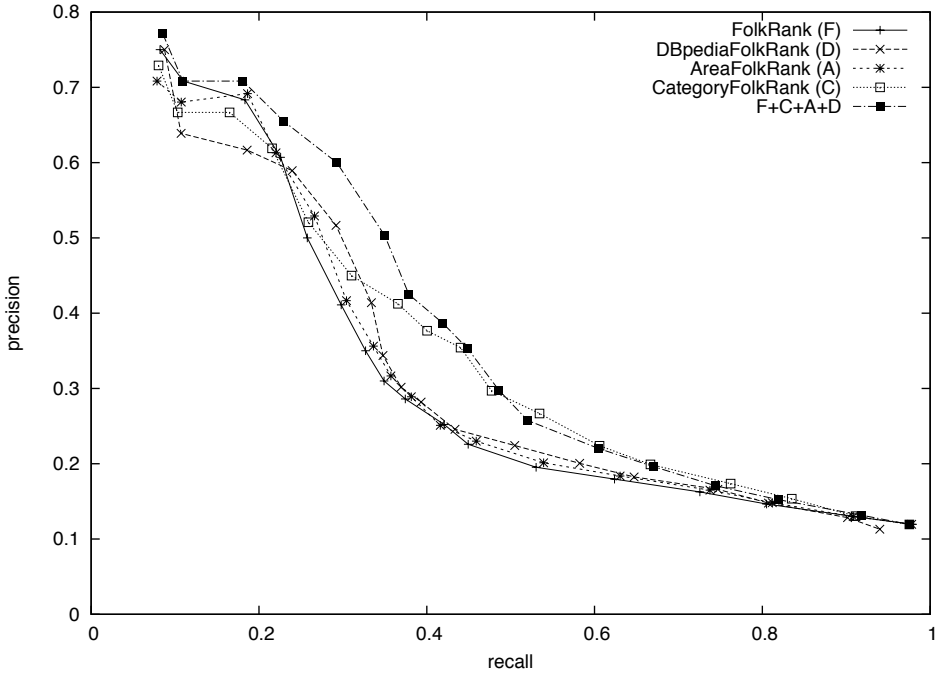


Figure 4.9: Precision recall diagram of FolkRank-based search algorithms.

Figure 4.9 depicts the precision-recall diagram of the different FolkRank-based ranking algorithms. It underlines that the context-based approach, which combines FolkRank with the strategies that exploit the category, area, and DBpedia context, is the best performing ranking strategy as it results in the best precision-recall ratio. In the low recall interval, i.e. within the very top of the resource rankings, FolkRank can compete with the other algorithms. For example, the probability that a relevant resource appears at the first rank is 75.0% for FolkRank and 79.2% for the combined strategy. However, with higher recall values, the precision of FolkRank drops significantly stronger than the one of the Category-based FolkRank or the combined strategy F+C+A+D: At a recall level of 0.5 the precision of F+C+A+D and CategoryFolkRank is 0.29 and therewith significantly higher (approx. 45%) as the precision of FolkRank.

4.4.3 Synopsis

In summary, the exploitation of context embedded in the folksonomy is beneficial for ranking resources. While the size and position of the area helps to improve the precision particularly at the top of the resource rankings, the DBpedia and category context successfully contribute to improve the recall. And by combining the different context types we are able to improve the ranking performance of FolkRank significantly.

Ranking Strategy	**applicable for**	**topic-sensitive**	**context-sensitive**
related work:			
FolkRank [124] (Sec. 2.2.2)	u, t, r	yes	no
SocialPageRank [51] (Sec. 2.2.2)	r	no	no
SocialSimRank [51] (Sec. 2.2.2)	t	yes	no
naive HITS [213] (Sec. 2.2.2)	u, t, r	yes	no
contributions of this thesis:			
SocialHITS [4] (Sec. 4.2.1)	u, t, r	yes	yes (groups)
Personalized SocialPageRank [23] (Sec. 4.2.1)	r	yes	no
GFolkRank [27] (Sec. 4.2.2)	u, t, r	yes	yes (groups)
CFolkRank [27] (Sec. 4.2.2)	u, t, r	yes	yes (groups)
GRank [1] (Sec. 4.2.2)	r	yes	yes (groups)
Category-based FolkRank [19] (Sec. 4.2.3)	u, t, r	yes	yes (categories)
Area-based FolkRank [19] (Sec. 4.2.3)	u, t, r	yes	yes (spatial context)
URI-based FolkRank [19] (Sec. 4.2.3)	u, t, r	yes	yes (URIs)

Table 4.5: Overview on characteristics of main ranking strategies applied and developed in this thesis.

4.5 Discussion

Table 4.5 summarizes features of the ranking strategies presented in this chapter and compares them with related research as presented in Section 2.2. The FolkRank-based algorithms as well as the HITS-based approaches are applicable for ranking of arbitrary folksonomy entities, i.e. users (u), tags (t), and resources (r). While ranking of resources is important for traditional search (see Section 4.3 and Section 4.4), ranking of tags and users becomes interesting for tag recommendations and expert search respectively (see Chapter 5). All algorithms—except for SocialPageRank—are *topic-sensitive*, which means that they do not only allow for the computation of static rankings but also allow for the adaptation of rankings to a certain topic. SocialPageRank computes static, global rankings independent of the given topic. For example, given a keyword query, SocialPageRank depends on an algorithms that detects (possibly) relevant resources, which are then re-ranked.

While the algorithms developed as part of this thesis support exploitation of additional semantics and contextual information, none of the algorithms, which are introduced by related research and listed in Table 4.5, exploit such information. GFolkRank, CFolkRank, and GRank denote search and ranking strategies, which exploit group structures of group context folksonomies (see Definition 3.2) and are therewith *context-sensitive*. FolkRank-based algorithms that make use of the tag propagation strategies (e.g. GFolkRank$^+$), presented in Section 4.2.2, are group-sensitive as well. The Category-, Area-, and URI-based FolkRank algorithms are also capable to exploit con-

Experiment	Items	Available Context	Results
Search and Ranking (Sec. 4.3.2)	bookmarks	group context	(1) Context-based algorithms (Sec. 4.2.2) better than FolkRank [124] (2) GRank (Sec. 4.2.2) is significantly the best algorithm (3) GFolkRank (Sec. 4.2.2) significatly better than FolkRank [124]
Search and Ranking (Sec. 4.4.2)	pictures	URIs, spatial information, categories	(1) Context-based algorithms (Sec. 4.2.3) better than FolkRank [124] (2) Exploitation of all context types leads to significant improvement over FolkRank [124]
Re-Ranking Search Results (Sec. 4.3.3)	bookmarks	group context	(1) GRank (Sec. 4.2.2) is significantly best base set detection algorithm (2) best re-ranking strategies: GFolkRank (Sec. 4.2.2), FolkRank [124] (3) Personalized SocialPageRank (Sec.4.2.1) improves performance of SocialPageRank [51] significantly

Table 4.6: Overview on search and ranking performance of the different algorithms.

textual information embedded in context folksonomies (see Definition 3.3). In particular, they analyze categories, spatial information and meaningful URIs in order to improve the ranking quality.

In the evaluations we measured and optimized the search and ranking performance of our algorithms. We compared the search performance of the context-based ranking strategies with approaches from related work (see Section 2.2) with reference to the search and ranking (see Problem 1) as well as re-ranking of search results (see Problem 2). Table 4.6 summarizes the results of our experiments. We conclude that algorithms which exploit contextual information available in folksonomies perform significantly better than algorithms which ignore such information. With respect to the two search tasks, our main findings regarding the research questions, (i) which ranking strategies perform best and (ii) what is the impact of contextual information on search, can be summarized as follows.

Search and Ranking. In the group context folksonomy setting, GRank outperformed the baseline ranking algorithm (FolkRank) clearly with respect to precision within the top 20 (82% in contrast to 40.5%). With GFolkRank we introduced a graph-based ranking approach that improves the search performance over the FolkRank baseline significantly (see Table 4.6). Overall, the consideration of contextual information has significant benefits for the search performance. We confirmed these findings for a setting where Flickr images were annotated with TagMe! and revealed that the exploitation of diverse context information improves the FolkRank baseline significantly by 20.0% and 21.4% regarding the P@10 and P@20 metrics respectively.

Re-Ranking Search Results. For the task of re-ranking search results we introduced a context-based search strategy that improves recall considerably from 0.28 (*return resources directly annotated with the query keyword*) to 0.89 (*exploit group structures with GRank*) and positively impacts the overall search performance of the re-ranking strategies. For example, the FolkRank algorithm can be improved from 0.45 to 0.68 with respect to the P@20 metric.

In addition to the good performance of the context-sensitive ranking algorithms introduced in Section 4.2.2 and Section 4.2.3, we observed that we also improved the performance of SocialPageRank by introducing Personalized SocialPageRank (see Section 4.2.1). However, still both algorithms cannot compete with algorithms that also make use of contextual information available in folksonomies. Having seen the positive impact of our context-sensitive ranking algorithms on search in folksonomy systems, we will in the following chapter investigate how these algorithms support personalization.

5 Context-based User Modeling and Personalization

In Chapter 4 we concluded that context-based ranking strategies (see Chapter 4) have a positive impact on search in social tagging systems. In this chapter we investigate whether usage of contextual information also improves the quality of personalization, i.e. personalized search and recommendation functionality. The main contributions of this chapter have been published in [3, 4, 20, 27].

5.1 Introduction: Towards Personalization in Social Web Systems

Today, information retrieval systems as well as the people who use these systems suffer from a tremendous information overload. For example, Google's search engine has indexed more than one trillion Web sites[1] and the Flickr folksonomy systems has to provide access to more than four billion pictures[2]. Given such huge amount of Web resources, the retrieval of relevant information becomes difficult. Mei and Church showed that leveraging user profile information available in search engine logs is beneficial to Web search as such profiles can be applied to infer the users' interests in Web pages [165]. The goal of personalization is to provide users with what they need without requiring them to ask for it explicitly [43], but rather infer user-specific needs from user interactions in order to adapt functionality [127]. In this chapter we introduce and evaluate different strategies for personalization in folksonomy systems based on the context models and context-based algorithms presented in Chapter 3 and Chapter 4 respectively. We investigate two personalization tasks, (1) personalized search and (2) generating recommendations.

Personalized search aims to adapt the search results for a given query to the actual needs of the user and can be defined as follows.

Problem 3 (Personalized Search) *Given a keyword query and a user in a specific context, the task of the ranking strategy is to compute a ranking of folksonomy entities so that entities that are most relevant to both the keyword query and the user context, appear at the top of the ranking.*

[1] `http://googleblog.blogspot.com/2008/07/we-knew-web-was-big.html`
[2] `http://blog.flickr.net/en/2009/10/12/4000000000/`

We propose strategies enabling the integration of context information independent from both the used ranking algorithm and the underlying folksonomy model. From a more technical perspective, there exist two main approaches to personalized search, modifying the search query issued by the user or processing the search result so that is conforms to the information needs of a user [179]. For example, Qiu and Cho adapt the search result lists by applying topic-sensitive PageRank scores [112] that correspond to the topic of the given query where topics are mapped to categories available in the open directory project so that the topic-specific PageRank score can be pre-computed [181]. Our strategies for contextualizing and personalizing search result rankings are based on *query expansion* [210]. Instead of applying co-occurrence based techniques [138] or using dictionaries, such as WordNet [94], we adopt approaches to personalized Web search as proposed by Lawrence [150], Chirita et al. [83], and Xiang et al. [215] and utilize context information to expand queries and adapt search result rankings to the actual user context. We thus not only exploit the users tagging activities to personalize search (cf. [172, 216]), but consider the current context of the users. Moreover, we evaluate our personalized search strategies with respect to ranking users, tags *and* resources as this has not been done by related research in the field of folksonomy systems.

In addition, we evaluate our personalization strategies with respect to the task that is more common in the area of folksonomy systems: computing recommendations. Lately, Sen et al. introduced the term *tagommenders* to describe recommender systems that exploit folksonomy structures to recommend items to a user [197]. In this chapter, we concentrate on tag recommendations as specified in Problem 4.

Problem 4 (Tag Recommendation) *Given a resource, the task of the ranking algorithm is to compute a ranking of tags so that tags, which are most relevant to the resource, appear at the very top of the ranking.*

Sigurbjörnsson and Zwol proposed the exploitation of tag co-occurrence statistics to recommend tags to a user [199]. Jäschke et al. [128] applied FolkRank to recommend tags and showed that it performs better than traditional collaborative filtering[193]. In this chapter, we examine different user and context modeling strategies in combination with context-based ranking algorithms and show that our approaches even improve the FolkRank baseline.

Given the two personalization task, personalized search and tag recommendation, we will answer the following research questions.

- How can user and context modeling strategies support personalization in Social Web systems?
- Which type of context and user modeling strategy is the most appropriate?
- Which algorithm performs best with respect to information retrieval metrics such as precision?

In the next section, we will first introduce user and context modeling strategies that exploit the structures of traditional (see Definition 2.1) and context folksonomies (see Definition 3.3). In Section 5.3 and 5.4 we will evaluate these strategies in combination with the context-based ranking algorithms (see Chapter 4) to the problem of personalized search and tag recommendation respectively. We conclude with a short summary in Section 5.5.

5.2 User Modeling and Contextualization

In this section we examine how user and context information can be deduced from user interactions. We not only focus on tagging activities performed by the user but also on other interactions with the folksonomy systems such as click-through data. To give some intuition for the notion of user and context information inferred from user interactions in folksonomy systems, we first describe a characteristic scenario in the GroupMe! tagging system, which we also used as test environment to conduct our experiments in the scope of personalized search and browsing (see Section 5.3.3). GroupMe! [10] enables users to manage their bookmarks and share them with other users and allows users to organize bookmarks in groups. Bookmarks as well as the groups can be annotated with tags (see Section 3.3).

5.2.1 Scenario

Let us consider that Bob is planning to travel to the Hypertext conference 2009. Therefore, he creates a GroupMe! group entitled "Trip to Hypertext '09, Turin", in which he adds bookmarks referring to the conference website or to some video showing sights of Turin. He also annotates his bookmarks with tags like "hypertext", "2009", or "conference" to facilitate future retrieval (cf. Figure 3.5(a)). Bob would appreciate some tag suggestions that expedite the tagging process. Alice is browsing through the GroupMe! system and stumbles upon Bob's group, because she is interested in submitting a paper to that conference. However, via the bookmarked conference website, which is part of the group, she finds out that the deadline has already passed. She now clicks on the tag "conference" and when she does so, likely she is not interested in *any* conference but in conferences that are related either to the same topics or to the year 2009 or that are related to combinations of all such features. Furthermore, she would be delighted to find expert users with whom she could discuss about appropriate conferences and corresponding topics.

In the scenario, the consideration of context can help to improve the usability of the folksonomy system: when computing tag suggestions, Bob's user profile as well as the tags that have already been assigned to other bookmarks in the "Trip to Hypertext '09, Turin" group can be considered. Further, when Alice clicks on the tag "conference"

she neither wants to retrieve bookmarks related to conferences in the field of biology nor seeks for information about past conferences, but she would like to obtain content relevant to computer science conferences in 2009. To adapt the search result to Alice's needs it would be appropriate to include the tags that occur within the Web page Alice visited when she clicked on "conference". Hence, adaptation would even be possible if Alice is not known to the tagging system or if she rarely interacts with the system so that the system has no detailed profile of Alice yet. Exploitation of the user context thus promises to improve the computation of personalized recommendations as well as personalized search and browsing experiences.

5.2.2 User and Context Modeling Strategies

Our approach models contextual user interactions in folksonomy systems as tag-based profiles, which are lists of weighted tags. Based on the traditional folksonomy model specified in Definition 2.1, a tag-based profile can be computed for a specific context as outlined in Definition 5.1.

Definition 5.1 (Tag-based Profile) *A tag-based profile is a set of weighted tags where the weight of a tag t is computed by a certain strategy w with respect to a given context c.*

$$P(c) = \{(t, w(c,t)) | t \in T, c \in U \cup T \cup R \cup \{\epsilon\}\} \tag{5.1}$$

$w(c,t)$ computes the weight that is associated with tag t in a given context c. ϵ describes the empty context.

Hence, a tag-based profile can be considered as a *tag cloud* where each weight characterizes the importance of the tag for the profile: the more important the tag, the higher the weight. In this chapter, we restrict the context to folksonomy entities (users, tags and resources) as well as the empty context ϵ for which the weighting functions needs to compute a context-independent score. Further, we compute the weight associated with a tag t by counting tag assignments in which t appears together with the given context. For example, the tag-based profile of a user can be defined as follows (see Definition 5.2).

Definition 5.2 (Tag-based User Profile) *The tag-based user profile $P_U(u)$ of a user u is deduced from the set of tag assignments performed by u.*

$$P_U(u) = \{(t, w(u,t)) | (u,t,r) \in Y, w(u,t) = |\{r \in R : (u,t,r) \in Y\}|\} \tag{5.2}$$

$w(u,t)$ is the number of tag assignments where user u assigned tag t to some resource.

Hence, the weight assigned to a tag simply corresponds to the usage frequency of the tag. Correspondingly, we define the tag-based profile of a resource $P_R(r)$.

Definition 5.3 (Tag-based Resource Profile) *The tag-based resource profile $P_R(r)$ of a resource r is deduced from the set of tag assignments where r occurs.*

$$P_R(r) = \{(t, w(r,t)) | (u,t,r) \in Y, w(r,t) = |\{u \in U : (u,t,r) \in Y\}|\} \tag{5.3}$$

$w(r,t)$ is the number of tag assignments where some user assigned tag t to resource r.

In the scenario above, Alice and Bob are acting in the GroupMe! system, which implies a *group context folksonomy* (see Definition 3.2). Given such a group context folksonomy, tag-based profiles for users, tags, and resources are computed correspondingly to traditional folksonomies, whereas a *tag-based group profile* $P_G(g)$ ($g \in G$) is computed by unifying $P_R(g)$ (groups are resources as well and can therefore be tagged) and the group-specific profiles of resources contained in g (see Definition 5.4).

Definition 5.4 (Tag-based Group Profile) *The tag-based profile $P_G(g)$ of a group g of resources is deduced from $P_R(g)$ and from the tag assignment that where performed in context of group g.*

$$\begin{aligned} P_G(g) \;=\; \{(t, w(g,t)) | g \in \breve{R}, t \in T, \\ w(g,t) = |\{u \in U, r \in R : (u,t,r,g) \in \breve{Y}\}| \\ +|\{u \in U : (u,t,g,\epsilon) \in \breve{Y}\}|\} \end{aligned} \tag{5.4}$$

$w(g,t)$ is the sum of the number of tag assignments where some user assigned tag t to group g and the number of tag assignments where t was assigned to some resource in context of g.

In this chapter, we normalize the weights so that the sum of the weights assigned to the tags in the tag-based profile is equal to 1. We use $\bar{P}(c)$ to explicitly refer to the tag-based profile where the sum of all weights is equal to 1. Furthermore, we use $P_U@k(u)$, $P_R@k(r)$ and $P_G@k(g)$ respectively to refer to the tag-based profiles that contains only the top k tags, which have the highest weight. More advanced tag-based profiles can be generated by allowing for other sorts of contexts or by applying a more complex weighting scheme. For example, in Chapter 6 we experiment with tag-based profiles that are generated by considering also temporal context as well as multiple context information sources instead of just using one single folksonomy entity as context. However, in this chapter we focus on the tag-based profile models specified above and show that even these simple models lead to significant improvements in the areas of personalized search and recommender systems. For personalized search and content exploration (see Section 5.3) and the recommendation experiments (see Section 5.4) we compare different lightweight approaches for constructing context from user interactions.

User. The user context is the top k tag-based profile of the user ($P_U@k(u)$), who is acting and whose actions should be contextualized, i.e. the tags he/she used most frequently.

Resource. If a user has navigated to a certain resource r then the tag-based profile of the resource $P_R@k(r)$ can be used as context to adapt to his/her next activities.

Group. Correspondingly, if the user currently browses a group g of resources, e.g. a GroupMe! group or a set of images in Flickr, then $P_G@k(g)$ can model his/her context.

The user context corresponds to the *naive user modeling strategy* described in [167] and only works if the user is already known to the system by means of previously performed tagging activities. In our evaluation we utilize the user context strategy as the benchmark and investigate whether the resource and group context strategies, which do not require any previous knowledge about the user, can compete with the user context strategy.

The context models are deliberately simple. More complex models can, for example, be constructed by combining the context models above or by logging resource and group context for a user over a specific period in time. In our evaluation in Section 5.3.3 we set $k = 20$ and thus considered the top 20 tags of the tag-based profiles while for the the recommendation experiments (see Section 5.4) k is set to 5.

5.3 Personalized Search

In this section we investigate how the user and context modeling strategies (see Section 5.2) in combination with the ranking algorithms (see Chapter 4) perform for personalized search (see Definition 3). We focus on search settings where users are browsing through a social tagging system in order to explore content, i.e. users issue a query, navigate to selected search results and explore further content by issuing another query.

5.3.1 Strategies for Personalized Search

Our approach to personalized search and personalized content exploration is to adapt the search result rankings computed by the ranking strategies to the given user and the user's context. In Section 5.2 we introduced different strategies for inferring contextual user profiles from user interactions. Topic-sensitive ranking algorithms can apply these strategies by adapting the topic which specified via the query. For example, a keyword query issued by a user can be enriched with further keywords that described the user's personal needs in the given context. The personalized search algorithms should thus generate a search result ranking that respects the query as well as the context given by means of a tag-based profile (see previous section). In the scenario above the query was given as single tag, e.g. Alice clicked on a tag to retrieve both a ranked list of resources and a ranked list of users, who are experts in Alice's current area of interest. A query might however also consist of multiple tags and can therewith be interpreted

as tag-based profile as well where the tags are usually weighted equally.

In Definition 5.5 we introduce a generic algorithm for computing personalized rankings that requires a topic-sensitive ranking strategy s like FolkRank or SocialHITS (see Chapter 4) as input. Given a (possibly multi keyword) query $P(q)$ and contextual information about the user $P(c)$ the ranking strategy s is applied to generate two rankings R_q and R_c by using $P(q)$ and $P(c)$ respectively as query. Using a common mixture approach both rankings are then combined to produce the ranking R_r that is finally returned as output. A contextualized ranking is thus the weighted average of the query and context ranking.

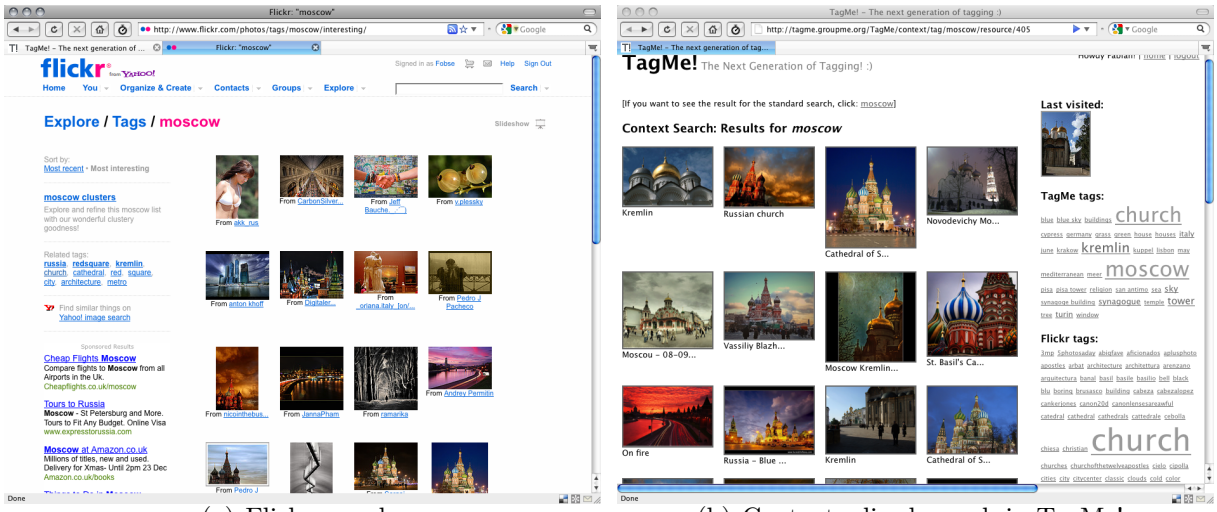
Definition 5.5 (Contextualization of Ranking) *The* generic algorithm for computing contextualized rankings *combines the ranking computed with respect to the query with the one computed for the tag-based context profile.*

1. ***Input:*** *query $P(q)$, context $P(c)$, folksonomy $\mathbb{F}$, ranking algorithm s, context influence $d \in [0..1]$.*
2. *Compute a ranking R_q based on the query tag profile, $R_q \leftarrow s.rank(P(q), \mathbb{F})$, and a ranking R_c based on the context tag profile, $R_c \leftarrow s.rank(P(c), \mathbb{F})$. R_q and R_c are sets of weighted entities (e_i, w_q) and (e_i, w_c) respectively.*
3. *Compute the result ranking R_r by averaging R_q and R_c. R_r contains weighted entities $(e_i, w_{i,r})$, where $w_{i,r} = (1-d) \cdot w_{i,q} + d \cdot w_{i,c}$ and d specifies the influence of the ranking scores computed via the tag-based context profile.*
4. ***Output:*** *R_r, the set of weighted entities $(e_i, w_{i,r})$, where $w_{i,r}$ denotes the weight (ranking score) assigned to the ith entity (user, tag, or resource).*

The generic algorithm for contextualizing rankings enabled us to test various approaches to personalized search by combining the ranking strategies introduced in Chapter 4 with the different user and context modeling strategies defined in Section 5.2.2.

In TagMe! we apply our approach to contextualize search for pictures so that end-users can immediately experience contextualized browsing. Figure 5.1 shows a comparison between Flickr search and the contextualized search in TagMe!. In both settings the user is searching with the tag "moscow" as the given query and in both settings the Flickr *interestingness approach*[3], which considers clicks, comments as well as tags in order to determine the interestingness of a picture with respect to a query, is utilized as ranking algorithm. However, TagMe! applies the algorithm for computing contextualized rankings, i.e. it queries Flickr first for pictures related to "moscow", then utilizes the context ($P(c)$)—and particularly the tag-based profile of the last visited resource ($P_R(r)$)—to retrieve related Flickr pictures and finally combines both rankings. In the example depicted in Figure 5.1(b), the user accessed an image showing a church in Moscow Kremlin

[3]http://www.flickr.com/explore/interesting/



(a) Flickr search (b) Contextualized search in TagMe!

Figure 5.1: Searching for pictures related to "moscow" – Flickr ranking according to *interestingness* vs. contextualized ranking in TagMe!

before clicking on the tag "moscow". TagMe! successfully adapts the resulting search ranking of Flickr pictures to that context as it ranks those pictures higher that are related to both, the search tag ("moscow") and the context tags (e.g., "church", "kremlin").

While the contextualized search and exploration interface of TagMe! is rather a showcase of our context-based approach to personalized search, we evaluate the approach extensively in Section 5.3.3.

5.3.2 Dataset Characteristics and Ground Truth

The experiments were performed on dataset of the GroupMe! tagging system (cf. Section 3.3) with respect to a test set of search settings. Given these settings we conducted an extensive user study to obtain a sufficient high number of judgements to gain statistically significant results.

Dataset Characteristics

In the GroupMe! dataset we had 450 users, who mainly come from the research community in Computer Science. Together they bookmarked 2189 Web resources, created 550 groups to organize these bookmarks and made 3190 tag assignments using 1699 different tags. Figure 5.2 illustrates that the tag usage reminds of a power law distribution as there are a lot of tags (72.04%), which were only used once, and only a few tags, which were applied frequently. For example, the tag "semantic web" was assigned 60 times and was therewith the most frequently used tag. Hence, regarding the tag usage distribution we observed similar characteristics as they occur also in larger data sets (cf. [88, 110]).

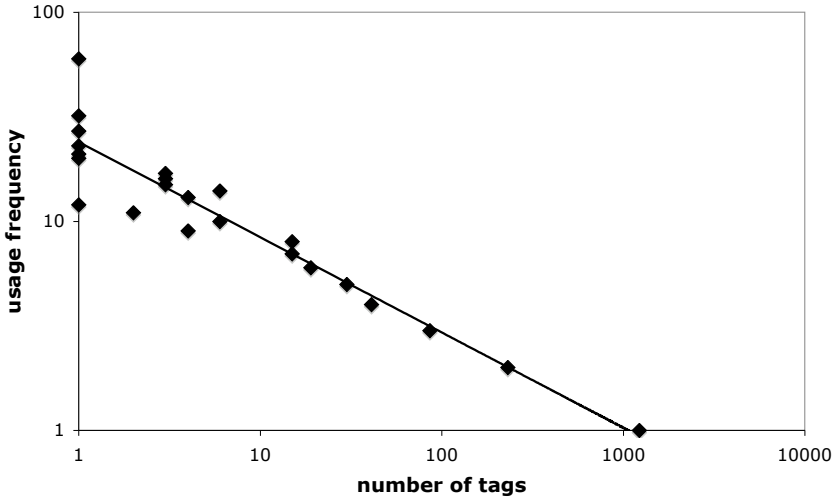


Figure 5.2: Tag usage in the GroupMe! data set on a logarithmic scale. Only a few distinct tags have been used frequently while most of the tags are only used once.

Test Set

For our experiments, we defined a test set of 19 search settings, where each setting was formed by a keyword query (tag) and a context consisting of (i) the *user* u, who performs a search activity, (ii) the *resource* r the user u accessed before initiating the search activity, and (iii) the *group* that contains r. We thus simulated the scenario described in Section 5.2.1, where the user Alice first accessed a group of resources, which were related to the "Hypertext '09 conference", then focused a certain resource (the conference website), before she finally clicked on the tag "conference" to search for related content. For the search settings, we selected tags as queries that cover the different spectra of the tag usage distribution. In particular, we chose 6 tags that were used 1-10 times (e.g. "soa" and "james bond"), 9 tags that were used 11-20 times (e.g. "conference" and "beer"), and 4 tags that were used more than 20 times (e.g. "hannover" and "semantic web"). The topics of the different search settings represented the diversity of topics available in the GroupMe! data set. For each of the 19 search settings we also selected a resource and a corresponding group as context, where the *tag-based resource context profile* (cf. $P_R(r)$, Section 5.2) contained 3.21 tags on average and the *tag-based group context profile* $P_G(g)$ contained 13.58 tags. Further, for each search setting we defined a user u as *actor*. Here, the condition was that the actor is also related to the topic of the setting, i.e. we only selected those users who already used the tags that occurred in the tag-based profile of the corresponding resource ($P_R(r)$) and group ($P_G(g)$) of the setting. Thereby, we tried to give the user modeling strategy ($P_U(u)$) the same opportunities as the resource and group context strategies to fulfill the task defined above.

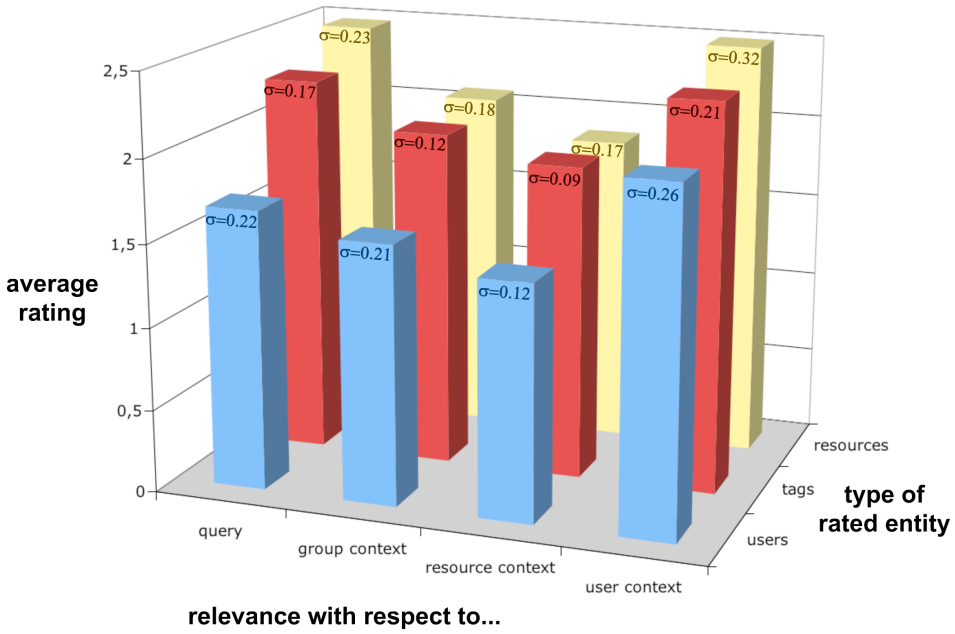


Figure 5.3: Characteristics of the judgment behavior in the user study with respect to the types of rated entities (user, tag, or resource) and the type of judgment basis (query, group context, resource context, or user context)

User Study and Ground Truth

Given the different search settings, we conducted a user study with users of the GroupMe! system (10 PhD students and student assistants) where the participants had to do relevance assessment (for the given setting, the low number of 10 participants was sufficient to obtain significant results). We presented the participants of the study a search setting together with a list of users, tags, and resources that were determined by accumulating the rankings of the different strategies for the given search setting. For each entity (user, tag, or resource) the participants judged the relevance of the entity with respect to the (i) query, (ii) group context, (iii) resource context, and (iv) user (actor) context. Therefore, they were enabled to easily gather information on which they could constitute their judgements, e.g. all involved entities were clickable and the participants were able to see an entity while judging it. In particular, the participants had to answer whether an entity is relevant or not on a five-point scale: *yes*, *rather yes*, *rather no*, *no*, and *don't know*. Thereby, we obtained a set of 8593 user-generated judgements, in particular 1550 *yes*, 1549 *rather yes*, 1097 *rather no*, 4242 *no*, and 155 *don't know* judgements. Figure 5.3 overviews the 8593 user judgements and the overall judging behavior of the participants with respect to the type of entity (user, tag, and resource) that was judged on the basis of its relevance to the query and the different parts of the context (group, resource, and user context). The average judgement is given as number, where 0 means *don't know*, 1 means *no*, 2 means *rather no*, etc. The standard deviation σ is averaged across the deviations of judgments, where the different participants evaluated the same

entity with respect to the same query/context.

Overall, the standard deviation indicates that the judgments of the participants were very homogeneous. Rating the relevance of entities with respect to the user context was probably the most difficult task for the participants, because they had to browse the profile of the corresponding user, i.e. the groups he/she created, the resources he/she bookmarked, and the tags he/she used in the past. Hence, the standard deviation for that judgement task is higher than for the others. Judging tags was the most intuitive task and also gained the most homogenous judgements. On average, the resources were rated better than tags, and users. This can be explained by the number of possibly relevant entities listed in the user study. For example, there were probably less than 5 of 22 users but more than 20 of 43 resources relevant to the query "james bond". However, even if there would be a slightly different judging behavior regarding the different types of entities (users, tags, and resources) then this would not influence our results as all algorithms were initialized with the same settings.

In general, the characteristics of the data set of judgements carried out during the user study enable us to gain statistically well-grounded results.

5.3.3 Personalized Search Experiment

In Section 5.2 we proposed different ways to infer contextual user profile information from user interactions by means of tag-based profiles that describe the actual setting of the user. In Section 5.3.1 we explained how rankings can be adapted to such context independent of the underlying ranking algorithm. Several applicable ranking algorithms were discussed in Section 2.2 and Section 4.2. In summary, we now have a *tool box* that helps tagging systems to adapt rankings to the actual desires of the users. In this section we will evaluate this tool box with respect to the personalized search challenge (see Definition 3).

According to the challenge of personalizing search result rankings, the different strategies have to rank users, tags, and resources with respect to a given search setting consisting of a query and context as described in Section 5.3.2. In our experiments, we combined the ranking algorithms presented in Chapter 4.2 that are applicable to group context folksonomies—FolkRank, GFolkRank, GRank, and SocialHITS—with the different context and user models presented in Section 5.2 and then passed them to the algorithm for contextualizing rankings (Definition 5.5 in Section 5.3.1). Thereby we obtained 12 strategies, e.g. *FolkRank(user)*, which denotes the strategy that applies the FolkRank algorithm together with the user context, or *GRank(resource)*, which is the strategy that contextualizes the ranking produced by GRank with the resource context. Each ranking strategy then had to compute a user, tag, and resource ranking for each of the 19 search settings, which consist of a query and the (user, group, and resource) context. Thus, each strategy had to compute 57 rankings. To measure the quality of the rankings we used the following metrics.

MRR The *MRR* (Mean Reciprocal Rank) indicates at which rank the first *relevant* entity occurs on average.

S@k The Success at rank k ($S@k$) stands for the mean probability that a *relevant* entity occurs within the top k of the ranking.

P@k Precision at rank k ($P@K$) represents the average proportion of *relevant* entities within the top k.

For our experiment we considered an entity as *relevant* iff the average user judgement is at least "rather yes" (rating score ≥ 3.0), e.g. given three "rather yes" (rating score $= 3$) judgments and two "rather no" judgments (rating score $= 2$) for the same entity with respect to some setting then this was treated as *not relevant*, because the average rating score is 2.6 and therewith smaller than 3.0 ("rather yes"). Judgements where the participant stated "don't know" were treated as "no".

We present the results according to the following structure. We first evaluate the performance of the newly introduced SocialHITS algorithm, independently from the used context strategy. Afterwards we overview our core results that allow us to answer the questions raised at the beginning of this section. In Subsection 5.3.3 we analyze the performance of the strategies when they have to rank (a) user and (b) resource entities. We will particularly investigate the ability of the algorithms to rank users, because this has not been studied extensively in previous work yet. Our result analysis finishes with a summary regarding the performance of the different context models, which are used to adapt the rankings to the actual context of a user.

We tested the statistical significance of all following results with a two-tailed t-Test and a significance level of $\alpha = 0.05$. The null hypothesis H0 is that some strategy s_1 is as good as another strategy s_2, while H1 states that s_1 is better than s_2.

SocialHITS vs. naive HITS

The SocialHITS algorithm, which we introduced in Definition 4.1, expects a graph construction strategy as input, which creates a directed graph from the given folksonomy. A naive approach to construct such a graph is presented in [213] (cf. Section 2.2.2). Figure 5.4 compares this straightforward application of HITS with SocialHITS, a more complex approach, which causes a graph with higher *compactness*. The results are based on 171 test runs, where the algorithms had to rank user, tags, or resources regarding the different search settings described above. Entities were considered as relevant iff they were, according to the user judgments, relevant to both, the query and the context. SocialHITS outperforms the naive HITS algorithm significantly with respect to all metrics. For example, the mean reciprocal rank (MRR), which indicates the average rank of the first relevant entity, is more than 50% better when using SocialHITS instead of the naive approach. The same holds for S@1. In particular, the probability that a relevant entity appears at the first rank is 47.4% when using SocialHITS in contrast to 28.7% when

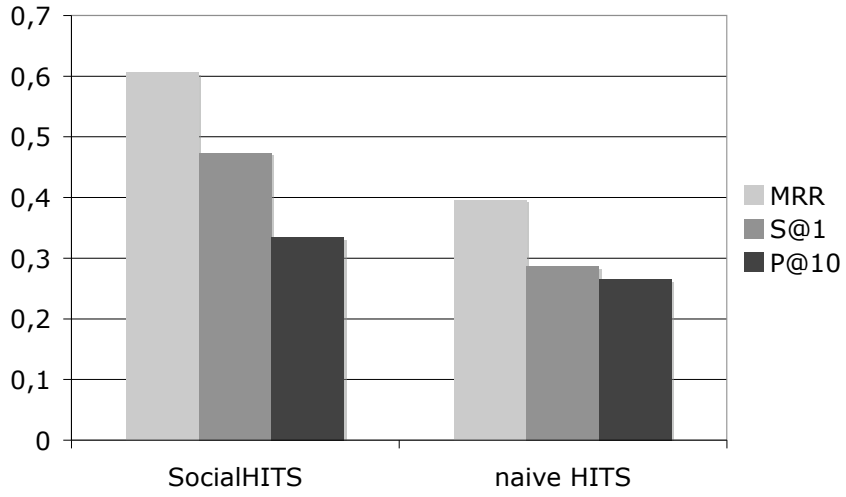


Figure 5.4: SocialHITS vs. naive HITS strategy (ordered by MRR(both)).

the naive approach is applied. Further, the precision within the top 10 is significantly higher for the SocialHITS algorithm.

The performance differences were obvious for every single ranking result. The naive HITS algorithm performed worst when it had to rank user entities. This can be explained from the underlying graph construction strategy, which implies an authority score of zero for user entities.

As SocialHITS outperforms the naive HITS approach we just consider SocialHITS for our comparisons with the other ranking algorithms presented in Section 4.2.

Result Overview

Figure 5.5 overviews the core results of our experiment. It shows the quality of the ranking algorithms in combination with the different user and context modeling strategies (Section 5.2) when using the contextualization strategy defined in Section 5.3.1. The metrics MRR(context), S@1(context), and P@10(context) determine the relevance of a particular entity with respect to the context, which is formed by the actor of a search setting as well as the resource and group context. For MRR(both), S@1(both), and P@10(both) relevance is given iff the entity is relevant to *both*, the query and the context of a search setting.

The GRank algorithm in combination with the resource context (*GRank(resource)*) is the most successful strategy for computing folksonomy entity rankings that should be adapted to a given search setting. *GRank(resource)* significantly performs better than all other strategies except for *GRank (group)* and *GFolkRank(resource)*. Overall, Figure 5.5 reveals two main results: (1) the GRank algorithm is the best performing algorithm and (2) independently from the used algorithm, the resource and group context models

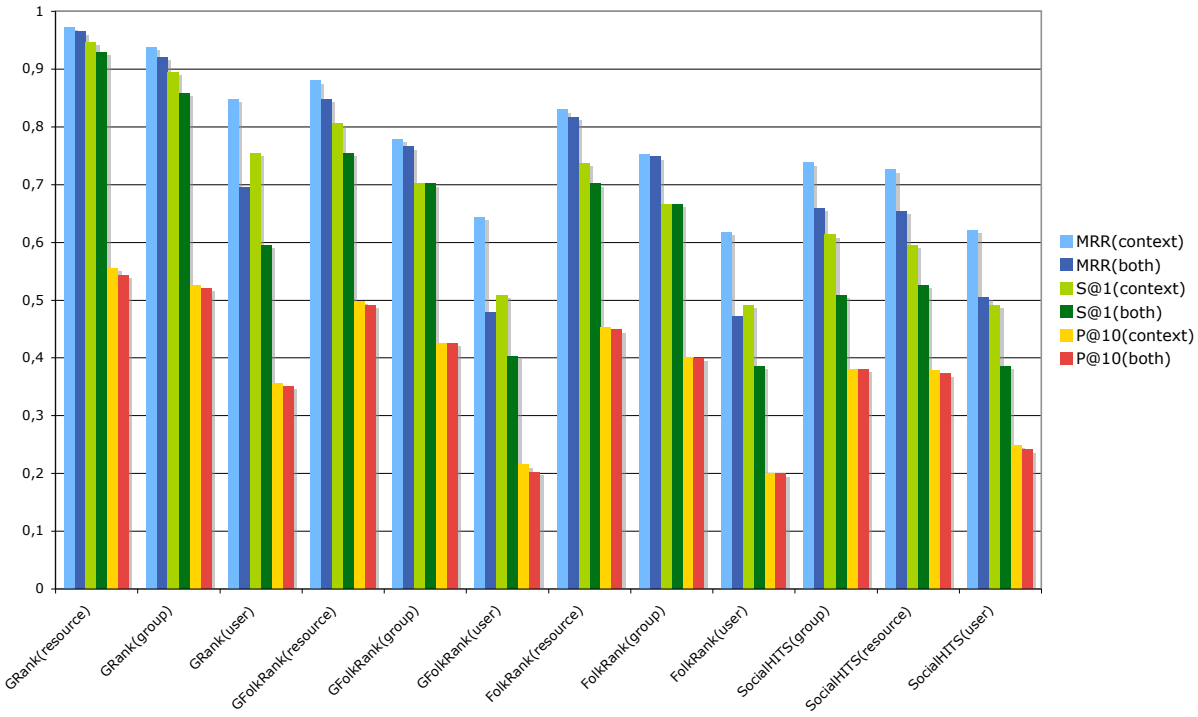


Figure 5.5: Performance of the different strategies with respect to the task of ranking folksonomy entities (ordered by MRR(both)).

produce better results than the user context strategy.

It is interesting to see that the precisions P@10(context) and P@10(both) do not differ significantly, which means that the items, which are included into the top 10 rankings because of their relevance to the context, are also relevant to the query. This gives supplemental motivation for the work, presented in this paper, as it indicates that the consideration of context does not reduce the precision of the result rankings within the top 10. Similarly, this motivation can be deduced from the S@1 metrics, as there is no significant difference between S@1(context) and S@1(both) for the strategies that make use of the resource or group context. However, the consideration of user context causes impreciseness regarding *query relevance* at the very top of the ranking. For example, the probability to retrieve an item that is relevant to the context of a search setting is 75.4% when *GRank(user)* is applied, whereas the probability that this item is relevant to the query as well is just 59.6%.

Between FolkRank and GFolkRank, the group-sensitive extension of FolkRank, there is not a significant difference in general, but GFolkRank performs better for all the different context models than FolkRank. The SocialHITS algorithm tends to be outperformed by the other algorithms. The performance of SocialHITS depends on the type of entity that should be ranked, while the performance of the other algorithms is rather constant, in this regard. SocialHITS significantly performs worse when it has to rank tags instead of users or resources. Hence, the role of tags in the model of SocialHITS (cf. Table 4.1)

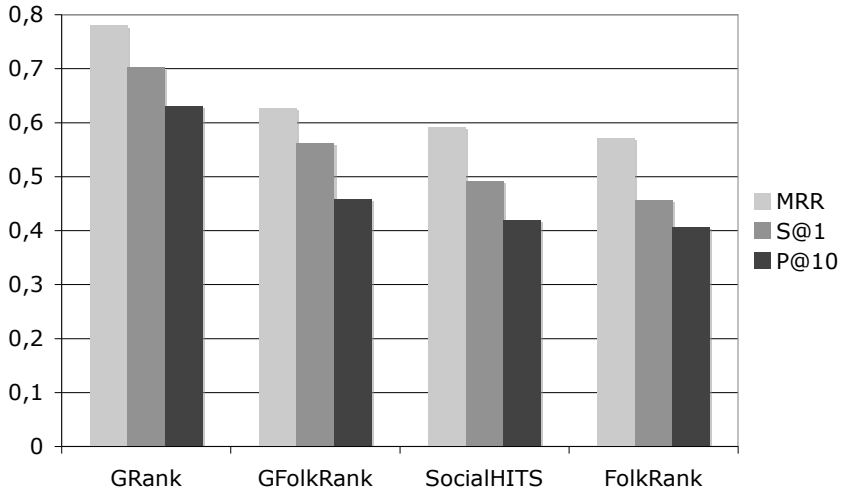


Figure 5.6: Performance of the different algorithms with respect to the task of ranking resources (ordered by MRR).

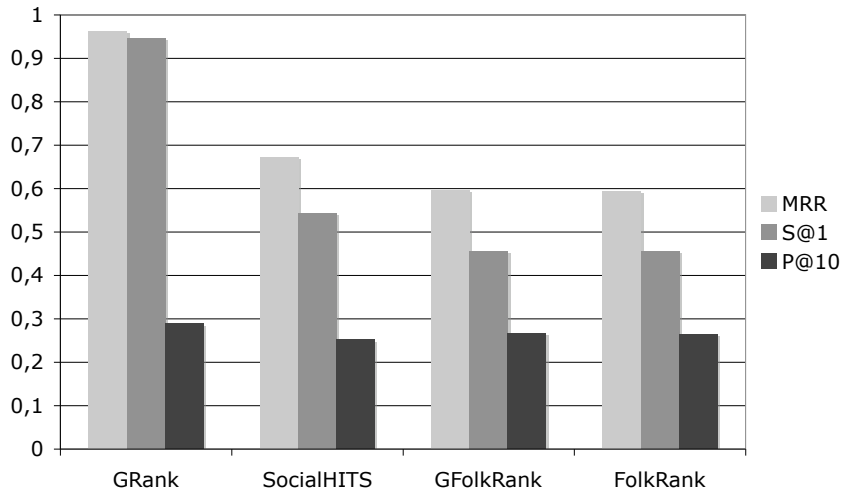


Figure 5.7: Performance of the different algorithms with respect to the task of ranking user (ordered by MRR).

can possibly be revised in future work to make SocialHITS also applicable to the ranking of tags.

Ranking Users and Resources

The task of ranking resources is possibly the most prominent ranking application, because it is, for example, applied to put search results into an appropriate order. Figure 5.6 overviews the performance of the different algorithms for that task averaged across the test runs targeting the different search settings while considering either the user, group,

or resource context. The metrics MRR, S@1, and P@10 are measured based on the relevance of a resource to both, the query and the context of the corresponding search setting.

GRank is significantly the best algorithm to rank resources followed by GFolkRank. Both algorithms exploit group structures in *group context folksonomies* (see Definition 3.2). Such folksonomies arise in tagging systems such as Flickr or GroupMe! which allow their users to group and tag the resources. In folksonomy systems that do not offer the notion of groups these algorithms would not work properly. In these systems SocialHITS would be the preferred choice because it shows better results than the FolkRank algorithm.

The results of the experiment focussing on ranking users is of particular interest because so far there exist – to the best of our knowledge – no studies which analyze the quality of folksonomy-based ranking algorithms in this regard. A set of exiting application can be realized with the aid of an user ranking functionality. For example, it can be applied to find experts on a certain topic or to recommend users to each other, who have – based on their tagging behavior – similar interests.

The qualification of the algorithms to rank user entities can be derived from the results shown in Figure 5.7. Overall, the outcomes are, regarding P@10, worse than the outcomes of the resource ranking experiment depicted in Figure 5.6. This can be explained by the absolute number of users possibly relevant to a search setting which is lower in comparison to the number of possibly relevant resources. GRank is again the best performing algorithm. For example, the probability that a user, who is relevant to the query and context, appears in the first position of the ranking is 94.7%. SocialHITS is the second best strategy having S@1 score that is 20% higher than the one of GFolkRank and FolkRank. Further, the mean reciprocal rank (MRR) of SocialHITS is more than 10% better than the one of GFolkRank and FolkRank, which do not differ significantly in their performance. Hence, SocialHITS is again the best choice for settings where no group context exists so that GRank is not applicable.

5.3.4 Synopsis

From the results presented in the previous subsections we can identify GRank, which we introduced in [27], as the best performing algorithm for ranking entities in *group context folksonomies* (see Definition 3.2). When it comes to the ranking of users or resources then SocialHITS, which significantly performs better than the naive HITS approach, is the best algorithm operating on the traditional folksonomy model (cf. Definition 2.1).

Figure 5.8 abstracts from the underlying ranking algorithms and summarizes the results listed in Figure 5.5 from the perspective of the context type that was considered by the algorithms to adapt the rankings to a particular search setting. According to the results shown in Figure 5.8, we can clearly put the strategies into an order: (1) the resource context gains significantly better results than the group and user context, (2) the group

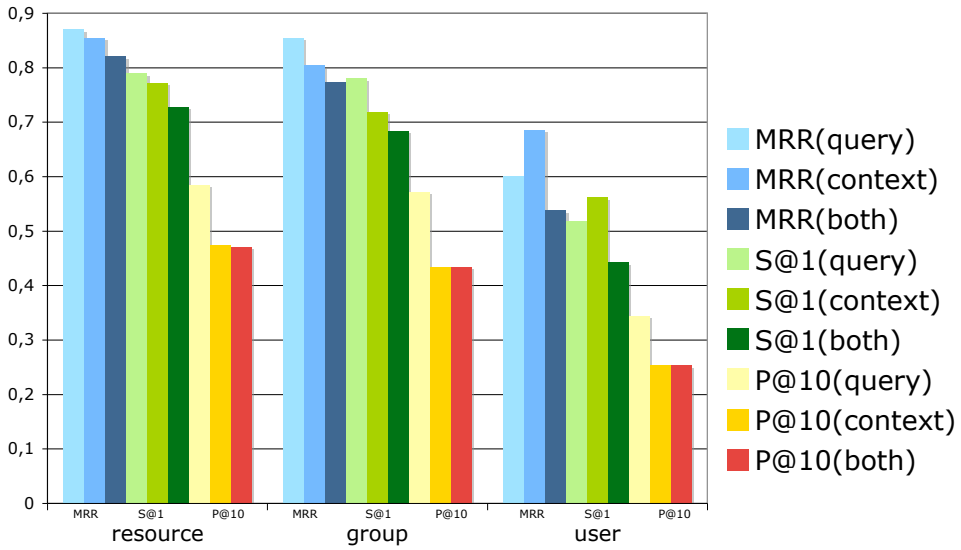


Figure 5.8: Performance depending on the used context type (ordered by MRR(both).

context strategy produces significantly better results than the user context strategy, while (3) the user context strategy performs worst. As described in Section 5.2, contextual user information is formed by the tag-based profile of a resource, group, or user respectively. The size of the different profile types differed: resource profiles contained on average 3.21 tags, tag-based group profiles 13.58, and tag-based user profiles were limited to 20 tags. However, the pure size of the context profiles does not only explain the outcomes of the experiment. For example, for some settings group context profiles containing more than 15 tags delivered better results than smaller tag-based profiles while for other settings it was the other way round. Hence, rather the homogeneity of a tag-based profile used as context seems to influence the quality of a contextualizing a ranking. The user context, i.e. the top tags of the user who performs a search activity, is thematically multi-faceted, which explains that the mean reciprocal rank measured with respect to the context (MRR(context)) is higher than the MRR measured regarding the relevance to the query (MRR(query)).

Overall, the excellent results of the resource and group context strategies are impressive, because they do not require any previous knowledge about the user, but just capture the current context of a user. The user modeling strategy on the contrary requires such knowledge. Our results have therewith a direct impact on the end users of a tagging system as they can benefit from the adaptation of result rankings to their current needs even if they are not known to the system.

5.4 Personalized Recommendations

In this section we evaluate the personalization framework, which consists of the user and context modeling strategies (see Section 5.2) and the context-based ranking algorithms (see Section 4.2), with respect to recommendation. In particular, we analyze the performance of tag recommendations (see Problem 4): given a resource the recommender strategy has to compute a ranking so that the tags a user might assign to this resource appear at the very top of the ranking.

5.4.1 Strategies for Computing Recommendations

The ranking algorithms we evaluate regarding the tag recommendation task are FolkRank [123] (see Section 2.2.2), SocialSimRank [51] (see Section 2.2.2), GFolkRank (see Section 4.2.2), and GFolkRank$^+$ (see Section 4.2.2). The other algorithms applicable to group context folksonomies—CFolkRank, GRank, and SocialPageRank—cannot be applied for ranking of tags without modifications. For the others, we utilize the following generic tag recommendation to exploit these algorithms to the recommendation problem.

Definition 5.6 (Generic Tag Recommender) *The generic algorithm for recommending tags to a resource r performs the following steps.*

1. *Select a preference vector $\vec{p}$ according to the context of r.*
2. *Compute the ranking of tags with respect to $\vec{p}$.*
3. *Optional: remove tags from the ranking that are already associated with r.*
4. *Recommend the top k tags of the ranking.*

In the first step of the algorithm the preference vector $\vec{p}$ weights tags that are relevant to the context of the given resource. In Section 5.2 we propose different user and context modeling strategies that can be utilized for constructing the preference vector. In this experiment we will evaluate the following three resource and group context modeling approaches for recommending tags a user might apply for a given resource r.

Tag-based Resource Profile (P_R) The tag-based profile $P_R(r)$ of a resource r (see Definition 5.3) is a weighted list of tags t, which are assigned to the resource. We determine the weight $w(r, t)$ according to the number of users, who assigned t to r.

Tag-based Group Profile (P_G) The tag-based profile $P_G(g)$ of a group g that contains the resource r the user would like to annotate. $P_G(g)$ is a weighted list of tags t, which have been assigned within the context of the group g (see Definition 5.4).

Group Tags (*GT*) When using the tags, which are directly assigned to a group g, as preference vector $\vec{p}$ for a resource r that is contained in g, we refer to this strategy as GT.

GT is the resource modeling strategy $P_R(g)$ (see Definition 5.3) where g is a group that contains the actual resource r for which the recommendations should be computed. For the FolkRank-based algorithms the preference vector $\vec{p}$ directly conforms to the preference vector in Definition 2.5 so that computation of the ranking of tags is simply done via executing the FolkRank-based algorithm. In a recommendation process for resource r, which applies SocialSimRank (cf. Definition 2.7), we compute similarity rankings $S_T(t_i, t_j)$ for each tag t_i, which is part of the preference vector $\vec{p}$, and compute the weighted mean of these rankings according to the weights, which are specified in the preference vector.

After computing the ranking of tags, the generic algorithm for recommending tags (see Definition 5.6) allows to remove those tags from the ranking that are already assigned to the resource. In our evaluations we perform this step and recommend only tags, which have not been assigned to the resource before. Finally, the top k entities of the computed (and possibly filtered) ranking are recommended.

By combining the ranking and preference selection strategies we gain twelve recommendation strategies ($FolkRank$ (P_R), $FolkRank(P_G)$, etc.), which we evaluate in the next sections.

5.4.2 Dataset Characteristics and Ground Truth

The tag recommendation experiments were conducted on the GroupMe! dataset described in Section 4.3.1. To measure the tag recommendation performance, we defined the *relevance* of a tag t to a given resource r is detected by two different modalities.

a. natural relevance t is the tag that was removed from resource r during the experiment, where we remove tags to evaluate if the ranking strategies are able to recommend exactly these removed tags to the resource again.

b. user-judged relevance t is a tag that was – in average – judged to be a *very good* or *good* description for the resource r.

The detection of *user-judged relevance* is executed on basis of a test set of user-judged tag recommendations. We randomly selected a test set of 52 resources, where each resource is equipped with at least two tags. The media type distribution within the test set corresponds to Figure 5.2 except for groups, which are not covered by the test set. We asked assessors to evaluate tag recommendations for each of the 52 resources. Therefore, we presented the assessors each resource together with a set of tags, which was gained agglomeratively by adding the respective top 10 tag recommendations of our different recommendation strategies (duplicates were eliminated). In correspondence to [199] and

for each resource r of the test set:
  for each tag t assigned to r:
    1. remove t from r;
    2. **compute tag recommendations:**
       run *generic tag recommender* (see Definition 5.6) for strategy s;
    3. **evaluate tag recommendation ranking:**
       apply MRR, S@k, and P@k metrics (see Section 5.3.3) to actual ranking;

average metrics values based on all computed rankings;

Figure 5.9: Applying *leave-one-out* method for evaluation of tag recommendations of strategy s.

our experiments presented in the previous section, for each tag the assessors judged the descriptiveness of the tag on a four-point scale: *very good*, *good*, *not good*, and *bad / don't know*. The set of user-generated judgements contains overall 3715 judgements, in particular 843 *very good*, 759 *good*, 617 *not good*, and 1496 *bad / don't know* judgements.

5.4.3 Tag Recommendation Experiment

In our evaluations of tag recommendation strategies, we used again MRR (mean reciprocal rank), $P@k$ (precision within the top k), and $S@k$ (probability that relevant item occurs within the top k) metrics (see Section 5.3.3) for measuring the performance of the strategies. Further, we run two kinds of experiments: *leave-one-out* [159] and *leave-many-out* [99] cross-validation.

Leave-one-out Evaluation. The leave-one-out method is convenient for small datasets [159] and is described in Figure 5.9. The removal of some tag t from a resource r has direct impact on the computation of the tag recommendations. It effects the characteristics of the preference vector, and it has an impact on the association matrices, which are utilized by the ranking algorithms (see Chapter 4). For example, it effects the construction of FolkRank's association matrix (cf. Definition 2.5) as the removed tag (assignment) is not considered for the folksonomy graph construction.

We run the leave-one-out method for each resource of the test set, which is described above. The average metrics scores are used to describe the quality of a certain recommendation strategy. For each ranking strategy, we repeat the experiment two times, using either the *natural relevance* or the *user-judged relevance* to decide if the recommended tag is appropriate or not appropriate.

Leave-many-out Evaluation. As outlined in Section 4.3.1, about 50% of the resources within the GroupMe! data set are not annotated with any tag. Therewith, the ability of recommending tags for resources, which are not tagged, becomes very important.

We evaluate the quality of our recommendation strategies with respect to untagged resources by applying *leave-many-out* validation [99], which is comparable to the leave-one-out method specified in Figure 5.9. However, instead of removing only one tag, all tags are removed from the resource, for which the tag recommendation is computed. The *natural relevance* of a tag t is given if t is one of the tags that was removed from resource r during the leave-many-out validation. Strategies, which exploit P_R as preferences, do not work for untagged resources and are thus not listed in Table 5.2.

Only the context-sensitive preference strategies (P_G and GT, see Section 5.4.1) are applicable for recommending tags to untagged resources, because untagged resources provide an empty tag-based resource profile (P_R), and therewith an empty preference vector $\vec{p}$.

Result Summary

By nature of the experiments, evaluations based on the *user-judged relevance* gain better results than the ones, which base on *natural relevance*, because in the latter approach the recommendation is only successful if exactly that tag, which was from the resource, is recommended back to the resource. Therewith, the precision values listed e.g. in Table 5.1.a, P@3 and P@5, are limited to $0.\bar{3}$ and 0.2, respectively.

Leave-one-out Evaluation. Table 5.1 lists the outcomes of the *leave-one-out* experiment. With respect to Table 5.1.a, $F(P_R)$ turns out to be the most successful recommendation strategy. However, the performance of the top 6 strategies does not differ significantly. The strategies, which employ group tags as preferences (GT), perform worse than strategies that make use of the tag-based resource profiles.

In Table 5.1.b we also list the results achieved in [199], where the authors proposed tag recommendation strategies, which are based on tag co-occurrence. Although the results in [199] were obtained in the same way as done in our experiment corresponding to Table 5.1.b, a one-to-one comparison is not feasible as the authors used another data set for their experiments. However, the graph-based algorithms seem to outperform the tag recommendation strategy proposed in [199]. Our recommendation strategies do especially well at the top ranks (S@1 and S@3) and regarding the average rank of the first relevant tag (MRR). For some resources within the GroupMe! dataset there hardly exist 5 relevant tags[4], which explains that the precision scores P@5 are not as high as the ones obtained from [199]. The strategies that apply SocialSimRank to rank the tag recommendations are clearly outperformed by the FolkRank-based approaches, e.g. considering S@1, $F(P_R)$ leads to an improvement of more than 39% compared to $S(P_R)$ (Table 5.1.b).

[4]Note that tags, which are already assigned to a resource, are not considered as tag recommendations.

Strategy	MRR	S@1	S@3	S@5	P@3	P@5
$\mathbf{F}(P_R)$	.7776	.7290	.7871	.8194	.2623	.1638
$\mathbf{G}^+(P_R)$	.7392	.7034	.7288	.7797	.2429	.1559
$\mathbf{G}(P_R)$	.7352	.6949	.7288	.7542	.2429	.1508
$\mathbf{G}(P_G)$	.7076	.6271	.7712	.8136	.2570	.1627
$\mathbf{G}^+(P_G)$	.6950	.6017	.7627	.8220	.2542	.1644
$\mathbf{F}(P_G)$	.6034	.4903	.7161	.7677	.2387	.1535
$\mathbf{S}(P_R)$	.5477	.4153	.6271	.7627	.2057	.1513
$\mathbf{G}(GT)$	.4328	.3814	.4322	.4831	.1440	.0966
$\mathbf{G}^+(GT)$	.4151	.3475	.4237	.4746	.1412	.0949
$\mathbf{F}(GT)$	.3371	.2323	.4000	.4645	.1333	.0929
$\mathbf{S}(GT)$	.1766	.0619	.1416	.2920	.0463	.0573
$\mathbf{S}(P_G)$	.1707	.0593	.1271	.2373	.0434	.0452

a. natural relevance

Strategy	MRR	S@1	S@3	S@5	P@3	P@5
$\mathbf{F}(P_R)$	.8777	.8387	.8903	.9290	.4172	.3187
$\mathbf{G}(GT)$	.8673	.8220	.9153	.9322	.5762	.4491
$\mathbf{G}^+(P_R)$	.8365	.8051	.8390	.8729	.4237	.3203
$\mathbf{G}(P_R)$	.8390	.7966	.8559	.8814	.3813	.2762
$\mathbf{G}^+(GT)$	.8477	.7881	.9068	.9322	.5734	.4440
$\mathbf{G}(P_G)$	.8572	.7712	.9576	.9746	.5367	.4169
$\mathbf{G}^+(P_G)$	.8490	.7542	.9492	.9746	.5734	.4355
$\mathbf{F}(GT)$	.8146	.7419	.8645	.9226	.5333	.4245
$\mathbf{F}(P_G)$	.8086	.6968	.9290	.9613	.5505	.4451
$\mathbf{S}(P_R)$	.7447	.6017	.8559	.9322	.4637	.3878
$\mathbf{S}(P_G)$	.5560	.3913	.6413	.7935	.3483	.3370
$\mathbf{S}(GT)$	.5556	.3621	.7069	.8448	.3594	.3739
Results of strategies as proposed in [199]:						
sum	.7628	.6550	-	.9200	-	.4930
vote	.6755	.4550	-	.8750	-	.4730
sum$^+$	.7718	.6600	-	.9450	-	.5080
vote$^+$	.7883	.6750	-	.9400	-	.5420

b. user-judged relevance

Table 5.1: Evaluation results for tag recommendation strategies measured via *leave-one-out validation* with respect to (a.) *natural relevance* and (b.) *user-judged relevance* (ordered by MRR). Results of benchmark strategies are obtained from [199]. F, G, G$^+$, and S denote FolkRank, GFolkRank, GFolkRank$^+$, and SocialSimRank respectively, which use P_R, P_G, or GT as preferences (see Section 5.4.1). MRR, S@k, and P@k are the metrics described above.

Leave-many-out Evaluation. The leave-many-out experiment evaluates the important task of recommending tags to resources, which do not have any tag. This task can only be solved by the group-sensitive ranking strategies. Table 5.2 lists the results of the corresponding experiment. Here, the results of the natural-relevance analysis are nearly as high as the ones of the user-judged relevance analysis, which is caused by the fact that all tags of a resource are removed and therewith the number of available, relevant tags in the data set increases. This especially impacts the precision of the tag recommendation.

In general, the ranking strategies which utilize the tag-based profile of a group (P_G) are the most successful strategies.

Strategy	MRR	S@1	S@3	S@5	P@3	P@5
$\mathbf{G}(P_G)$	.9426	.9268	.9268	1.0000	.5528	.3756
$\mathbf{G}^+(P_G)$	.9426	.9268	.9268	1.0000	.5528	.3756
$\mathbf{F}(P_G)$	.8959	.8537	.9024	1.0000	.5365	.3756
$\mathbf{G}^+(GT)$	.6591	.5610	.6829	.8537	.2926	.2243
$\mathbf{G}(GT)$	.6278	.5122	.6341	.8537	.2845	.2341
$\mathbf{F}(GT)$	.5733	.4390	.6341	.8049	.2845	.2439
$\mathbf{S}(P_G)$	.3523	.1951	.3902	.5122	.2051	.2051
$\mathbf{S}(GT)$	.2763	.1000	.3250	.5000	.1452	.1692

a. natural relevance

Strategy	MRR	S@1	S@3	S@5	P@3	P@5
$\mathbf{G}(P_G)$	.9682	.9512	.9756	1.0000	.6991	.5951
$\mathbf{G}^+(P_G)$	.9682	.9512	.9756	1.0000	.7398	.5951
$\mathbf{F}(P_G)$	.9520	.9268	.9756	1.0000	.6991	.5951
$\mathbf{G}(GT)$	.9174	.8780	.9512	1.0000	.6504	.5414
$\mathbf{G}^+(GT)$	.9052	.8537	.9512	1.0000	.6504	.5219
$\mathbf{F}(GT)$	.8776	.8293	.9024	.9756	.6260	.5414
$\mathbf{S}(GT)$	.6300	.4500	.7750	.8750	.4358	.4461
$\mathbf{S}(P_G)$	.5899	.4146	.6829	.8293	.4615	.4358

b. user-judged relevance

Table 5.2: Evaluation results for tag recommendation strategies measured via *leave-many-out validation* with respect to (a.) *natural relevance* and (b.) *user-judged relevance* (ordered by MRR).

5.4.4 Synopsis

The FolkRank-based algorithms outperform the strategies which apply SocialSimRank to rank the tags. SocialSimRank merely exploits the relations between resources and tags, whereas the FolkRank-based approaches additionally utilize user-tag and resource-user relations. Relations gained by the group context are essential when tag recommendations should be determined for untagged resources. For untagged resources the group-sensitive ranking strategies provide outstanding results, e.g. GFolkRank utilizing the tag-based profile of a group as preferences is, with MRR of 0.9682, the most successful strategy with respect to the mean rank of the first relevant tag (see Table 5.2.b).

5.5 Discussion

In this chapter, we designed strategies for modeling information about users and their current context when interacting with social tagging systems. We proposed to model such information via tag-based profiles (see Definition 5.1) and introduced different techniques for constructing such profiles. Further, we presented generic algorithms that allow for using such user and context models as input for traditional ranking algorithms outlined in Section 2.2 as well as for the context-based algorithms introduced in Chapter 4. User and context modeling strategies together with the context-based ranking algorithms thus constitute a personalization framework for social tagging systems.

We tested this personalization framework with respect to two personalization tasks, personalized search and tag recommendation, and discovered that these strategies which make use of contextual information lead to significant improvements over the baseline strategies. Our main findings can be summarized as follows.

Personalized Search. Given a user who is issuing a keyword query in a specific context, the lightweight context modeling strategies, which exploit the tag-based profile of the resource the user visited before querying, form the best source for personalizing the search results and improve the search performance significantly in comparison with heavy user modeling strategies as proposed by related work [97, 167] that apply the complete tagging history of the user. In particular, GRank in combination with the resource and group profile strategies (P_R and P_G) was clearly the best approach for adapting the search results to the personal information needs of a user. For example, regarding S@1—the probability that a relevant item appears at the first rank of the search result list—$GRank(P_R)$ achieved a performance of over 90% in contrast to the $FolkRank(P_U)$ baseline where S@1 was less than 40%.

In our experiments we also evaluated the performance of the approaches when searching for users which is new in the research on folksonomies and further promises high impact on social networking. Here, we identified SocialHITS (see Section 4.2.1) as one of the most promising ranking algorithms which indicates that the notion of hubs and authorities is applicable and meaningful when ranking users.

Tag Recommendation. The tag recommendation experiments confirmed our findings from personalized search. The lightweight context modeling strategies such as P_R succeed and exploiting contextual information becomes extremely important for recommending tags for resources that do not have any tag yet (untagged resources). For example, with S@1 of 0.927 $GFolkRank(P_G)$ outperforms the corresponding baseline strategies $FolkRank(P_G)$ (S@1 = 0.854) and $SocialSimRank(P_G)$ (S@1 = 0.195) clearly. Further, these baseline strategies benefit already from group context modeling and fail if they consider only traditional folksonomy structure.

We can thus summarize the answers to the research questions raised at the beginning of this chapter as follows.

- User and context modeling strategies (see Section 5.2) produce tag-based profiles which can be applied as preferences for the topic-sensitive ranking algorithms (see Chapter 4) to support personalization in folksonomy systems.
- Lightweight context modeling strategies performed better than heavy user modeling strategies.
- GRank performed best for personalized search, followed by SocialHITS (for ranking users) and GFolkRank, which was the best algorithm for recommending tags to untagged resources.

Comparison with related work. For the tag recommendation experiments we compared with the FolkRank-based recommender strategy as proposed in [128] and with SocialSimRank as introduced by Bao et al. [51] and showed that our tag recommendation strategies improve the performance of these approaches. Our tag recommendation methods successfully predict tags for untagged resources by exploiting contextual information while, by contrast, other approaches such as co-occurrence-based methods by Sigurbjörnsson and van Zwol [199] do not succeed.

For the personalized search experiments we compared different user and context modeling strategies and proved that our context-based user modeling strategies produce better results than user modeling approaches introduced by related work [97, 167]. Xu et al. study topic-sensitive search in folksonomy systems and try to infer topics the user is interested in to adapt search results to these topics [216]. However, Xu et al. do not evaluate the search performance with respect to specific search activities as we do, but apply the personomy of a user itself—and therewith the interests which are also used to personalize the search—to measure user satisfaction. Further, the approach proposed by Xu et al. is not resistant against untagged resources or users, who have not performed any tag assignment yet. Our approaches, by contrast, can handle these situations by exploiting contextual information so that users who do not tag can benefit from personalization as well.

6 Cross-System User Modeling in the Social Web

In Chapter 4 and Chapter 5 we revealed the benefits of our context and user modeling strategies for information retrieval as well as personal information retrieval in folksonomy systems. In this chapter we will investigate the benefits of modeling users across Social Web system boundaries, i.e. in context of users' Social Web activities. The main contributions of this chapter have been published in [13, 17, 18, 152].

6.1 Introduction: User Modeling across Social Web System Boundaries

In order to adapt functionality to the individual users, systems require information about their users [127]. The Social Web provides opportunities to gather such information: users leave a plethora of traces on the Web, varying from profile data to tags. In this chapter we analyze the nature of these distributed user data traces and investigate the advantages of interweaving publicly available profile data originating from different sources: social networking services (Facebook, LinkedIn), social media services (Flickr, Delicious, StumbleUpon, Twitter) and others (Google).

Connecting data from different sources and services is in line with today's Web 2.0 trend of creating *mashups* of various applications [220]. Support for the development of interoperable services is provided by initiatives such as the dataportability project[1], standardization of APIs (e.g. OpenSocial [173]) and authentication and authorization protocols (e.g. OpenID [183], OAuth [111]), as well as by (Semantic) Web standards such as RDF [140], RSS [211] and specific Microformats such as hCard [79] or RelTag [78]. Further, it becomes easier to connect distributed user profiles—including social connections—due to the increasing take-up of standards like FOAF [67], SIOC [64], or GUMO [114]. Conversion approaches allow for flexible user modeling [45]. Solutions for user identification form the basis for personalization across application boundaries [76, 125]. Google's Social Graph API[2] enables application developers to obtain the social connections of an individual user across different services. Generic user modeling servers

[1] `http://www.dataportability.org`
[2] `http://socialgraph.apis.google.com`

such as CUMULATE [219] or PersonIs [46] as well as frameworks, which we developed for mashing up profile information [5, 13, 29], appear that facilitate handling of aggregated user data. Given these developments, it becomes more and more important to investigate the possibilities of cross-system user modeling in context of today's Social Web scenery.

Mehta et al. showed that cross-system personalization [164] makes recommender systems more robust against spam and cold start problems [163]. However, Mehta et al. could not test their approaches on Social Web data where individual user interactions are performed across different systems and domains, but experiments have been conducted on user data, which originated from one system and was split to simulate different systems [162, 163]. Szomszor et al. present an approach to combine profiles generated in two different tagging platforms to obtain richer interest profiles [204]; Stewart et al. demonstrate the benefits of combining blogging data and tag assignments from Last.fm to improve the quality of music recommendations [202]. In this chapter we analyze the benefits of modeling tag-based user profiles across system boundaries (see Section 5.2, which we enrich with WordNet [94] facets. Further, we expand our analysis by considering explicitly provided profiles coming from five different social networking and social media services. We introduce an approach for interweaving user profiles that originate from diverse Social Web systems and prove that our approach positively impacts tag and resource recommendations. We particularly focus on cold-start recommendations [196] and investigate the performance of different recommender strategies over time beyond the *cold start*. In the subsequent section we will thus answer the following research question.

- What are the characteristics of user profile data distributed on the Social Web?
- How to model users across system boundaries in the Social Web?
- What are the benefits of cross-system user modeling in the Social Web?
- How does cross-system user modeling impact the performance of social recommender systems?

In Section 6.2 we will introduce our approach to distributed user modeling as well as the corresponding implementation. We evaluate both implementation and the general strategy before we analyze the impact on tag and resource recommendations in Section6.3. We conclude this chapter with a summary and discussion in Section 6.4.

6.2 Cross-system User Modeling with Mypes

In this section we introduce our strategy for user modeling across folksonomy system boundaries. We implemented our approach in the so-called *Mypes* service which we outline in Section 6.2.1. In Section 6.2.3 and 6.2.4 we evaluate the benefits of the Mypes service.

Our general approach to distributed user modeling is to aggregate user and context information from the different sources available on the Social Web. In Section 5.2 we proposed to model users and contextual information by means of tag-based profiles. Hence, for distributed settings we suggest to aggregate tag-based profiles that represent the same entity in different contexts. For example, a user might have tag-based profiles at different services such as Last.fm, Flickr, or Delicious. The aggregated tag-based profile can thus be computed by accumulating the profiles provided by the different services. However, as the tag-based profiles originating from the different sources may have different importance for the application that requires an aggregated profile, it should be possible to (de-)emphasize weights of the processed tag-based profiles with respect to the context in which these profiles have been generated.

In Definition 6.1 we specify how we implement the aggregation of tag-based profiles. The weight associated with a tag t_j is the sum of all weights—(de-)emphasized with parameter α_i—associated with t_j in the different profiles $P_i(c_i)$. Via parameters α_i one can adjust the influence of profile P_i on the aggregated profile P_{new}. In our experiments in Section 6.3, we set $\alpha_1 = ... = \alpha_n = 1$ unless otherwise stated.

Definition 6.1 (Profile Aggregation) *Given a set of tag-based profiles $P_1(c_1)$,..., $P_n(c_n)$, which were constructed in context of c_1,..., and c_n respectively, the aggregated profile P_{new} is computed by accumulating the tag-weight pairs (t_j, w_j) of the given tag-based profiles. The parameter α_i allows for (de-)emphasizing the weights originating from profile P_i.*

Input: $Profiles = \{(P_1(c_1), \alpha_1), ..., (P_n(c_n), \alpha_n)\}$
TC_A = *empty tag cloud*
for $(P_i(c_i), \alpha_i) \in Profiles$*:*
    $P_i(c_i) = \bar{P}_i(c_i)$ *(normalize so that sum of weights is equal to 1)*
    for $(t_j, w_j) \in P_i(c_i)$*:*
        if $(t_j, w_{P_{new}}) \in P_{new}$*:*
            replace $(t_j, w_{P_{new}})$ *in* P_{new} *with* $(t_j, w_{P_{new}} + \alpha_i \cdot w_j)$
        else*:*
            add $(t_j, \alpha_i \cdot w_j)$ *to* P_{new}
        end
    end
end
Output: $\bar{P}_{new}$

Aggregated profiles can be computed for the different types of entities (cf. Section 5.2.2) such as users so that one obtains an aggregated tag-based user profile. With *Mypes* [18] we introduce a service that allows for the aggregation of tag-based profiles. Further Mypes features include linkage, alignment, and enrichment of distributed user data.

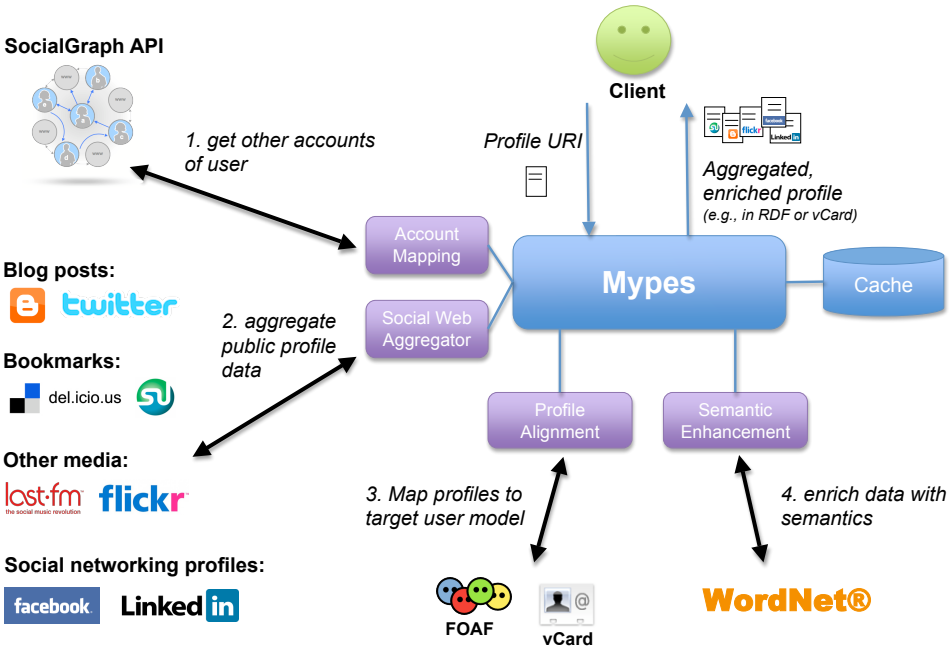


Figure 6.1: Aggregation and enrichment of profile data with Mypes.

6.2.1 Mypes Approach to User Modeling

Mypes supports the task of gathering information about users for user adaptive systems [127]. The Mypes service aims to provide a uniform interface to public profile data distributed on the Web. Such interface is valuable for casual users, who would like to overview their distributed profile data, as well as systems that require additional information about their users. To feature access to the distributed profile data, Mypes and the corresponding components depicted in Figure 6.1 respectively perform the following steps:

1. **Account Mapping** Given a user the first challenge is to identify the different online accounts of the user, e.g. her Facebook ID, her Twitter blog, etcetera. Mypes gathers other online accounts of the same user by exploiting the Google Social Graph API, which provides such account mappings for all users who linked their accounts via their Google profile, for example (cf. `foaf:holdsAccount` in Figure 6.3(b)):

```
"http://www.google.com/profiles/fabian.abel":
   "claimed_nodes": [
    "http://delicious.com/fabianabel",
    "http://fabianabel.stumbleupon.com",
    "http://www.last.fm/user/fabianabel/",
    ...
   ]
```

   For those users whose mappings cannot be obtained via the API, it is possible to provide appropriate mappings by hand. The account mapping module finally provides a list of online accounts that are associated to a particular user.

traditional profile attributes	Facebook	LinkedIn	Twitter	Blogspot	Flickr	Delicious	Stumble Upon	Last.fm	Google
nickname	x	x	x	x	x	x	x	x	x
first name	x	x							
last name	x	x							
full name	x	x	x		x				x
profile photo	x		x		x				x
about		x							x
email (hash)	x				x				
homepage	x	x	x						x
blog/feed			x	x	x	x	x	x	
location		x	x		x				x
locale settings	x								
interests		x							
education		x							
affiliations	x	x							
industry		x							
tag-based profile					x	x	x	x	
posts			x	x	x	x	x		
friend connections					x			x	

Table 6.1: Profile data for which Mypes provides crawling capabilities: (i) traditional profile attributes, (ii) tag-based profiles (= tagging activities performed by the user), (iii) blog, photo, and bookmark posts respectively, and (iv) friend connections.

Further, we implemented methods for identifying users across social tagging systems by analyzing their tag-based profiles as well as their usernames. Our experiments reveal that this can be done with high precision of approx. 80% [125]. However, in this article we apply account mappings as specified within the individual Google profiles, because for these mappings we observed an accuracy of 100% (see Section 6.2.2).

2. **Profile Aggregation** For the URIs associated with a user one then needs to aggregate the profiles referenced by the URIs. The aggregation module of Mypes gathers diverse profile data from the corresponding services. In particular, traditional profile information (e.g., name, homepage, location, etc.), tag-based profiles (tagging activities), posts (e.g., bookmark postings, blog posts, picture uploads), and friend connections (Flickr contacts and Last.fm friends) are harvested from nine different services as depicted in Table 6.1.

3. **Profile Alignment** To abstract from service-specific user models and create an appropriate aggregated user profile (see Definition 6.1) the profiles gathered from the different services have to be aligned. Mypes aligns the profiles with a uniform user model by means of hand-crafted rules. Further, Mypes provides functionality to export the aggregated profile data into different formats such as FOAF and vCard.

4. **Semantic Enrichment** Tag-based profiles are further enriched and clustered by means of WordNet categories which allows clients, for example, to access particular parts of a tag-based profile such as facets related to locations or people. For this purpose, Mypes performs a WordNet dictionary lookup to obtain the top-level categories

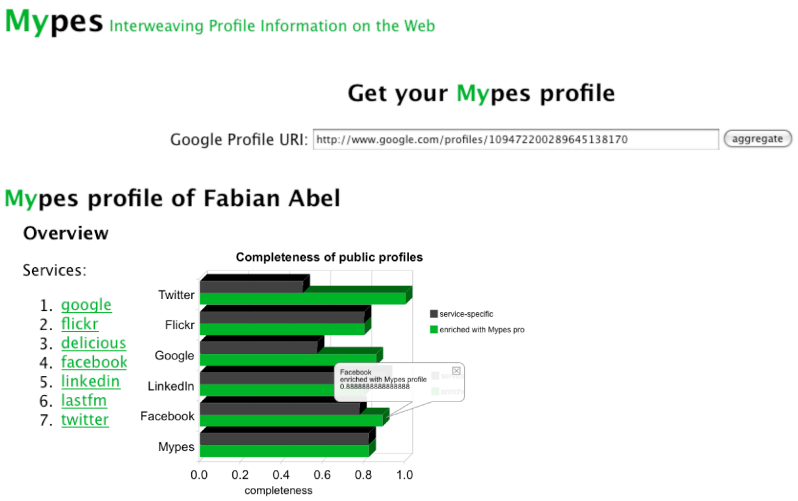


Figure 6.2: Overview on distributed profiles depicts to which degree the profiles at the different services are filled and to which degree they could be filled if profile information from the different services is merged.

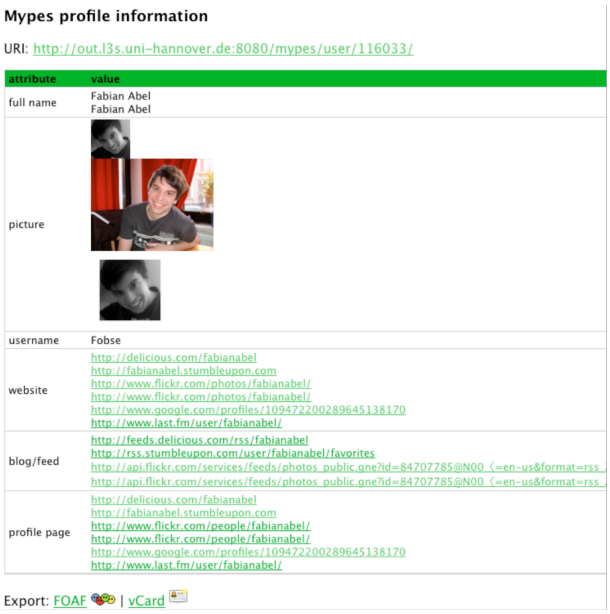
that can be deduced from the correspond to the lexicographer file organization[3]. Only tags that are contained in the WordNet dictionary will be mapped to WordNet categories. We discovered that approx. 65% of the tags can be mapped to appropriate WordNet categories [17].

Mypes Service Features

As we will discuss in more detail in Section 6.2.3, we observed that individual users complete their profiles for different services to a different degree. For example, the average Twitter profile is only filled for less than 50%, while LinkedIn profiles are completed to more than 80%. We believe that there exist users who intentionally do not complete their Twitter profiles and who are not aware that their Twitter account can be connected with other accounts that provide the missing profile information. To make users aware of their distributed profile traces, Mypes enables users to overview the completeness of their public profiles as depicted in Figure 6.2. Users can moreover inspect to which degree the Mypes profile, i.e. the aggregation of the different profiles, could complete their profiles at the different services. In the example shown in Figure 6.2, the completeness of the user's actual Twitter profile is 50%. However, all missing entries are available via the Mypes profile.

Figure 6.3 shows an example of an aggregated Mypes profile, namely the traditional profile attributes gathered from the diverse services (see Table 6.1). The traditional profile is also accessible in FOAF and vCard format via HTTP.GET. For example, the FOAF profile in RDF/XML syntax as listed in Figure 6.3(b) is returned when the

[3] `http://wordnet.princeton.edu/man/lexnames.5WN.html`



(a) Mypes profile

```
<foaf:Person
  rdf:about="http://mypes.groupme.org/mypes/user/116033/rdf">
  <foaf:name>Fabian Abel</foaf:name>
  <foaf:nick>Fobse</foaf:nick>
  <foaf:depiction
       rdf:resource="..buddyicons/84707785@N00.jpg"/>
  <foaf:depiction
       rdf:resource="..photos/11877/thumb/fob.png?1252108263"/>
  <foaf:weblog
       rdf:resource="http://feeds.delicious.com/rss/fabianabel"/>
  <foaf:weblog
       rdf:resource="http://rss.stumbleupon.com/user/fabianabel"/>

  <foaf:holdsAccount>
    <foaf:OnlineAccount rdf:about="http://fabianabel.stumbleupon.com">
      <foaf:name>StumbleUpon</foaf:name>
      <foaf:accountServiceHomepage
            rdf:resource="http://stumbleupon.com"/>
      <foaf:accountName>
        fabianabel
      </foaf:accountName>
    </foaf:OnlineAccount>
  </foaf:holdsAccount>
  <foaf:holdsAccount>
    <foaf:OnlineAccount rdf:about="http://www.last.fm/user/fabianabel/">
      <foaf:name>Last.fm</foaf:name>
      <foaf:accountServiceHomepage
            rdf:resource="http://www.last.fm"/>
      <foaf:accountName>
        fabianabel
      </foaf:accountName>
    </foaf:OnlineAccount>
  </foaf:holdsAccount>
  ...
</foaf:Person>
```

(b) Mypes profile exported to FOAF

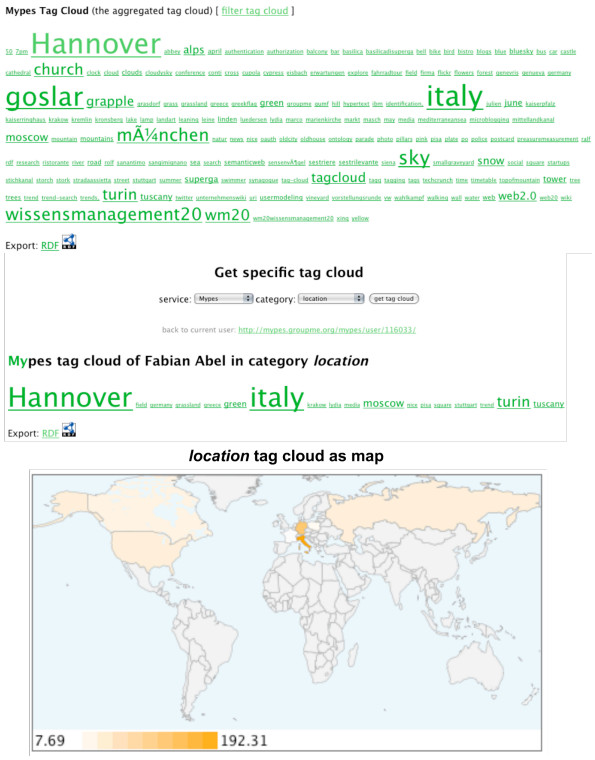
Figure 6.3: Aggregation of traditional profile information: (a) visualization of aggregated profile for end-users and (b) FOAF export of Mypes profile.

client accesses `http://mypes.groupme.org/mypes/user/116033/rdf`. Mypes exports all available values for a profile attribute, e.g., if a user specifies her name differently at the different services then all these different values are provided.

Mypes also connects the tagging activities that users perform in the various tagging systems. Figure 6.4(a) shows the aggregated tag-based profile visualized as a tag cloud. As Mypes enriches tag assignments with meta-information, stating to which WordNet category the corresponding tag belongs to, it is possible to filter tag-based profiles according to these WordNet categories. For example, Figure 6.4(a) also shows the aggregated tag cloud that is filtered to only display tags related to locations. For this kind of tag cloud, Mypes provides an alternative visualization: tags related to locations are mapped to country codes (using the *GeoNames* Web service[4]), which are sent to Google's visualization API to draw a geographical intensity map that highlights those countries that are frequently referenced by tags (referring to the country's name or to a city located in the country) in the profile (see bottom in Figure 6.4(a)). Mypes also features RDF export for these (specific facets of) tag-based profiles using the Tag Ontology[5] and SCOT[6] vocabulary. Figure 6.4(b) lists the RDF representation of the tag cloud related to locations that is visualized in Figure 6.4(a): for each tag the absolute usage frequency is specified.

In summary, Mypes makes the different types of profiles, tag-based as well as traditional

[4] `http://www.geonames.org/`
[5] `http://www.holygoat.co.uk/projects/tags/`
[6] `http://scot-project.org/scot/`



(a) Mypes profile

```
<scot:TagCloud
  rdf:about="http://mypes.groupme.org/mypes/user/116033/tagcloud/rdf">
  <rdfs:comment>Tag Cloud resulting from the user's latest tagging
       activities in Flickr, Delicious, StumbleUpon, and Last.fm.
  </rdfs:comment>
  <scot:createdBy
       rdf:resource="http://mypes.groupme.org/mypes/user/116033/rdf"/>

  <scot:tagSpace rdf:resource="http://mypes.groupme.org/mypes/"/>

  <scot:contains>
    <scot:Tag>
      <tags:name>Hannover</tags:name>
      <scot:ownAFrequency>13</scot:ownAFrequency>
    </scot:Tag>
  </scot:contains>

  <scot:contains>
    <scot:Tag>
      <tags:name>italy</tags:name>
      <scot:ownAFrequency>13</scot:ownAFrequency>
    </scot:Tag>
  </scot:contains>

  <scot:contains>
    <scot:Tag>
      <tags:name>turin</tags:name>
      <scot:ownAFrequency>7</scot:ownAFrequency>
    </scot:Tag>
  </scot:contains>

  <scot:contains>
    <scot:Tag>
      <tags:name>moscow</tags:name>
      <scot:ownAFrequency>5</scot:ownAFrequency>
    </scot:Tag>
  </scot:contains>
  ...
</scot:TagCloud>
```

(b) Mypes profile exported to FOAF

Figure 6.4: Aggregation of tag-based profile information: (a) visualization of aggregated profile for end-users and (b) FOAF export of Mypes profile.

profiles, available in RDF which allows third-party applications to benefit from profile aggregation, alignment and enrichment.

6.2.2 Evaluation of the Mypes Service

In order to evaluate the accuracy and runtime behavior of Mypes we crawled the public profiles of more than 100000 distinct users via Google's profile search[7]. From this collection we obtained (i) 338 users who have specified a *traditional profile* at Facebook, LinkedIn, Twitter, Flickr, and Google profiles, (ii) 139 users who have a *tag-based profile* their Flickr, StumbleUpon, Delicious, and Last.fm account, and 53 users who have an account at all services mentioned before. Given these users and their profile data, we first evaluate the Mypes service and particularly answer the two questions.

1. How accurate does the Mypes service work?
2. How fast does the Mypes service work?

[7] `http://www.google.com/profiles?q=query`

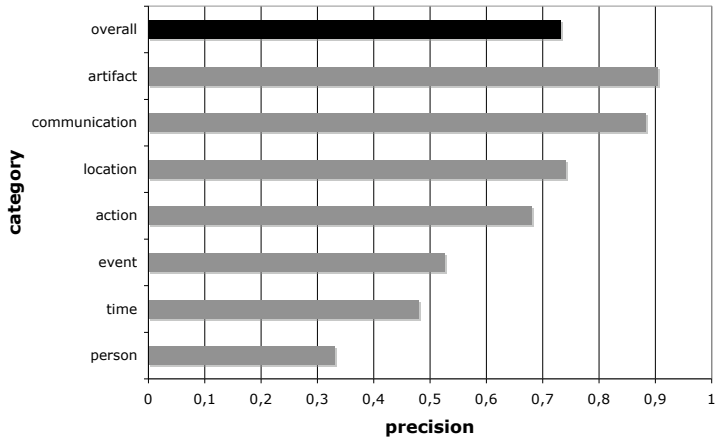


Figure 6.5: Precision of semantic enrichment with WordNet categories.

In the following subsections we will answer the questions above. The general benefits of the Mypes approach to user modeling will be discussed in Section 6.2.3 and 6.2.4. The impact of Mypes user modeling techniques on personalization will be evaluated will be investigated in Section 6.3.

Accuracy of Mypes

The accuracy of Mypes depends on the the accuracy of the single Mypes components which are depicted in Figure 6.1.

1. The precision of the *account mapping* is influenced by the users who link their different online accounts in their Google profile. It is possible that users claim that some online account belongs to them even if it does belong to another user (see *My Links* at Google Profile editing page[8]). However, for the 53 users we crawled who linked the nine services mentioned in Table 6.1 this did not happen.
2. We assume that the accuracy of the *profile aggregation* is always 100% because it could only drop below 100% if a service provider would deliver profile information that does not belong to the account for which Mypes is requesting information.
3. The *profile alignment* of traditional profiles does not affect the accuracy negatively as it is based on hand-crafted rules that map service-specific attributes to a uniform user model.
4. The *semantic enrichment* component is intended to add further value to the aggregated profiles: tag-based profiles are enriched with metadata that specifies to which WordNet category a tag belongs to. Such metadata might be wrong. Hence, we analyze the accuracy of the semantic enrichment in more detail.

[8] `http://www.google.com/profiles/me/editprofile?edit=s`

We randomly selected 30 users, inspected all tag-based Mypes profiles and marked whether the attached metadata—i.e. the WordNet category assigned to a tag—is correct. On average, the tag-based profiles contained 159.4 tags. Figure 6.5 lists the precision of the semantic enrichment: the number of *correct* WordNet category assignments divided by the *overall* number of WordNet category assignments.

The overall precision of the semantic enrichment is 73.1%. However, the quality varies strongly with the particular WordNet category. For example, regarding tags related to *artifacts* (e.g., bike) or *communication* (e.g., hypertext, web) the accuracy is best with 90.5% and 88.2% respectively. By contrast, 33.1% precision for tags related to *persons* (e.g., me, george) is rather poor.

In summary, we discover that the accuracy of Mypes depends on the single components. Account mapping, profile aggregation and profile alignment are based on hand-crafted rules and thus do not influence the accuracy negatively. The semantic enrichment which automatically attaches semantics to the tag-based profiles produces a high precision of 73.1%. Possible future research might focus on optimizing precision of the semantic enrichment and extend the enrichment by means of DBpedia URIs as done in TagMe! (see Section 3.4).

Runtime Analysis

Given the 30 users randomly selected users from the previous section, we measured runtime behavior of Mypes. Figure 6.6 summarizes the results of this evaluation.

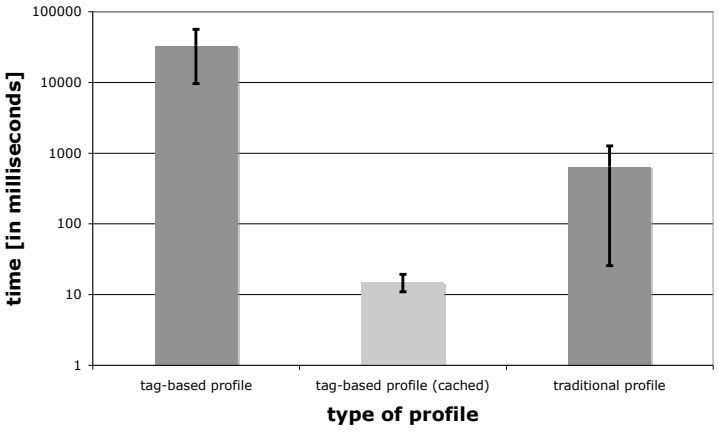


Figure 6.6: Average time (in milliseconds on a logarithmic scale) required for obtaining tag-based and traditional profiles and the corresponding standard deviation.

The aggregation of traditional profiles took, on average, 645 milliseconds and is therewith much faster then gathering the tag-based profiles which took, on average, 32830 milliseconds. The huge difference can be explained by the high number of tagging activities: Mypes considered, on average, 526.3 tagging activities (= tag assignments) to construct

the tag-based profiles which required to call the service APIs multiple times to obtain the required data. Mypes thus caches tag-based profiles (cf. Figure 6.1) which improves the performance significantly as depicted in Figure 6.6. Once a user is thus known to Mypes, runtime is not an issue, because profile data can continuously be synchronized with the Mypes data repository.

In summary, Mypes aggregates traditional profiles very fast (less than one second) while the aggregation of tag-based profiles works slowly. Mypes therefore provides caching functionality which allows for continuous synchronization of the Mypes profile repository with the profiles available in the Social Web systems and reduces the runtime of answering Mypes profile requests significantly.

6.2.3 Analysis of Distributed Traditional Profiles

Currently, users need to manually enter their profile attributes in each separate Web system. These attributes—such as the user's *full name*, current *affiliations*, or the *location* they are living at—are particularly important for social networking services such as LinkedIn or Facebook, but may be considered as less important in services such as Twitter. In our analysis, we measure to which degree users fill in their profile attributes in different services. To investigate the benefits of profile aggregation in particular we address the following questions.

1. How detailed do users fill in their public profiles at social networking and social media services?
2. Does the aggregated user profile reveal more information about a particular user than the profile created in some specific service?
3. Can the aggregated profile data be used to enrich an incomplete profile in an individual service?
4. To which extent can the service-specific profiles and the aggregated profile be applied to fill up standardized profiles such as FOAF [67] and vCard [86]?

Dataset

To answer the questions above, we crawled the public profiles of 116032 distinct users via the Mypes service introduced above. On average, the 116032 users linked 1.26 accounts while 70963 did not link any account.

For our analysis on traditional profiles we were interested in popular services where users can have public profiles. We therefore focused on the social networking services Facebook and LinkedIn, as well as on Twitter, Flickr, and Google. Table 6.2.3 lists the number of public profiles and the concrete profile attributes we obtained from each service. We did not consider private information, but only crawled attributes that were

Service	# crawled profiles	crawled profile attributes
Facebook	3080	nickname, first/last/full name, photo, email (hash), homepage, locale settings, affiliations
LinkedIn	3606	nickname, first/last/full name, about, homepage, location, interests, education, affiliations, industry
Twitter	1538	nickname, full name, photo, homepage, blog, location
Flickr	2490	nickname, full name, photo, email, location
Google	15947	nickname, full name, photo, about, homepage, blog, location

Table 6.2: Number of public profiles as well as the profile attributes that were crawled from the different services.

publicly available. Among the users for whom we crawled the Facebook, LinkedIn, Twitter, Flickr, and Google profiles were 338 users who had an account at all five different services.

Completeness of Individual and Aggregated Profiles

The completeness of the profiles varies from service to service. The public profiles available in the social networking sites Facebook and LinkedIn are filled more accurately than the Twitter, Flickr, or Google profiles—see Figure 6.7. Although Twitter does not ask many attributes for its user profile, users completed their profile up to just 48.9% on average. In particular the *location* and *homepage*—which can also be a URL to another profile page, such as MySpace—are omitted most often. By contrast, the average Facebook and LinkedIn profile is filled up to 85.4% and 82.6% respectively. Obviously, some user data is replicated at multiple services: name and profile picture are specified at nearly all services, location was provided at 2,9 out of five services. However, inconsistencies can be found in the data: for example, 37.3% of the users' *full names* in Facebook are not exactly the same as the ones specified at Twitter.

For each user we aggregated the public profile information from Facebook, LinkedIn, Twitter, Flickr, and Google, i.e. for each user we gathered attribute-value pairs and mapped them to a uniform user model. Aggregated profiles reveal more facets (17 distinct attributes) about the users than the public profiles available in each separate service. On average, the completeness of the aggregated profile is 83.3%: more than 14 attributes are filled with meaningful values. As a comparison, this is 7.6 for Facebook,

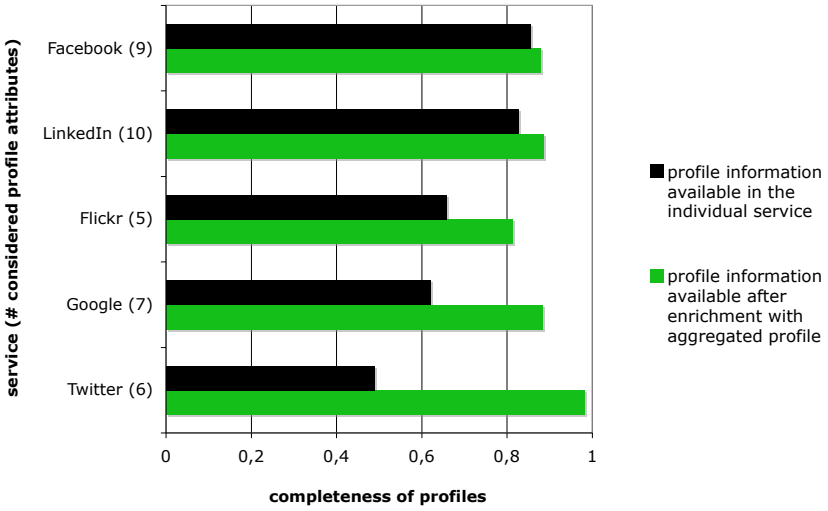


Figure 6.7: Completing service profiles with aggregated profile data. Only the 338 users who have an account at each of the listed services are considered.

8.2 for LinkedIn and 3.3 for Flickr. Aggregated profiles therewith reveal significantly more information about the users than the public profiles of the single services.

Further, profile aggregation enables completion of the profiles available at the specific services. For example, by enriching the incomplete Twitter profiles with information gathered from the other services, the completeness increases to more than 98% (see Figure 6.7): profile fields that are often left blank, such as location and homepage, can be obtained from the social networking sites. Moreover, even the rather complete Facebook and LinkedIn profiles can benefit from profile aggregation: LinkedIn profiles can, on average, be improved by 7%, even though LinkedIn provides three attributes—*interests*, *education* and *industry*—that are not in the public profiles of the other services (cf. Figure 6.1).

In summary, profile aggregation results in an extensive user profile that reveals more information than the profiles at the individual services. Moreover, aggregation can be used to fill in missing attributes at the individual services.

FOAF and vCard Generation

In most Web 2.0 services, user profiles are primarily intended to be presented to other end-users. It would also be very practical to use the profile data to generate FOAF [67] profiles or vCard [86] entries that can be fed into applications such as Outlook, Thunderbird or FOAF Explorer.

Figure 6.1 lists the attributes each service can contribute to fill in a FOAF or vCard profile, if the corresponding fields are filled out by the user. Figure 6.8 shows to which degree the real service profiles of the 338 considered users can actually be applied to fill in the corresponding attributes with adequate values.

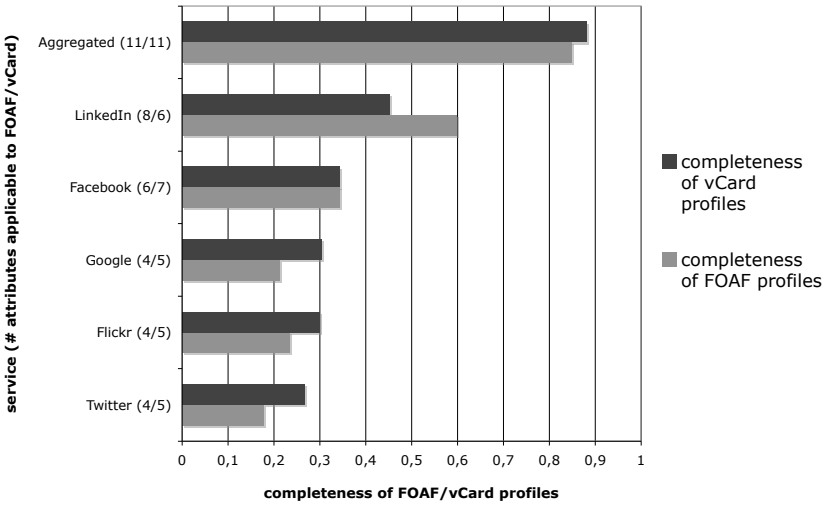


Figure 6.8: Completing FOAF and vCard profiles with data from the actual user profiles.

Using the aggregated profile data of the users, it is possible to generate FOAF profiles and vCard entries to an average degree of more than 84% and 88% respectively—the corresponding attributes are listed in Figure 6.1. Google, Flickr and Twitter profiles provide much less information applicable to fill the FOAF and vCard details. Although Facebook and LinkedIn both provide seven attributes that can potentially be applied to generate the vCard profile, it is interesting to see that the actual LinkedIn user profiles are more valuable and produce vCard entries with average completeness of 45%; using Facebook as a data source this is only 34%. In summary, the aggregated profiles are thus a far better source of information to generate FOAF/vCard entries than the service-specific profiles.

Result Summary

Our analysis of the user profiles distributed across the different services point out several advantages of profile aggregation and motivate the intertwining of profiles on the Web. With respect to the key questions raised at the beginning of the section, the main outcomes can be summarized as follows.

1. Users fill in their public profiles at social networking services (Facebook, LinkedIn) more extensively than profiles at social media services (Flickr, Twitter) which can possibly be explained by differences in purpose of the different systems.
2. Profile aggregation provides multi-faceted profiles that reveal significantly more information about the users than individual service profiles can provide.
3. The aggregated user profile can be used to enrich incomplete profiles in individual services, to make them more complete.

4. Service-specific profiles as well as the aggregated profiles can be applied to generate FOAF profiles and vCard entries. The aggregated profile represents the most useful profile, as it completes the FOAF profiles and vCard entries to 84% and 88% respectively.

As user profiles distributed on the Web describe different facets of the user, profile aggregation brings some advantages: users do not have to fill their profiles over and over again; applications can make use of more and richer facets/attributes of the user (e.g. for personalization purposes). However, our analysis shows also the risk of intertwining user profiles. For example, users who deliberately leave out some fields when filling their Twitter profile might not be aware that the corresponding information can be gathered from other sources.

6.2.4 Analysis of Distributed Tag-based Profiles

In our analysis on tag-based profiles (see Definition 5.2) we examine the nature of these profiles in different systems. Again, we investigate the the benefits of aggregating profile data and answer the following questions.

1. What kind of tag-based profiles do individual users have in the different systems?
2. Does the aggregation of tag-based user profiles reveal more information about the users than the profiles available in some specific service?
3. Is it possible to predict tag-based profiles in a system, based on profile data gathered from another system?

Individual Tagging Behavior across different Systems

From the 116032 users, 139 users linked their Flickr, StumbleUpon, *and* Delicious accounts. For these users, we crawled 78412 tag assignments that were performed on the 200 latest images (Flickr) or bookmarks (Delicious and StumbleUpon). Table 6.3 lists the corresponding tagging statistics. Overall, users tagged more actively in Delicious than in the other systems: more than 75% of the tagging activities originate from Delicious, 16.3% from StumbleUpon and 5% from Flickr. The usage frequency of the distinct tags shows a typical power-law distribution in all three systems, as well as in the aggregated set of tag assignments: while some tags are used very often, the majority of tags is used rarely or even just once.

On average, each user provided 564.12 tag assignments across the different systems. The user activity distribution corresponds to a gaussian distribution: 26.6% of the users have less than 200 tag assignments, 10.1% have more than 1000 and 63.3% have between 200 and 1000 tag assignments. Interestingly, people who actively tagged in one system do not necessarily perform many tag assignments in another system. For example, none

	Flickr	StumbleUpon	Delicious	Overall
tag assignments	3781	12747	61884	78412
distinct tags	691	2345	11760	13212
tag assignments per user	27.2	91.71	445.21	564.12
distinct tags per user	5.22	44.42	165.83	171.82

Table 6.3: Tagging statistics of the 139 users who have an account at Flickr, StumbleUpon, and Delicious.

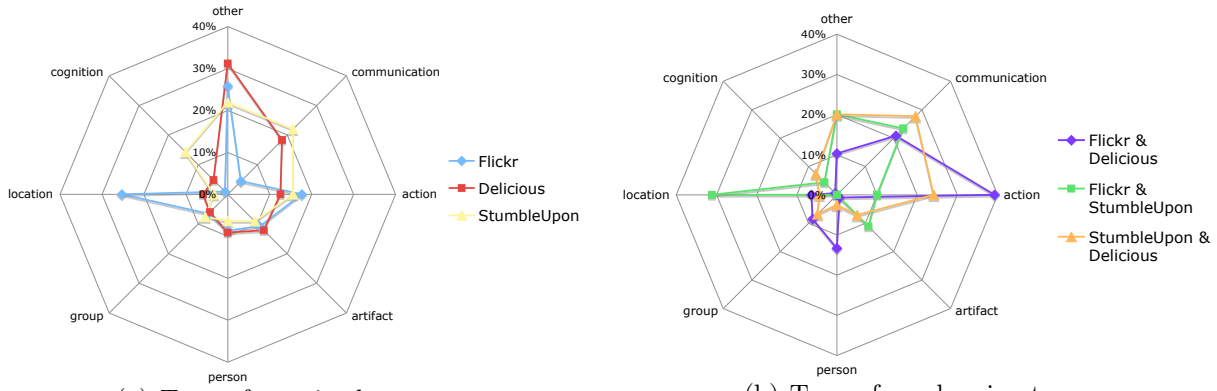
of the top 5% taggers in Flickr or StumbleUpon is also among the top 10% taggers in Delicious.

This observation of focussed tagging behavior across different systems again suggests potential advantages of profile aggregation for current tagging systems: given a sparse tag-based user profile focussing on specific topics, the consideration of profiles produced in other systems might be used to tackle sparsity problems and cover the different topics the user refers to in the specific systems.

Commonalities and Differences in Tagging Activities

In order to analyze commonalities and differences of the users' tag-based profiles in the different systems, we mapped tags to Wordnet categories and considered only those 65% of the tags for which such a mapping exists. Figure 6.9(a) shows that the type of tags in StumbleUpon and Delicious are quite similar, except for *cognition* tags (e.g., research, thinking), which are used more often in StumbleUpon than in Delicious. For both systems, most of the tags—21.9% in StumbleUpon and 18.3% in Delicious—belong to the category *communication* (e.g., hypertext, web). By contrast, only 4.4% of the Flickr tags refer to the field of communication; the majority of tags (25.2%) denote locations (e.g., Hamburg, tuscany). *Action* (e.g., walking), *people* (e.g., me), and *group* tags (e.g., community) as well as words referring to some *artifact* (e.g., bike) occur in all three systems with similar frequency. However, the concrete tags seem to be different. For example, while artifacts in Delicious refer to things like "tool" or "mobile device", the artifact tags in Flickr describe things like "church" or "painting". This observation is supported by Figure 6.9(b), which shows the average overlap of the individual category-specific tag profiles. On average, each user applied only 0.9% of the Flickr artifact tags tags also in Delicious. For Flickr and Delicious, action tags allocate the biggest fraction of overlapping tags. It is interesting to see that the overlap of location tags between Flickr and StumbleUpon is 31.1%, even though location tags are used very seldomly in StumbleUpon (3.3%, as depicted in Figure 6.9(a)). This means that if someone utilizes a location tag in StumbleUpon, it is likely that she will also use the same tag in Flickr.

Having knowledge on the different (aggregated) tagging facets of a user opens the door for interesting applications. For example, a system could exploit StumbleUpon tags



(a) Type of tags in the systems

(b) Type of overlapping tags

Figure 6.9: Tag usage characterized with Wordnet categories: (a) Type of tags users apply in the different systems and (b) type of tags individual users apply in two different systems.

referring to locations to recommend Flickr pictures even if the user's Flickr profile is empty. In the subsequent sections we will present an approach that takes advantage of the faceted tag-based profiles for predicting tagging behavior and recommending tags.

Aggregation of Tagging Activities

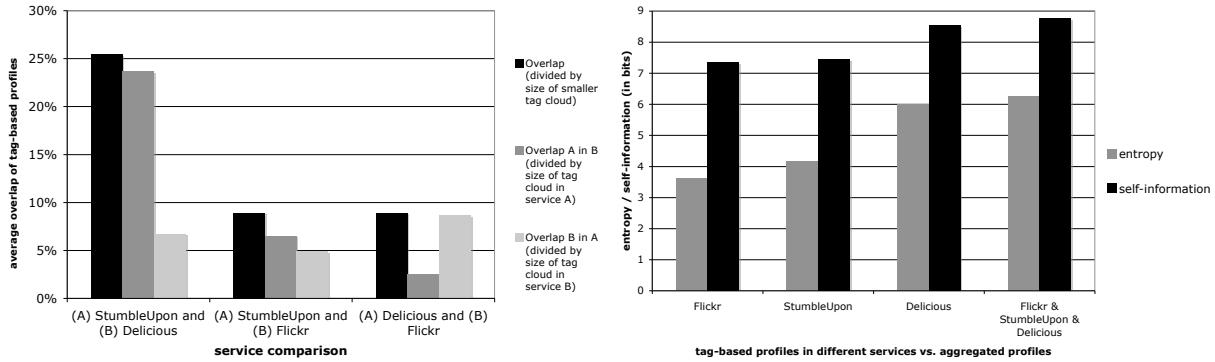
To analyze the benefits of aggregating tag-based profiles in more detail we measure the information gain, entropy and overlap of the individual profiles. Figure 6.10(a) describes the average overlap with respect to three different metrics: given two tag-based profiles A and B, the overlap is (1) $overlap = \frac{A \cap B}{min(|A|,|B|)}$, (2) $overlap_{AinB} = \frac{A \cap B}{|A|}$, or (3) $overlap_{BinA} = \frac{A \cap B}{|B|}$. For example, $overlap_{AinB}$ denotes the percentage of tags in A that also occur in B.

The overlap of the tag-based profiles produced in Delicious and StumbleUpon is significantly higher than the overlap of service combinations that include Flickr. However, on average, a user still just applies 6.8% of her Delicious tags in StumbleUpon as well, which is approximately as high as the percentage of tags a StumbleUpon user also applies in Flickr. Overall, the tag-based user profiles do not overlap strongly. Hence, users reveal different facets of their profiles in the different services.

Figure 6.10(b) compares the average entropy and self-information of the tag-based profiles obtained from the different services with the aggregated profile. The entropy of a tag-based profile T, which contains of a set of tags t, is computed as follows.

$$entropy(T) = \sum_{t \in T} p(t) \cdot self\text{-}information(t) \tag{6.1}$$

In Equation 6.1, $p(t)$ denotes the probability that the tag t was utilized by the corre-



(a) Overlap of tag-based profiles (b) Entropy and self-information

Figure 6.10: Aggregation of tag-based profiles: (a) average overlap and (b) entropy and self-information of service-specific profiles in comparison to the aggregated profiles.

sponding user. Self-information is the logarithm of $p(t)$ multiplied by -1:

$$\textit{self-information}(t) = -log_2(p(t)) \tag{6.2}$$

Using base 2 for the computation of the logarithm allows for measuring self-information as well as entropy in bits. For modeling the probability $p(t)$ that a tag t appears in a given user profile, we apply the individual usage frequencies of the tags, i.e. for a specific user u the usage frequency of tag t is the fraction of u's tag assignments where u referred to t.

To clarify the meaning of entropy and self-information in context of the tag-based user profiles, we apply the metrics to example profiles that belong to a specific user, whom we call Bob (see Table 6.4).

The self-information and entropy of the example profiles listed in Table 6.4 depend on the number of tags that appear in the profiles and the corresponding usage frequencies as well. Bob's tag-based profiles in Flickr (*flickr-bob*) and StumbleUpon (*stumble-bob*) both contain two distinct tags. However, the self-information of the StumbleUpon profile is higher than the self-information of the Flickr profile as tags appear with different probabilities (e.g., *p(research) = 8/12* and *p(semantic web) = 4/12*) instead of being uniformly distributed (e.g., *p(hannover) = 8/16* and *p(italy) = 8/16*). In contrast, entropy is higher for those tag-based profiles having a rather uniform distribution and implying a higher level of randomness. The aggregation of the three profiles listed in Table 6.4 (*mypes-bob*) reveals the highest self-information and entropy.

In Figure 6.10(b), we summarize self-information by building the average of the mean self-information of the users' tag-based profiles. Among the service-specific profiles, the tag-based profiles in Delicious, which also have the largest size, bear the highest entropy and average self-information. By aggregating the tag-based profiles, self-information

profile	tag (frequency)	self-information	entropy
flickr-bob	hannover (8) italy (8)	1	1
stumble-bob	research (8) semantic web (4)	1.08	0.92
delicious-bob	semantic web (10) social web (5) hannover (3) user modeling (3)	2.19	1.8
mypes-bob (aggregated)	semantic web (14) hannover (11) italy (8) research (8) social web (5) user modeling (3)	2.75	2.44

Table 6.4: Entropy and average self-information of example profiles. The tag-based profiles contain for each tag the corresponding usage frequency which is applied to model the probability p(t) that the tag t appears in the user profile.

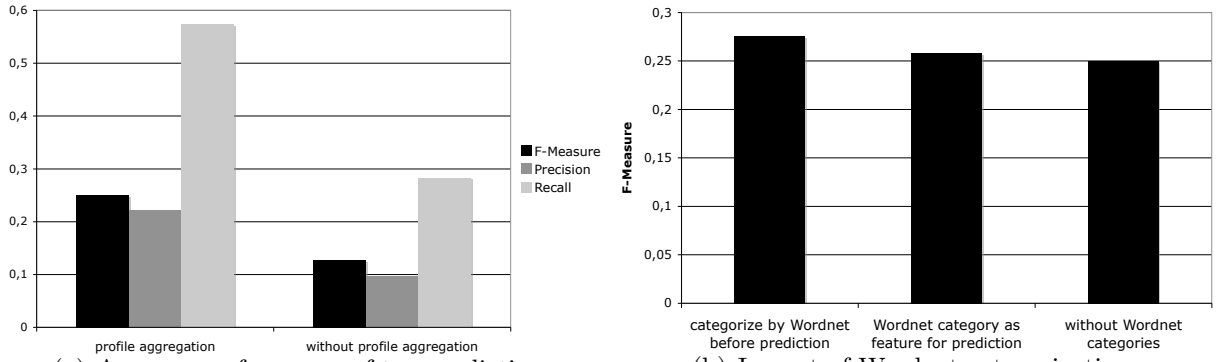
increases clearly by 19.5% and 17.7% with respect to the Flickr and StumbleUpon profiles respectively. Further, the tag-based profiles in Delicious can benefit from the profile aggregation as the self-information would increase by 2.7% (from 8.53 bit to 8.76 bit) which is also considerably higher, considering that self-information is measured in bits (e.g., with 8.53 bits one could describe 370 states while 8.76 bits allow for decoding of 434 states).

Aggregation of tag-based profiles thus reveals more valuable new information about individual users than focusing just on information from single services. However, some fraction of the profiles also overlap between different systems, as depicted in Figure 6.10(a). In the next section we analyze whether it is possible to predict those overlapping tags.

Prediction of Tagging Behavior

Systems that rely on user data usually have to struggle with the *cold start problem*; especially those systems that are infrequently used or do not have a large base of users require solutions to that problem. In this section we investigate the applicability of profile aggregation. Therefore, we evaluate different approaches with respect to the following task.

Task: Tag prediction *Given a set of tags that occur in the tag-based profile of user u in system A, the task of the tag prediction strategy is to predict those tags that will also occur in u's profile in system B.*



(a) Average performance of tag prediction

(b) Impact of Wordnet categorization

Figure 6.11: Performance of tag prediction: (a) with and without aggregation of tag-based profiles and (b) improving prediction performance (with profile aggregation) by means of Wordnet categorization.

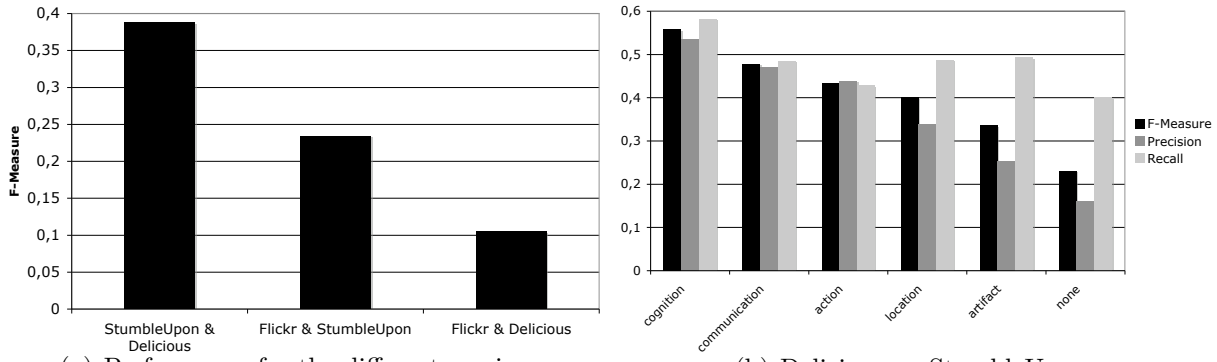
We measure the performance by means of *precision* (= correctly classified as overlapping tags divided by tags classified as overlapping tags) and *recall* (= correctly classified as overlapping tags divided by the number of overlapping tags) as specified in Definition 4.5 as well as *F-measure* which is defined as follows.

Definition 6.2 (F-measure) *The F-measure is the harmonic mean of recall and precision.*

$$F - measure = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{6.3}$$

Our intention is not to find the best prediction algorithm, but to examine the impact of features extracted from profile aggregation. Hence, we apply a *Naive Bayes classifier*, which we feed with different features. The benchmark tag prediction strategy (*without profile aggregation*) bases its decision on a single feature: (F1) overall usage frequency of t in system B. In contrast, the strategy that makes use of *profile aggregation* also applies (F2) u's usage frequency of t in system A and (F3) size of u's profile in system A.

Figure 6.11(a) compares the average performance of both tag prediction strategies. For each of the 139 users and each service combination (Flickr → Delicious, Delicious → Flickr, StumbleUpon → Delicious, etc.) the strategies had to tackle the prediction task specified above. The benefits of the profile aggregation features are significant. The profile aggregation strategy performs—with respect to the f-measure—96.1% better than the strategy that does not benefit from profile aggregation (correspondingly, the improvement of precision and recall is explicit). Further, it is important to notice that the average percentage of overlapping tags is less than 4%. Thus, a random strategy, which simply guesses whether tag t will overlap or not (probability of 0.5), would fail with a precision lower than 2%.



(a) Performance for the different services

(b) Delicious → StumbleUpon

Figure 6.12: Tag prediction performance for specific services.

On average, the profile aggregation strategy can thus detect 57.4% of the tags in system A that will also be part of the tag-based profile in system B. The performance can further be improved by clustering the tag-based profiles according to Wordnet categories. Figure 6.11(b) shows that the consideration of Wordnet features—(F4) Wordnet category of t and (F5) relative size of corresponding Wordnet category cluster in u's profile—leads to a small improvement from 0.25 to 0.26 regarding the f-measure. However, if tag predictions are done for each Wordnet cluster of the profiles separately, the improvement is considerably high as the f-measure increases from 0.25 to 0.28.

Figure 6.12 shows the tag prediction performance (using features F1-5) focusing on specific service combinations. While tag predictions for Flickr/Delicious based on tag-based profiles from Delicious/Flickr perform quite weak, the predictions between Flickr and StumbleUpon show a much better performance (f-measure: 0.23). For the two bookmarking services, StumbleUpon and Delicious, which also have the highest average overlap (cf. Figure 6.10(a)), tag prediction works best with f-measure of 0.39 and precision of 0.36. Figure 6.12(b) illustrates for what kind of tags prediction performs best between Delicious and StumbleUpon. For tags that cannot be assigned to a Wordnet category (*none*), the precision is just 16% while recall of 40% might still be acceptable. However, given tags that can be mapped to Wordnet categories, the performance is up to 0.57 regarding f-measures. Given *cognition* tags (e.g., search, ranking) of a particular user u, the profile aggregation strategy, which applies the features F1-5, can predict the cognition tags u will use in StumbleUpon with a precision of nearly 60%: even if a user has not performed any tagging activity in StumbleUpon, one could recommend 10 cognition tags out of which 6 are relevant for u.

Result Summary

The results of our analyses and experiments indicate several benefits of aggregating and interweaving tag-based user profiles. We showed that users reveal different types of facets (illustrated by means of WordNet categories) in the different systems. By combining tag-

based profiles from Flickr, StumbleUpon, and Delicious, the average self-information of the profiles increases significantly. Although the tag-based service-specific profiles overlap just to a small degree, we proved that the consideration of profile data from other sources can be applied to solve cold start problems. In particular, we showed that the profile aggregation strategy for predicting tag-based profiles significantly outperforms the benchmark that does not incorporate profile features from other sources.

6.2.5 Synopsis

In this section we introduced our approach to model users across folksonomy system boundaries. The core idea is to aggregate and align user profile data from different systems. We developed the so-called Mypes service that allows for profile linkage, aggregation, alignment and enrichment of public user profiles. The evaluation of the service showed the practicability of Mypes with respect to accuracy and runtime. Caching was implemented to optimize the runtime performance and avoid delays that would occur for real time aggregation of large tag-based profiles.

Further, we analyzed our approach in detail and revealed several benefits of interweaving public profile data on the Social Web. For both explicitly provided profile information (e.g. name, hometown, etc.) and rather implicitly provided tag-based profiles (e.g. tags assigned to bookmarks), the aggregation of profile data from different services (e.g, LinkedIn, Facebook, Flickr, etc.) reveals significantly more facets about the individual users than one can deduce from the separated profiles.

Our experiments show the advantages of interweaving distributed user data for various applications, such as completing service-specific profiles, generating FOAF or vCard profiles, producing multi-faceted tag-based profiles, and predicting tag-based profiles to solve cold start problems. End-users and application developers can immediately benefit from our research by using the Mypes service.

In the context of our experiments presented in Section 6.2.3 and 6.2.4 we further discovered correlations between traditional and tag-based profiles. For example, we analyzed whether tag-based profiles conform to the skills users specified at LinkedIn and discovered that 76.2% of the users applied at least one of the, on average, 8.56 LinkedIn skills also as a tag in Delicious. Further, we found first evidence that for users, who belong to the same group based on their social networking profile (in particular location and industry), the similarities between the tag-based profiles is higher than for users belonging to different groups. There are opportunities for future work to investigate how explicitly provided profile data can be exploited in folksonomy systems, and how tag-based profiles can be semantically enhanced to enrich traditional social networking profiles (cf. Chapter 7).

6.3 Personalized Recommendations based on Cross-System User Modeling

In Chapter 5 we discovered that our user and context modeling strategies enable folksonomy systems to provide personalization such as personalized search or tag recommendations. The data that was applied to construct these user and context representations originated from the folksonomy system interested in providing personalized services. For example, to provide recommendations to the GroupMe! users we utilized data available in GroupMe!. In this chapter we analyze whether we can also take advantage from data distributed on the Web. Our goal is to model users *in context* of their Social Web activities to improve the quality of personalization in particular systems. In Section 6.2 we analyzed already the nature of user profile traces distributed on the Social Web and introduced strategies for modeling users across folksonomy system boundaries. In this section we will evaluate these strategies with respect to tag and resource recommendation tasks, which we interpret as ranking problems.

Problem 5 (Tag Recommendation) *Given a tag-based user profile $P(u)$, the personomy of the user $\mathbb{P}_u = (T_u, R_u, I_u)$ and a set of tags T, which are not explicitly connected to u ($T_u \cap T = \emptyset$), the task of the tag recommendation strategies is to rank these tags $t \in T$ so that tags that are most relevant to the user u appear at the very top of the ranking.*

In contrast to the tag recommendation task specified in Chapter 5, tag recommendations are computed for specific users independently from any resources. Hence, the application we have in mind is to suggest tags which people can use to explore content of a folksonomy system. A user profile should be modeled by means of a user-specific tag-based profile $P_U(u)$ (cf. Definition 5.2). Further, $P_U(u)$ might be an aggregation of tag-based profiles (cf. Definition 6.1) or might contain only a subset of tags ($P_U(u)@k$) used by u in some tagging system(s). The resource recommendation challenge can be defined accordingly.

Problem 6 (Resource Recommendation) *Given a tag-based user profile $P(u)$, the personomy of the user $\mathbb{P}_{u,target} = (T_u, R_u, I_u)$ and a set of resources R, which are not explicitly connected to u ($R_u \cap R = \emptyset$), the task of the resource recommendation strategies is to rank these resources $r \in R$ so that resources that are most relevant to the user u appear at the very top of the ranking.*

In this section we investigate how the cross-folksonomy user modeling strategies (see Section 6.2) impact these tag and resource recommendation tasks. We concentrate on the user modeling challenge instead of tuning the overall performance of the recommender algorithms. Hence, the core challenge we tackle can be defined as follows.

Problem 7 (User Modeling) *Given a user u, the user modeling strategies have to construct a tag-based profile $P_U(u)$ so that the performance of the tag and resource recommenders is maximized.*

We will thus elicit one algorithm (see Section 6.3.1) in combination with different user modeling strategies with respect to the recommender tasks. Further, we will particularly focus on *cold-start situations* [196] where new users come into play that have not performed any tagging activity in the system.

6.3.1 Mypes Recommender Algorithms

The tag and resource recommendation tasks are defined as ranking problems and can thus be tackled by ranking algorithms introduced at the beginning of this thesis. As we are interested in evaluating the quality of different approaches for modeling users across folksonomy system boundaries, we will apply a standard ranking algorithm—FolkRank (see Section 2.2.2)—which we will input with profiles generated by the different user modeling strategies. Accordingly to the tag recommendation experiments presented in Section 5.4 we can define a generic recommender algorithm that expects the actual ranking and user modeling strategy as input (Definition 5.6).

Definition 6.3 (Generic Recommender Algorithm) *The* generic recommender algorithm *computes a ranked list of entities appropriate to a user u by exploiting a given ranking algorithm and a given user modeling strategy.*

1. ***Input:*** *ranking strategy s, user modeling strategy um, user u*
2. $P_U(u) = um.modelUser(u)$ *(compute user profile)*
3. $\tau = s.rank(P_U(u))$ *(rank entities with respect to user profile)*
4. ***Output:*** τ *(ranked list of entities)*

For the tag and resource recommendation tasks, the output of the generic ranking algorithm is a ranked list of tags and resources, i.e. a set of weighted tags or resources. In the following recommender experiments we will compare user modeling strategies that all make use of profile aggregation, but differ in the selection of the source profiles applied to construct an aggregated tag-based profile.

As users are modeled in context of their Social Web environment, there are several tag-based profiles available for an individual user, which originate from the different folksonomy systems the user is actively participating in. For example, when recommending Delicious bookmarks to user u, user modeling strategy um_a might consider

only u's tag-based Delicious profile while another strategy um_b might aggregate u's Delicious and StumbleUpon profiles. In detail, we will analyze the following types of user modeling strategies.

Target Profile. The traditional user modeling approach is to consider only the user's tag-based profile from the *target system*, i.e. the folksonomy system where recommendations should be provided. Hence, the *target profile*, $P_{U,target}(u)$, conforms to the user-specific tag cloud specified in Definition 5.2 and $P_{U,target}(u)@k$ denotes the tag-based user profile that contains the k tags most frequently used by u.

Popular Profile. If the target profile $P_{U,target}(u)$ is rather sparse or even empty, one has to find other sources of information that are applicable to generate a user profile. Therefore, we define another baseline strategy that considers the most popular tags within the target folksonomy system (which provides folksonomy $\mathbb{F}$ with users U, see Definition 2.1) and computes the tag-based profile by aggregating the profiles of all users $u_i \in U$ different from u: $P_{popular}(u) = \textit{aggregate profiles } P_U(u_i) \textit{ where } u \neq u_i$. In our experiments we apply top k profiles $P_{popular}(u)@k$ and set $k = 150$.

Mypes Profile. The so-called *Mypes profile* aggregates tag-based profiles of user u that originate also from other folksonomy systems. Hence, the tag-based Mypes profile is an aggregation of profiles $P_{U,service}$ where *service* can differ from the *target system*: $P_{U,Mypes}(u) = \textit{aggregate tag-based profiles } P_{U,service_i}(u) \textit{ from different services } i$.

In the tag and resource recommendation experiments we further mix the above strategies. For example, we combine the *Mypes profile* $P_{U,Mypes}(u)$ with the most popular tag representation $P_{popular}(u)$. In general, these user modeling strategies produce tag-based profiles that serve as input for the generic recommender strategy and the given ranking strategy in particular (see Definition 6.3). Our approach to utilize such profiles as preferences for FolkRank is to adapt the construction of the folksonomy graph $\mathbb{G}_{\mathbb{F}}$ (see Definition 2.2) represented by the adjacency matrix A (cf. Section 2.2.2). In particular, with respect to a given profile $P_U(u)$ we adjust the way for computing the weights of edges between users and tags $w(u_i, t_j)$.

$$w(u_i, t_j) = \begin{cases} |\{r \in R : (u, t, r) \in Y\}| & \text{if } u_i \neq u \\ (t_j, w_x) & \text{if } u_i = u \wedge (t_j, w_x) \in P_U(u) \\ 0 & \text{otherwise} \end{cases}$$

Further, when computing tag and resource recommendations for a specific user u with FolkRank, we set the preference vector p so that the dimension associated with u is equal to 1 while all other dimensions are set to zero. Finally, we compute run the FolkRank algorithm as specified in Definition 2.5 and rank the tags and resources according to their the FolkRank scores in order to provide tag and resource recommendations respectively.

	Flickr	Delicious	Stumble Upon	All
distinct tags	18240	21239	8663	39399
TAS	171092	155230	61464	387786
distinct tags/user	90.05	192.67	90.95	349.04
TAS/user	532.99	483.58	191.48	1208.06

Table 6.5: Tagging statistics for the 321 users who have an account at Flickr, Delicious, and StumbleUpon.

6.3.2 Dataset Characteristics

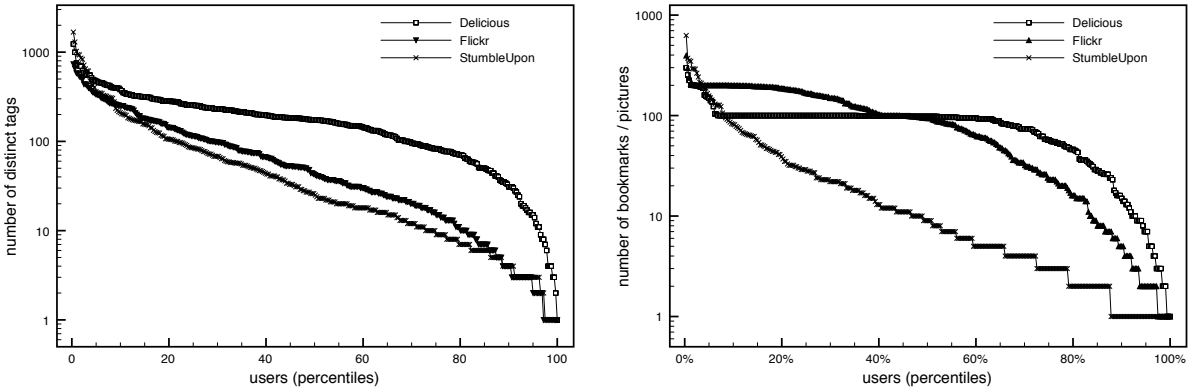
To analyze the performance of the recommender strategies, we crawled public profiles of 421188 distinct users via the Mypes service (see Section 6.2.1). We applied the following strategy to crawl profiles: (1) we used common first names (e.g., *John*, *Peter*, *Mary*, *Sarah*) as search query at Google's profile search interface[9] to obtain profile URIs constituting the input for Mypes and (2) we crawled the profiles of friends that were linked by users which were obtained in the first step.

For our analysis we were interested in users having accounts at several social tagging systems. 142184 of the 421188 users did not link any other account. On average, the remaining 279004 users linked 3.1 of their online accounts and Web sites. However, only a few users linked the profiles they have at social tagging platforms: 14450 users specified their Flickr account, 2005 users linked their Delicious account and 813 users listed their StumbleUpon profile. Among these users, 1467 people had a Flickr *and* Delicious profile and only 321 users had a tag-based profile at all the three different systems, i.e. Flickr *and* Delicious *and* StumbleUpon.

The tagging statistics of these 321 users having tag-based profiles at Flickr, Delicious, *and* StumbleUpon are listed in Table 6.3.2. Overall, these users performed 387786 tag assignments (TAS). In Flickr users tagged most actively with an average of 532.99 tag assignments, followed by Delicious (483.58 TAS) and StumbleUpon (191.48 TAS). It is interesting to see that Delicious tags constitute the largest vocabulary, even though most tagging activities were done in Flickr: the Delicious folksonomy contains 21239 distinct tags, while the Flickr folksonomy covers only 18240 distinct tags. Correspondingly, tag-based Delicious profiles have an average of 192.67 distinct tags in contrast to 90.05 distinct tags for the Flickr profiles.

Figure 6.13(a) shows the distribution of the number of distinct tags for the different

[9]Searching for Google profiles related to "john": `http://www.google.com/profiles?q=john`



(a) Number of distinct tags per user

(b) Tagged resources per user

Figure 6.13: Characteristics of tagging behavior: (a) size of tag-based profiles per user and (b) number of distinct resources each user annotated.

services. For more than 80% of the users the tag-based Flickr and StumbleUpon profile contains less than 200 distinct tags. In Delicious, people use a greater variety of tags as almost 40% of the users applied more than 200 tags. However, the fraction of tag-based profiles that contain more than 500 tags is for all services less than 5% while the majority of profiles is rather sparse.

The distribution of the number of resources annotated by the users differs slightly from the distribution of tags (see Figure 6.13(b)). Induced by Delicious API restrictions, there are many Delicious users for whom we crawled 100 bookmarks although the crawling process was repeated several times within a time period of two months. Hence, when we initiated Delicious bookmark crawling for the first time, Mypes was able to aggregate the complete bookmarking history. However, more than 20% of the users were inactive within the period of crawling so that the number of bookmarks did not grow further. For Flickr and StumbleUpon such restrictions were not given so that the distribution of the number of pictures and bookmarks corresponds to the actual behavior of the users: again less than 5% of the users annotated more than 200 resources while the majority of users tagged only few resources.

Overlap of Tag-based Profiles

Another remarkable feature of the dataset is that only a few tags occur in more than one service: less than 20% of the distinct tags were used in more than one system.

Figure 6.14 shows to which degree the profiles of the individual users in the different services overlap with each other. For each user u and each pair of service A and B, we

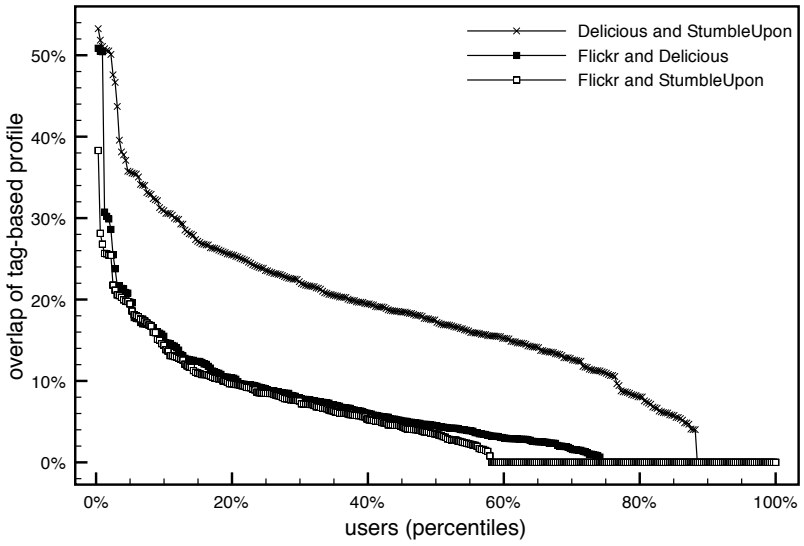


Figure 6.14: How much do the individual tag-based profiles overlap?

compute the overlap as follows:

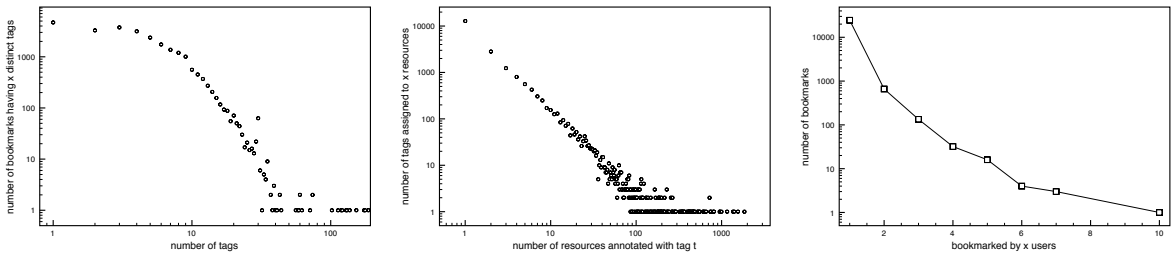
$$overlap(u_A, u_B) = \frac{1}{2} \cdot \left(\frac{|T_{u,A} \cap T_{u,B}|}{|T_{u,A}|} + \frac{|T_{u,A} \cap T_{u,B}|}{|T_{u,B}|}\right) \tag{6.4}$$

$T_{u,A}$ and $T_{u,B}$ denote the set of distinct tags that occur in the tag-based profile of user u in service A and B respectively. Hence, $|T_{u,A} \cap T_{u,B}|$ is the number of distinct tags that occur in both profiles, u_A and u_B. Figure 6.14 illustrates that the individual Delicious and StumbleUpon profiles have the biggest overlap. However, the overlap is rather small: for more than 50% of the users the overlap of their Delicious and StumbleUpon profiles is less than 20% and there exist only 6 users for whom the overlap is slightly larger than 50%. It is interesting that the overlap is so small, as in both Delicious and StumbleUpon the same type of resources are tagged, probably the tools are used for separate task. Flickr and StumbleUpon profiles offer the least overlap as for more than 40% the overlap is 0%.

In summary, the small overlaps between the individual tag-based profiles indicate that the computation of cold-start recommendations for some specific system is still a non-trivial task—even if profile information from other folksonomy systems is considered as well (see 6.2). We will show that our algorithms nevertheless manage to succeed in recommending tags and resources to new users.

Commonalities and Differences in Bookmarking Behavior

In Section 6.3.4, the resource recommendation experiments will focus on recommending Delicious bookmarks to users. Figure 6.15 characterizes these Delicious bookmarks.



(a) Tags per bookmark (b) Shared tags (c) Shared bookmarks

Figure 6.15: Delicious bookmarks: (a) number of bookmarks that are annotated with x distinct tags and (b) number of tags assigned to x different resources and (c) number of bookmarks that were bookmarked by x users.

The majority of bookmark resources have only a few tags (see Figure 6.15(a)). For example, more than 4500 of the resources are annotated with just one tag, whereas only 10 resources are annotated with more than 100 distinct tags. Figure 6.15(b) depicts the number of tags that are assigned to x different resources and shows that more than 12000 tags are just used once. Considering the tripartite folksonomy graph, which is exploited by the recommender algorithms, this means that more than 12000 tag nodes are each connected with one user and resource node only so that weighting of these nodes becomes difficult if no further preferences would be considered.

Figure 6.15(c) illustrates that the number of bookmarks shared among the 321 users is rather low. 24515 resources are bookmarked by just one user, 660 resources are bookmarked by two different users and solely one resource is bookmarked by 10 users. These number indicate that traditional collaborative recommender strategies, which recommend items based on user similarities computed via user-resource connections [193], would get problems because of too few connections between users and recommender strategies that also exploit user-tag and tag-resource connections are rather promising.

6.3.3 Tag Recommendation Experiment

Within the scope of the tag recommendation experiment, we evaluated the user modeling strategies by means of a *leave-many-out evaluation* [99]. For simulating a cold-start where a new user u registers to the folksonomy system A and is interested in tag recommendations, we removed all tag assignments Y_u performed by u in system A from the folksonomy. Each recommender strategy then had to compute tag recommendations. The quality of the recommendations was measured via *MRR* (Mean Reciprocal Rank), which indicates at which rank the first *relevant* tag occurs on average, *S@k* (success at rank k), which stands for the mean probability that a *relevant* tag occurs within the top k of the ranking, and *P@k* (precision at rank k), which represents the average proportion of *relevant* tags within the top k (cf. Section 5.3.3). We considered only those tags as

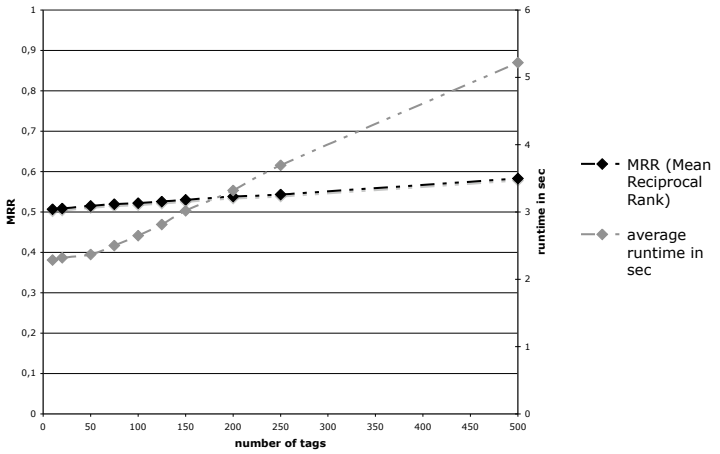


Figure 6.16: Adjustment of experimental setup (size of tag-based user profiles): recommendation quality (MRR) vs. runtime for computing these recommendations.

relevant to which the user u actually referred to in the tag assignments Y_u that were removed before computing the recommendations.

We ran the experiments for each of the 321 users, who actively contributed tags in Flickr, Delicious *and* StumbleUpon. To reduce the computation time required during the experiments for adjusting the folksonomy graph for each user, we limited the size of the tag-based profiles to 150 entries. As depicted in Figure 6.16, the size of the tag-based profile directly influences the runtime of adjusting the folksonomy graph, which had, for example, more than 45000 nodes for Delicious. In general, more profile information results in a better performance of the tag recommendations. However, with 150 entries and 3 seconds per folksonomy graph adjustment, we found a reasonable trade-off between runtime and recommendation quality.

We tested the statistical significance of our results with a two-tailed t-Test where the significance level was set to $\alpha = 0.01$. The null hypothesis H0 is that some user modeling strategy um_1 is as good as another strategy um_2 for computing tag recommendations, while H1 states that um_1 is better than um_2.

Cold-start tag recommendations

Figure 6.17 overviews the results for computing tag recommendations for the cold-start setting where in the *target system* there is no information available about the user to whom the personalized recommendations should be delivered. The diagram shows averaged results, i.e. for all users and all service constellations possible with a given user modeling strategy (cf. Section 6.3.1). For example, *Mypes (single service)*, which takes advantage of the user's profile available in another system different from the target system, is averaged over each users and each possible constellation such as "recommend

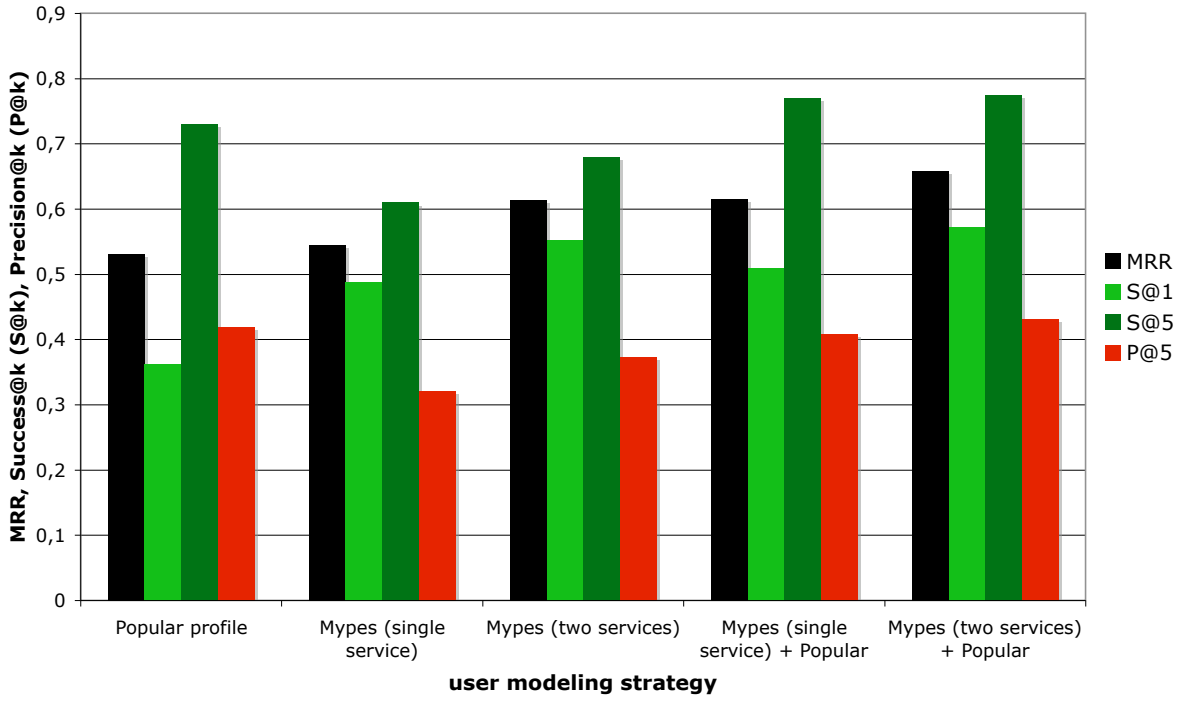
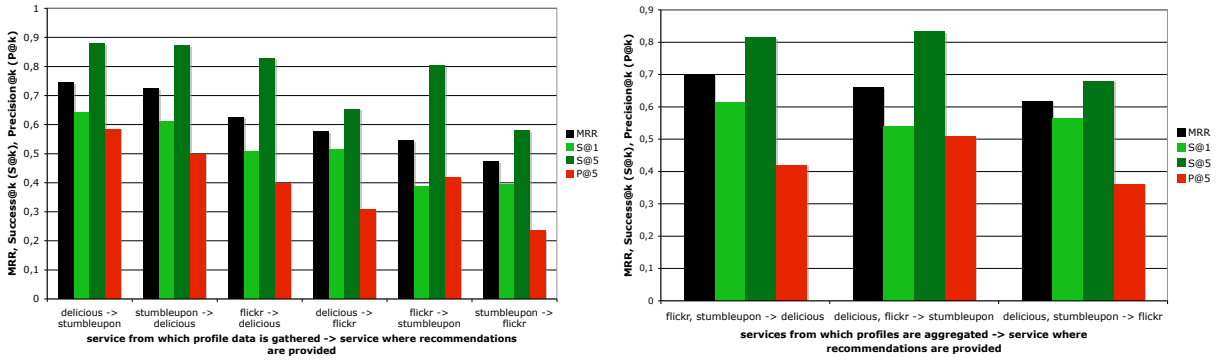


Figure 6.17: Comparison of user modeling strategies with respect to *tag recommendation* quality.

tags in Flickr by exploiting the user's Delicious profile", "recommend tags in Flickr by exploiting the user's StumbleUpon profile", etc.

Overall, the non-personalized baseline user modeling strategy that applies the most popular tags in the target system as profile (*Popular profile*) performs worst with respect to MRR (0.53). Further, the probability that a relevant tag appears at rank 1 of the tag recommendation list is just 0.36 and is therewith significantly lower than for all the other Mypes-powered user modeling strategies that utilize profile information from other sources.

It is interesting to see that the consideration of tag-based profiles coming from more than one other folksonomy system is beneficial to the recommendation quality: *Mypes (two services)*, which aggregates the user's tag-based profiles from two other services, performs—with respect top all metrics—significantly better than *Mypes (single services)*, which utilizes the user's tag-based profile of just one other service. For example, when recommending Delicious tags we achieve higher accuracy if we merge the user's StumbleUpon and Flickr profile instead of just using her StumbleUpon profile. As the size of the tag-based profiles is restricted to 150 tag-weight pairs for all strategies, this improvement cannot be explained by some increase of number of tags, for which we know that they have been applied by the user, but rather it seems that by aggregating multiple tag-based profiles originating from different folksonomy systems we can more precisely identify these tags that are essentially of interest to the user.



(a) Mypes (single service) (b) Mypes (two services)

Figure 6.18: Performance of Mypes-based tag recommendations for different settings where the Mypes profile originates from (a) one service or (b) two services different from the target service where recommendations are provided.

Figure 6.17 also reveals that the mixture of popular and Mypes profiles leads to further improvements regarding the recommendation performance. In particular, the mixture of *Mypes (two services)* and the *Popular profile* strategy, for which the tag-based profile $P_{Mypes,popular}(u)@150$ is constructed by combining $P_{U,Mypes}(u)@150$ (= aggregation of $P_{U,service_1}(u)$ and $P_{U,service_2}(u)$) and $P_{popular}(u)@150$ (see Profile Aggregation, Definition 6.1), is the best strategy with regards to all metrics. It performs significantly better than the baseline strategy (*Popular profile*) and improves MRR and S@1 by 24% and 58% respectively.

Given the high performance of the Mypes-based user modeling strategies, we conclude that user-specific preferences are essential for computing tag recommendations. However, in addition to user-specific characteristics it is also important to consider tagging characteristics that are specific to the individual folksonomy systems. Thus, the user modeling strategies that combine individual and folksonomy-specific characteristics achieve the best results for the tag recommendation task.

Figure 6.18 details the performances of the Mypes-based strategies for the different settings. Using the users' Delicious profiles to recommend StumbleUpon tags and vice versa achieves significantly the best performance (see Figure 6.18(a)). Correspondingly, Figure 6.18(b) shows that recommending Flickr tags based on the aggregated Delicious and StumbleUpon profiles is most difficult. We assume that this can be explained by the characteristics of the folksonomy systems: Delicious and StumbleUpon have similar purposes (*bookmarking*) in contrast to Flickr (*photo sharing*) and the individual users apply similar tags in both systems—at least least the overlap of the individual Delicious and StumbleUpon profiles is higher than the overlap of Flickr and Delicious/StumbleUpon profiles (cf. Section 6.2.4).

Delicious profiles turn out to be more valuable for computing cold-start tag recommendations than StumbleUpon profiles. This can be explained by the smaller average size

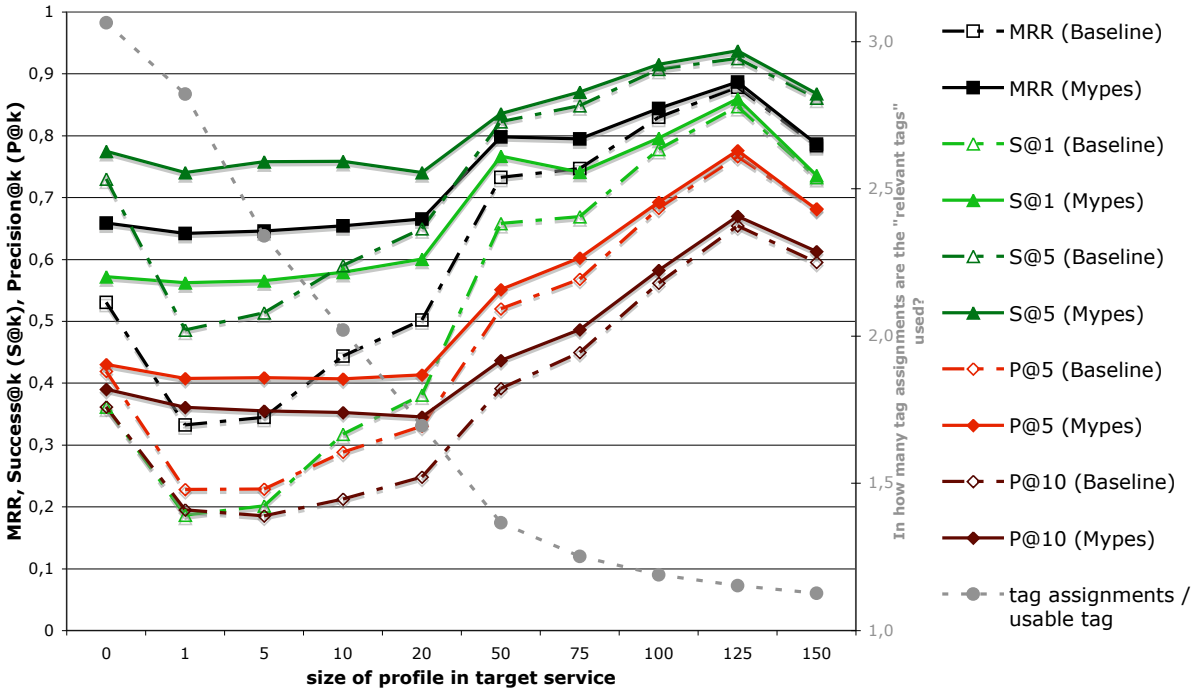


Figure 6.19: Recommending new tags when the user starts interacting in the target system. Comparison between *baseline* strategy that exploits the user profile of the target system (*Target + Popular profile*) and the Mypes strategy that also utilizes profile information from another system (*Mypes (two services) + Popular + Target profile*).

of the StumbleUpon profiles (cf. Table 6.3.2) as well as by the smaller variety of distinct tags available in the StumbleUpon folksonomy. This smaller variety might be caused by the tag suggestions provided by the tagging interface of StumbleUpon where users can simply click on suggested tags instead of entering their own tags. In Section 3.4.2 we saw that such suggestions can foster the alignment of the tagging vocabulary of a folksonomy and the results depicted in Figure 6.18 suggest that this results in less valuable user profiles.

Cold-start tag recommendations over time: growing profiles

For simulating the cold-start tag recommendations of the previous experiment we removed all tags from the user profiles. Thus we were not able to test the user modeling strategy that creates a tag-based profile from the tagging activities the user performed in the target system (*Target profile*, see Section 6.3.1). Now, we would like to analyze how the recommendation quality evolves for the different strategies when the user starts interacting with the tagging system, i.e. when the target profiles starts growing. The challenge of the recommender strategies is to compute these tags the user will apply in the future. Hence, tags that are already contained in the target profile are not considered as relevant tag recommendations as they are already known to the user.

Figure 6.19 shows how the recommendation quality evolves over time when the profile available in the target system grows (i.e., the number of entries in $P_{target}(u)$ increases). While the baseline strategy is restricted to profile information available in the folksonomy system where the recommendations are delivered (*Target + Popular profile*) , the Mypes approach can also take advantage from the user-specific profiles available in other folksonomy systems (*Mypes (two services) + Popular + Target profile*).

For both strategies we see that the performance increases over time. Hence, the more profile information is available in the target system the better the quality of the recommendations. Given our experimental setup, such behavior is not necessarily expected because the recommendation task becomes more difficult when the size of the target profile grows as the number of relevant tags—*new tags* the user has not applied yet—decreases and the relevant tags the recommenders have to identify originate rather from the *long tail* of the user profiles, i.e. these tags are rather infrequently used as indicated by the ratio of tag assignments per usable (*recommendable*) tag (see Figure 6.19). For example, when the target profile contains 150 distinct tags then the recommender algorithms have to detect these tags which are, on average, applied in 1.13 tag assignments. Hence, these tags are almost just applied once. These hard conditions might also explain the small decrease in performance when the profiles contain already 150 tags.

In general, the Mypes approach, which models users across folksonomy system boundaries, clearly performs better than the baseline approach, which does not consider external knowledge available in the Social Web. For example, given a target profile that contains already 20 entries specifying the interests in tags, the success rates are 0.6 and 0.74 regarding S@1 and S@5 metrics for the Mypes approach in contrast to 0.38 and 0.65 for the baseline approach.

The predominance of the Mypes approach is consistent over time. Mypes performs significantly (paired T-test, $alpha = 0.01$) with respect to all metrics for the different target profile sizes in the range of 0 and 75. Hence, even if the target profile contains already 75 tags, the consideration of external profile information still leads to a significant improvement of the tag recommendation quality. When the target profile size exceeds 100 tags then the performance differences are no longer significant, but Mypes still generates better results than the baseline strategy.

6.3.4 Resource Recommendation Experiment

The setup of the resource recommendation experiment is analog to the tag recommendation experiment presented in the previous section. We evaluated the user modeling strategies again by means of a *leave-many-out evaluation* [99] and removed all tag assignments Y_u performed by u in system A from the folksonomy to simulate the cold-start where u is a new user to whom we would like to recommend resources and Delicious bookmarks in particular. We applied MRR (Mean Reciprocal Rank), S@k (success at rank k) and P@k (precision at rank k) to measure the quality of the recommendations

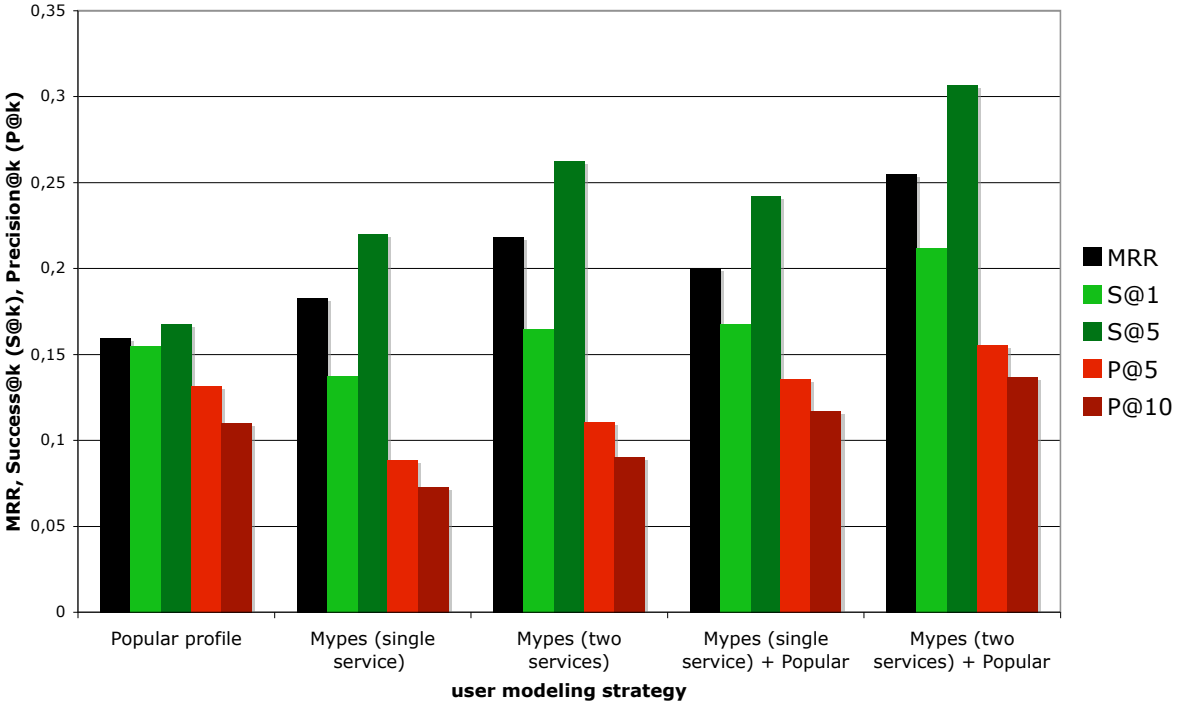


Figure 6.20: Comparison of user modeling strategies with respect to *resource recommendation* quality.

and considered these resources as relevant that were tagged by the user u, i.e. these resources that are referenced from the tag assignments Y_u that were removed before computing the recommendations. Statistical significance was tested via a two-tailed t-Test where the significance level was set to $\alpha = 0.01$.

Cold-start resource recommendations

The results of the cold-start resource recommendations are summarized in Figure 6.20 and confirm our findings revealed by the tag recommendation experiments. The Mypes strategies (*Mypes (single service)* and *Mypes (two services)*) perform significantly better than the baseline strategy (*Popular profile*) with respect to MRR and S@5. However, regarding the precisions of the recommendations (P@5 and P@10) these two strategies that consider only external profile information perform significantly worse than the baseline. In detail, we observed that the baseline user modeling strategy, which utilizes popular Delicious tags as user profile, specifically promotes "popular" resources that are shared by at least two users while the Mypes approaches (*Mypes (single service)* and *Mypes (two services)*) recommend resources independently from their popularity (cf. Figure 6.15(b)).

The mixtures of the basic Mypes approaches with the popular profile strategy are the most successful user modeling strategies. *Mypes (single service) + Popular* and *Mypes (two services) + Popular* both perform with respect to all metrics significantly better

than the baseline strategy. The absolute success rates of the resource recommendations are lower than the success rates of the tag recommendations. We identify two main reasons for this.

1. The user modeling strategies identify preferences regarding tags. For the tag recommendation task (see Problem 5) these preferences can directly be exploited to deduce tags which should be recommended to the user: in the tripartite folksonomy graph $\mathbb{G}_{\mathbb{F}}$ (cf. Section 2.2.2) these nodes that should be recommended to the user correspond to the nodes for which the user modeling strategies inferred specific preferences (e.g., $u \leftrightarrow t_{preference,recommendation}$). For the resource recommendation task (see Problem 6), on the contrary, the strategies have to infer the recommendations via the tags (cf. [197]): the nodes for which the user modeling strategies deduced preferences do not correspond to the type of nodes that should be recommended to these user (e.g., $u \leftrightarrow t_{preference} \leftrightarrow r_{recommendation}$).
2. The fraction of relevant items is much lower for the resource recommendation task than for the tag recommendation task. For example, when computing cold-start tag recommendations in Delicious there are, on average, 192.67 relevant of the overall 21239 tags relevant, i.e. given a strategy that would simply guess a tag to be recommended to the user would achieve 0.0091 regarding S@1. By contrast, there are, on average, just 82.55 of the overall 25365 Delicious resources relevant which would result in $S@1 = 0.0039$.

Considering these challenges, the performance of the resource recommendation strategies is very encouraging. The best strategy (*Mypes (two services) + Popular*), which considers profile information from external folksonomy systems, achieves a precision within the top ten recommendations (P@10) of 13.7%, i.e. if the Mypes recommender suggests 10 out of more than 25000 resources to a new user, for whom there is no profile information available in Delicious, then at least 1.37 resources of these recommendations would, on average, be bookmarked by the user. The actual quality of the resource recommendations might even be higher as we do not know how much the users appreciate those resources they have not bookmarked.

6.3.5 Synopsis

In summary, our experiments show that user modeling across folksonomy system boundaries is beneficial for tag and resource recommendations. In particular, this holds for cold-start recommendations, for which no or little user profile information is available in the folksonomy system. Regarding the tag recommendation task, we further measured the recommendation quality over time and revealed that even when there is considerably much user-specific profile data available in the target system (e.g., if the *target profile* contains 75 entires), Mypes-based user modeling still improves the recommendation quality significantly (paired t-test, significance level $\alpha = 0.01$).

6.4 Discussion

In this chapter we introduced strategies for modeling users across Social Web system boundaries. These strategies model the users *in context* of their Social Web activities. Instead of constructing user profiles based on a single source of information, i.e. the data available in a given system, our strategies also exploit the user profile traces distributed on the Social Web.

We analyzed the nature of these user profile traces and discovered that aggregating the individual profiles is beneficial to user modeling and personalization. For both explicitly provided profile information (e.g. name, location, etc.) as well as rather implicitly provided tag-based profiles, the aggregated profiles reveal significantly more facets about the individual users.

We implemented our user modeling approach into the so-called Mypes service that supports linkage, aggregation, alignment and semantic enrichment of user profiles available in Social Web systems such as Flickr, Delicious, or Facebook. Further, we applied Mypes to investigate the impact of cross-system user modeling in the Social Web on recommender systems and found out that aggregated profiles improve tag and resource recommendation performance significantly. In summary, we can thus answer the research questions raised at the beginning of this chapter as follows.

User Modeling approach. Our general approach to distributed user modeling is based on profile aggregation. We model tag-based user profiles by means of weighted tags; an aggregated profile is an accumulation of given profiles. With Mypes we introduce a service that supports profile aggregation from Social Web systems such as Flickr, Delicious, Last.fm and StumbleUpon as well as social networking services such as LinkedIn or Facebook. It enables developers to immediately take advantage from our cross-system user modeling approach and enables end-users to inspect their distributed profiles so that they become aware of the information available about them on the Social Web.

General benefits. Aggregated profiles reveal significantly more information about the users than the profiles available in the individual services. Our analysis shows that aggregated profiles can be applied to complete missing profile facets in particular services or to generate more complete FOAF and vCard profiles. Further, although the individual tag-based profiles overlap just little between the different services, we proved that profile aggregation helps to solve cold-start problems: given a user u, who is new to a service A, as well as profile information aggregated and enriched by Mypes, one can predict certain facets of u's tag-based profile in service A with precision and recall higher than 50%.

Impact on Recommender Systems. Our recommendation experiments suggest that the consideration of external profile information has significant impact on the quality of tag and resource recommendations. Using Mypes profiles as input for our

FolkRank-based recommender algorithm, we achieved significantly better results and outperformed these strategies that ignored user profile information from external sources.

Cross-system user modeling in the Social Web thus allows individual services to enrich the profiles of their users and can be applied to improve personalization functionality significantly. The proposed user modeling strategies have been implemented in the Mypes framework and are available online to support researchers, application developers and end-users in harnessing user data distributed on the Social Web.

7 Summary

The Social Web is not a new Web, but rather a paradigm that describes the culture of social participation on the Web. Today, the power of the Web heavily depends on the power of the people who publish and share resources on the Web. People write articles for the Wikipedia encyclopedia, share their thoughts and express their opinions in blogs, communicate with other people via social networking sites like Facebook, and upload their pictures, bookmarks, or videos to resource sharing systems like Flickr, Delicious, and YouTube respectively. Given this culture of participation, the number of Web resources is growing massively and information retrieval becomes a non-trivial problem. To tackle this problem and support users in organizing and retrieving Web resources, many online systems feature social tagging and exploit folksonomy structures that emerge from social tagging. In this thesis we proposed novel context and user modeling approaches for Social Web systems and introduced algorithms that improve search and recommender functionality in these systems. This chapter summarizes the main contributions and outlines possible future work.

7.1 Summary of Contributions

We summarize the main contributions of this thesis by answering the core research questions, which we identified based on related research in Section 2.3.

Context Modeling in Folksonomy Systems. In related work [124, 169, 214], social tagging structures are modeled without referring to contextual information describing the semantics of tagging activities in more detail: a traditional tag assignment – the core structure of a folksonomy – specifies which user assigned which tag to which resource. The semantics of these tag assignments are not well defined. To better capture the semantics of folksonomies, meaningful contextual information is required and the following research questions had to be answered.

- How can contextual information be modeled in folksonomies?
- How can folksonomy systems deduce semantically meaningful contextual information from tagging activities?

We answered these questions in Chapter 3 and proposed a generic context folksonomy model that allows to attach arbitrary contextual information to tag assignments. We further implemented this model and developed two systems that

allow for different kind of context information: (1) *GroupMe!* is a social bookmarking system that enables users to group their bookmarks; tagging activities are performed in context of a group of related bookmarks which allows to better capture the semantics of tag assignments. (2) *TagMe!* is a social tagging and exploration service for Flickr pictures which enriches tag assignments with URIs that define the semantic meaning of a tag and enables users to attach metadata to their tag assignments. Both applications demonstrate how social tagging systems can benefit from the context folksonomy model and infer valuable semantics from tagging activities.

Usage data analyses further showed that users appreciate the novel tagging features provided by these systems and depicted several benefits of the corresponding context folksonomies. For example, context information embedded in the GroupMe! folksonomy supports the retrieval of untagged multimedia resources and tagging context in TagMe! allows for the deduction of semantic relationships between tags. Moreover, both systems enable also other applications to take advantage from the semantically enhanced folksonomies. For example, in GroupMe! folksonomy data is published according to the principles of Linked Data and third-party applications are already connected to GroupMe! to exploit the additional semantics for improving organization and sharing of learning resources or improving image search. The Semantic Web community valued GroupMe! as part of the Semantic Web Challenge 2007 as it illustrates the interconnection of Social Web and Semantic Web design principles.

Search and Ranking in Folksonomy Systems. Ranking algorithms that support information retrieval in folksonomy systems have been proposed by Hotho et al. [124] or Bao et al. [51]. However, these algorithms do not consider contextual information of social tagging activities. We thus answered open research questions regarding search and ranking in the Social Web and folksonomy systems in particular.

- How to design ranking algorithms that exploit context information available in folksonomies?
- How does the exploitation of context information available in folksonomies impact information retrieval performance?

By introducing new algorithms like *GRank* and *SocialHITS* and enhancing existing algorithms like FolkRank or SocialPageRank, we gave answers to the first question in Chapter 4. In summary, these algorithms exploit the context folksonomy model proposed in Chapter 3 by representing the folksonomy as a graph, in which nodes are connected in a more meaningful way than in folksonomy graphs constructed based on traditional folksonomy models.

In Chapter 4 we further applied these algorithms for search in folksonomies and proved that algorithms like *GRank* or *GFolkRank*, which make use of contextual information available in folksonomies, significantly improve search performance in

comparison to algorithms, which do not harness additional context information. Further, we showed how our algorithms can further be optimized and how they can be applied as search strategies in a re-ranking scenario so that also ranking algorithms, which are not context-aware by themselves, can benefit from the context folksonomy model and gain better search performance.

In summary, we thus showed in various experiments the benefits of our context folksonomy model and demonstrated that the exploitation of context information with the suite of context-aware ranking algorithms leads to a significant improvement of the information retrieval performance in Social Web systems.

User Modeling and Personalization in Social Web Systems. User modeling and personalization has not been studied extensively in the context of Social Web and folksonomy systems yet. Hence, this thesis is an important contribution to better understand the design of adaptive applications on the Social Web. Research on user modeling on the Web suggests to model both user *and* context to better support personalization in adaptive systems [126]. In this thesis we investigated user and context modeling strategies for personalization in Social Web systems.

- How can user and context modeling strategies support personalization in Social Web systems?
- Which type of user and context modeling strategy is the most appropriate for recommender systems and personalized search?

In Chapter 5 we developed a framework of user modeling and personalization strategies for social tagging systems to answer these questions. We introduced strategies for modeling information about the user and the user's actual context by means of tag-based profiles. We then proposed methods that allow for using such user and context models in combination with ranking algorithms outlined in Chapter 4 to make tagging systems adaptive. We applied our user modeling and personalization framework to provide personalized search and tag recommendations in folksonomy systems and evaluated the impact of the user and context modeling strategies on these personalization tasks.

For *personalized search* we saw that lightweight context modeling strategies, which utilize a tag-based representation of the user's browsing history, are more appropriate than heavy user modeling strategies, which exploit the complete user profile. This observation was significant and consistent over the different search experiments. Correspondingly, these context modeling strategies performed also best in the *tag recommendation* experiments. We further confirmed the results from Chapter 4 and showed that the context-sensitive ranking algorithms lead to significant improvements over baseline strategies such as FolkRank. In particular, SocialHITS was successfully applied to search and rank people in tagging systems while overall GRank was the most successful algorithm.

Cross-system User Modeling in the Social Web. With the design of Web systems like GroupMe! and TagMe! we enable interoperability between Social Web systems and support a paradigm shift towards the Social Semantic Web where social data can easily be shared across system boundaries. Cross-system user modeling in the Social Web and its impact on personalization has not been researched in detail yet. We thus answered the following research questions.

- How to model users across system boundaries in the Social Web?
- What are the benefits of cross-system user modeling and how does it impact the performance of recommender systems in the Social Web?

With profile aggregation strategies proposed in Chapter 6 and the corresponding *Mypes* service that we developed to support linkage, aggregation, alignment and semantic enrichment of user profiles distributed on the Social Web, we answered the first question.

By analyzing a large dataset of distributed user profile traces on the Social Web, we identified significant benefits of cross-system user modeling. The core advantage is that aggregated profiles reveal significantly more profile facets about the individual users than system-specific profiles and therefore allow for various applications, such as synchronizing and completing system-specific profiles.

Further, our proposed cross-system user modeling methods improved both tag and resource recommendation performance significantly. In particular, we analyzed cold-start situations, in which recommendations should be computed for new users. We also measured recommendation quality over time and revealed that even when user-specific profile data available in the target system is growing, the consideration of external profile information – aggregated and semantically enhanced with Mypes – still improves the recommendation quality significantly.

In summary, this thesis contributes to research on information retrieval as well as user modeling and personalization on the Social Web. We introduced a first contextualization model for social tagging, developed ranking algorithms that exploit this model to improve search in tagging systems. We further developed a context-based user modeling and personalization framework that is proven to make tagging systems adaptive and moreover allows for advanced user modeling and personalization across system boundaries in the Social Web.

7.2 Outlook

The framework of user and context models, ranking and personalization algorithms as well as the corresponding implementations such as Mypes or GroupMe! open new interesting research paths that are worth to be explored in the future. Specifically in the areas of user modeling and personalization there are some exciting problems and ideas we would like to outline in this section.

First, the *temporal dynamics of user profiles in folksonomy systems* can further be studied. In this thesis we showed, for example, that the actual user context forms a more valuable profile than a rather long term user history for applications such as personalized search and tag recommendations. Recently, Koren [145] showed that a fine-grained distinction between transient and long term profile patterns can lead to significant improvement of the traditional collaborative filtering approach to recommender systems. To explore these findings within the scope of folksonomy systems in more detail and also in other application contexts than social resource sharing systems like GroupMe!, we developed Radiotube.tv[1] [98], which connects Last.fm and YouTube to provide personalized music video recommendations and enables researchers to plug-in and evaluate user modeling and recommender strategies.

Second, with support of our methods for enriching folksonomies and user profiles with additional semantics, *knowledge extraction from tag-based profiles* becomes a feasible research topic. In line with Rattenbury et al. [182], who investigated how events and places can be deduced from the Flickr folksonomy, an analysis on how knowledge can be extracted from individual user profiles would be valuable. Our studies in Section 6.2 showed that there is a correlation between tag-based profiles, which emerge from the users' tagging activities, and social networking profiles, which are explicitly filled by users. However, further research is required to learn the semantics of these correlations so that tagging activities of a user can be transformed in some sort of structured knowledge base.

Third, in the field of *cross-system user modeling and personalization* on the Social Web and across tagging systems particularly further applications can be researched. Mehta et al. proposed cross-system personalization approaches, which aim to make recommender systems more robust against spam and cold-start problems. However, they could not evaluate their methods on Social Web data, where individual user interactions are performed across different systems and domains. Experiments have instead been conducted on user data, which originates from one system and was split to simulate different systems [164, 163]. With the cross-system user modeling framework Mypes we developed a tool that allows researchers to explore cross-system user modeling on real user data distributed on the Social Web. Further, it enables developers to immediately benefit from cross-system user modeling approaches proposed in this thesis. While our evaluation revealed significant benefits of cross-system user modeling for recommender systems in the scope of social bookmarking and photo sharing, there are more types of correlations that can be studied to further explain the interdependency between user interactions performed in different systems and domains.

Finally, the user and context modeling approaches we proposed, the corresponding algorithms and tools we developed as well as the datasets we made available will hopefully stimulate the research community to further advance the Social Web.

[1] `http://radiotube.tv`

Bibliography

[1] F. Abel. The Benefit of additional Semantics in Folksonomy Systems. In *Proceedings of the 2nd PhD workshop on Information and Knowledge Management (PIKM '08)*, pages 49–56, New York, NY, USA, 2008. ACM.

[2] F. Abel, S. Araujo, N. Henze, E. Herder, G.-J. Houben, and E. Leonardi, editors. *Proceedings of the International Workshop on User Data Interoperability in the Social Web (UDISW 2010)*, Hong Kong, China, February 2010. ACM.

[3] F. Abel, M. Baldoni, C. Baroglio, N. Henze, R. Kawase, D. Krause, and V. Patti. Leveraging search and content exploration by exploiting context in folksonomy systems. In D. Cunliffe and D. Tudhope, editors, *New Review of Hypermedia and Multimedia: Web Science*, volume 16, pages 33–70. Taylor & Francis, April 2010.

[4] F. Abel, M. Baldoni, C. Baroglio, N. Henze, D. Krause, and V. Patti. Context-based ranking in folksonomies. In C. Cattuto, G. Ruffo, and F. Menczer, editors, *HYPERTEXT 2009, Proceedings of the 20th ACM Conference on Hypertext and Hypermedia, Torino, Italy, June 29 - July 1, 2009*, pages 209–218, New York, NY, USA, 2009. ACM.

[5] F. Abel, R. Baumgartner, A. Brooks, C. Enzi, G. Gottlob, N. Henze, M. Herzog, M. Kriesell, W. Nejdl, and K. Tomaschewski. The Personal Publication Reader. In Y. Gil, E. Motta, V. R. Benjamins, and M. A. Musen, editors, *International Semantic Web Conference*, volume 3729 of *Lecture Notes in Computer Science*, pages 1050–1053. Springer, 2005.

[6] F. Abel, I. I. Bittencourt, E. de Barros Costa, N. Henze, D. Krause, and J. Vassileva. Recommendations in online discussion forums for e-learning systems. *IEEE Transactions on Learning Technologies (TLT)*, 3(2):165–176, 2010.

[7] F. Abel, I. I. Bittencourt, N. Henze, D. Krause, and J. Vassileva. A rule-based recommender system for online discussion forums. In W. Nejdl, J. Kay, P. Pu, and E. Herder, editors, *Proceedings of the 5th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH '08), Hannover, Germany*, volume 5149 of *Lecture Notes in Computer Science*, pages 12–21. Springer, 2008.

[8] F. Abel, J. L. D. Coi, N. Henze, A. W. Koesling, D. Krause, and D. Olmedilla. Enabling advanced and context-dependent access control in rdf stores. In Aberer et al. [39], pages 1–14.

[9] F. Abel, J. L. D. Coi, N. Henze, A. W. Koesling, D. Krause, and D. Olmedilla. A user interface to define and adjust policies for dynamic user models. In J. Filipe and J. Cordeiro, editors, *Proceedings of the Fifth International Conference on Web Information Systems and Technologies (WEBIST '09), Lisbon, Portugal, March 23-26, 2009*, pages 184–191. INSTICC Press, 2009.

[10] F. Abel, M. Frank, N. Henze, D. Krause, D. Plappert, and P. Siehndel. GroupMe! – Where Semantic Web meets Web 2.0. In A. et al., editor, *The Semantic Web, 6th International Semantic Web Conference (ISWC), 2nd Asian Semantic Web Conference (ASWC)*, Berlin, Heidelberg, November 2007. Springer Verlag.

[11] F. Abel, M. Frank, N. Henze, D. Krause, D. Plappert, and P. Siehndel. GroupMe! - Where Semantic Web meets Web 2.0. In J. Golbeck and P. Mika, editors, *Semantic Web Challenge*, volume 295 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2007.

[12] F. Abel, M. Frank, N. Henze, D. Krause, and P. Siehndel. GroupMe! - Combining ideas of Wikis, Social Bookmarking, and Blogging. In E. Adar, M. Hurst, T. Finin, N. Glance, N. Nicolov, and B. Tseng, editors, *2nd International Conference on Weblogs and Social Media (ICWSM 2008), Seattle, USA.*, pages 230–231, Menlo Park, California, April 2008. The AAAI Press.

[13] F. Abel, D. Heckmann, E. Herder, J. Hidders, G.-J. Houben, D. Krause, E. Leonardi, and K. van der Slujis. A framework for flexible user profile mashups. In A. Dattolo, C. Tasso, R. Farzan, S. Kleanthous, D. B. Vallejo, and J. Vassileva, editors, *Int. Workshop on Adaptation and Personalization for Web 2.0 at UMAP '09*, pages 1–10. CEUR Workshop Proceedings, 2009.

[14] F. Abel, D. Heckmann, E. Herder, J. Hidders, G.-J. Houben, D. Krause, E. Leonardi, and K. van der Slujis. Mashing up user data in the grapple user modeling framework. In D. Hauger, M. Köck, and A. Nauerz, editors, *Workshop on Adaptivity and User Modeling in Interactive Systems (ABIS '09)*, pages 1–2, 2009.

[15] F. Abel, D. Heckmann, E. Herder, J. Hidders, G.-J. Houben, E. Leonardi, and K. van der Sluijs. Definition of an appropriate User Profile format. Technical report, Grapple Project, EU FP7, Reference 215434, 2009.

[16] F. Abel, N. Henze, E. Herder, G.-J. Houben, D. Krause, and E. Leonardi. Building Blocks for User Modeling with data from the Social Web. In *International Workshop on Architectures and Building Blocks of Web-Based User-Adaptive Systems (WABBWUAS) co-located with International Conference on User Modeling, Adaptation and Personalization (UMAP), Hawaii, USA*, June 2010.

[17] F. Abel, N. Henze, E. Herder, and D. Krause. Interweaving public user profiles on the web. In P. D. Bra, A. Kobsa, and D. N. Chin, editors, *User Modeling, Adaptation, and Personalization, 18th International Conference, UMAP 2010, Big Island, HI, USA, June 20-24, 2010. Proceedings*, volume 6075 of *Lecture Notes in*

Computer Science, pages 16–27. Springer, 2010.

[18] F. Abel, N. Henze, E. Herder, and D. Krause. Linkage, Aggregation, Alignment and Enrichment of Public User Profiles with Mypes. In A. Blumauer, R. Cyganiak, N. Henze, A. Paschke, and T. Pellegrini, editors, *International Conference on Semantic Systems (I-Semantics), Graz, Austria*. ACM, September 2010.

[19] F. Abel, N. Henze, R. Kawase, and D. Krause. The impact of multifaceted tagging on learning tag relations and search. In *Extended Semantic Web Conference, Heraklion, Greece*. Springer, May 2010.

[20] F. Abel, N. Henze, and D. Krause. Exploiting additional Context for Graph-based Tag Recommendations in Folksonomy Systems. In *Int. Conf. on Web Intelligence and Intelligent Agent Technology (WI/IAT '08)*. ACM Press, December 2008.

[21] F. Abel, N. Henze, and D. Krause. GroupMe! - Where Information meets. In J. Huai, R. Chen, H.-W. Hon, Y. Liu, W.-Y. Ma, A. Tomkins, and X. Zhang, editors, *Proceedings of the 17th International Conference on World Wide Web (WWW '08), Beijing, China*, pages 1147–1148. ACM, 2008.

[22] F. Abel, N. Henze, and D. Krause. A Novel Approach to Social Tagging: GroupMe! In *4th International Conference on Web Information Systems and Technologies (WEBIST '08)*, May 2008.

[23] F. Abel, N. Henze, and D. Krause. Ranking in Folksonomy Systems: can context help? In J. G. Shanahan, S. Amer-Yahia, I. Manolescu, Y. Zhang, D. A. Evans, A. Kolcz, K.-S. Choi, and A. Chowdhury, editors, *Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM '08), Napa Valley, California, USA*, pages 1429–1430. ACM, 2008.

[24] F. Abel, N. Henze, and D. Krause. Context-aware ranking algorithms in folksonomies. In J. Filipe and J. Cordeiro, editors, *Proceedings of the Fifth International Conference on Web Information Systems and Technologies (WEBIST '09), Lisbon, Portugal*, pages 167–174. INSTICC Press, 2009.

[25] F. Abel, N. Henze, and D. Krause. Social semantic web at work: Annotating and grouping social media content. In S. H. Jose Cordeiro and J. Filipe, editors, *Web Information Systems and Technologies 4th International Conference, WEBIST 2008, Funchal, Madeira, Portugal, May 4-7, 2008, Revised Selected Papers, Lecture Notes in Business Information Processing*, pages 199–213. Springer, April 2009.

[26] F. Abel, N. Henze, and D. Krause. Optimizing search and ranking in folksonomy systems by exploiting context information. *Lecture Notes in Business Information Processing*, 45(2):113–127, 2010.

[27] F. Abel, N. Henze, D. Krause, and M. Kriesell. On the effect of group structures on ranking strategies in folksonomies. In R. Baeza-Yates and I. King, editors,

Weaving Services and People on the World Wide Web, pages 275–300. Springer, July 2009.

[28] F. Abel, N. Henze, D. Krause, and M. Kriesell. Semantic enhancement of social tagging systems. In V. Devedzic and D. Gasevic, editors, *Annals of Information Systems – Web 2.0 & Semantic Web*, volume 6, pages 25–54, April 2009.

[29] F. Abel, N. Henze, D. Krause, and D. Plappert. User Modeling and User Profile Exchange for Semantic Web applications. In J. Baumeister and M. Atzmüller, editors, *LWA*, volume 448 of *Technical Report*, pages 4–9. Department of Computer Science, University of Würzburg, Germany, 2008.

[30] F. Abel, E. Herder, G.-J. Houben, and E. Leonardi, editors. *Proceedings of the International Workshop on Linking of User Profiles and Applications in the Social Semantic Web (LUPAS 2010)*, volume 595, Heraklion, Greece, May 2010. CEUR-WS.org.

[31] F. Abel, E. Herder, G.-J. Houben, M. Pechenizkiy, and M. Yudelson, editors. *Proceedings of the International Workshop on Architectures and Building Blocks of Web-Based User-Adaptive Systems (WABBWUAS 2010)*, volume 609, Hawaii, USA, June 2010. CEUR-WS.org.

[32] F. Abel, E. Herder, P. Kärger, D. Olmedilla, and W. Siberski. Exploiting preference queries for searching learning resources. In E. Duval, R. Klamma, and M. Wolpers, editors, *Creating New Learning Experiences on a Global Scale, Second European Conference on Technology Enhanced Learning (EC-TEL '07), Crete, Greece, September 17-20, 2007, Proceedings*, volume 4753 of *Lecture Notes in Computer Science*, pages 143–157. Springer, 2007.

[33] F. Abel, E. Herder, I. Marenzi, W. Nejdl, and S. Zerr. Evaluating the Benefits of Social Annotation for Collaborative Search. In E. Agichtein, M. Hearst, and I. Soboroff, editors, *2nd Annual Workshop on Search in Social Media (SSM '09), co-located with ACM SIGIR 2009 Conference on Information Retrieval, Boston, USA*, July 2009.

[34] F. Abel, R. Kawase, and D. Krause. Leveraging multi-faceted tagging to improve search in folksonomy systems. In M. Chignell and E. Toms, editors, *Proceedings of the 21st ACM Conference on Hypertext and Hypermedia (Poster Track)*. ACM, June 2010.

[35] F. Abel, R. Kawase, D. Krause, and P. Siehndel. Multi-faceted Tagging in TagMe! In *8th International Semantic Web Conference (ISWC2009)*, October 2009.

[36] F. Abel, R. Kawase, D. Krause, P. Siehndel, and N. Ullmann. The Art of multi-faceted Tagging – Interweaving spatial annotations, categories, meaningful URIs and tags. In *6th International Conference on Web Information Systems and Technologies (WEBIST 2010)*, April 2010.

[37] F. Abel, I. Marenzi, W. Nejdl, and S. Zerr. LearnWeb2.0: Resource Sharing in Social Media. In *Workshop on Social Information Retrieval for Technology-Enhanced Learning (SIRTEL'09) at the International Conference on Web-based Learning (ICWL '09), Aachen, Germany*, August 2009.

[38] F. Abel, I. Marenzi, W. Nejdl, and S. Zerr. Sharing Distributed Resources in LearnWeb2.0. In U. Cress, V. Dimitrova, and M. Specht, editors, *Learning in the Synergy of Multiple Disciplines, 4th European Conference on Technology Enhanced Learning (EC-TEL 2009), Nice, France*, volume 5794 of *Lecture Notes in Computer Science*, pages 154–159. Springer, 2009.

[39] K. Aberer, K.-S. Choi, N. F. Noy, D. Allemang, K.-I. Lee, L. J. B. Nixon, J. Golbeck, P. Mika, D. Maynard, R. Mizoguchi, G. Schreiber, and P. Cudré-Mauroux, editors. *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007*, volume 4825 of *Lecture Notes in Computer Science*. Springer, 2007.

[40] B. Adrian, L. Sauermann, and T. Roth-Berghofer. ConTag: A semantic tag recommendation system. In *Proceedings of International Conference on Semantic Systems (I-Semantics '07)*, pages 297–304, 2007.

[41] R. Agrawal, T. Imieliński, and A. Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD international conference on Management of data (SIGMOD '93)*, pages 207–216, New York, NY, USA, 1993. ACM.

[42] M. Ames and M. Naaman. Why we tag: motivations for annotation in mobile and online media. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '07)*, pages 971–980, New York, NY, USA, 2007. ACM.

[43] S. Anand and B. Mobasher. Intelligent techniques for web personalization. *Intelligent Techniques for Web Personalization, IJCAI 2003 Workshop, ITWP 2003, Acapulco, Mexico, August 11, 2003, Revised Selected Papers*, pages 1–36, 2005.

[44] A. Ankolekar, M. Krötzsch, T. Tran, and D. Vrandecic. The two cultures: mashing up Web 2.0 and the Semantic Web. In *Proceedings of the 16th international conference on World Wide Web (WWW '07)*, pages 825–834, New York, NY, USA, 2007. ACM.

[45] L. Aroyo, P. Dolog, G. Houben, M. Kravcik, A. Naeve, M. Nilsson, and F. Wild. Interoperability in pesonalized adaptive learning. *Journal of Educational Technology & Society*, 9 (2):4–18, 2006.

[46] M. Assad, D. Carmichael, J. Kay, and B. Kummerfeld. PersonisAD: Distributed, Active, Scrutable Model Framework for Context-Aware Services. *Pervasive Computing*, pages 55–72, 2007.

[47] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. DBpedia: A Nucleus for a Web of Open Data. In K. Aberer, K.-S. Choi, N. F. Noy, D. Allemang, K.-I. Lee, L. J. B. Nixon, J. Golbeck, P. Mika, D. Maynard, R. Mizoguchi, G. Schreiber, and P. Cudré-Mauroux, editors, *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11-15, 2007*, volume 4825 of *Lecture Notes in Computer Science*, pages 715–728. Springer, November 2007.

[48] R. Baeza-Yates and E. Davis. Web page ranking using link attributes. In *Proceedings of the 13th international World Wide Web conference (WWW '04)*, pages 328–329, New York, NY, USA, 2004. ACM.

[49] R. A. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.

[50] L. Balby Marinho, K. Buza, and L. Schmidt-Thieme. Folksonomy-based collabulary learning. In A. P. Sheth, S. Staab, M. Dean, M. Paolucci, D. Maynard, T. W. Finin, and K. Thirunarayan, editors, *The Semantic Web - ISWC 2008, 7th International Semantic Web Conference, ISWC 2008, Karlsruhe, Germany, October 26-30, 2008. Proceedings*, volume 5318 of *Lecture Notes in Computer Science*, pages 261–276, Berlin, Heidelberg, 2008. Springer-Verlag.

[51] S. Bao, G. Xue, X. Wu, Y. Yu, B. Fei, and Z. Su. Optimizing Web Search using Social Annotations. In *Proc. of 16th Int. World Wide Web Conference (WWW '07)*, pages 501–510. ACM Press, 2007.

[52] D. Beckett. RDF/XML syntax specification (Revised). Technical report, World Wide Web Consortium (W3C), February 2004.

[53] C. Berge. *Graphs and Hypergraphs*. Elsevier Science Ltd, October 1985.

[54] T. Berners-Lee. Information Management: A Proposal. Technical report, CERN, Geneva, Switzerland, March 1989.

[55] T. Berners-Lee. WWW at 15 years: looking forward. In A. Ellis and T. Hagino, editors, *Proceedings of the 14th international conference on World Wide Web (WWW '05), Chiba, Japan*, page 1. ACM, May 2005.

[56] T. Berners-Lee. Linked Data - design issues. Technical report, W3C, May 2007. http://www.w3.org/DesignIssues/LinkedData.html.

[57] T. Berners-Lee and D. Connolly. Hypertext markup language 2.0. Request for Comments (RFC 1866), Internet Engineering Task Force (IETF), Network Working Group, November 1995.

[58] T. Berners-Lee and D. Connolly. Notation3 (N3): A readable RDF syntax. Technical report, World Wide Web Consortium (W3C), January 2008.

[59] T. Berners-Lee, R. Fielding, and L. Masinter. Uniform Resource Identifier (URI):

Generic Syntax. Request for Comments (RFC 3986), Internet Engineering Task Force (IETF), Network Working Group, January 2005.

[60] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, 284(5):34–43, 2001.

[61] K. Bischoff, C. Firan, R. Paiu, and W. Nejdl. Can All Tags Be Used for Search? In J. G. Shanahan, S. Amer-Yahia, I. Manolescu, Y. Zhang, D. A. Evans, A. Kolcz, K.-S. Choi, and A. Chowdhury, editors, *Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM '08), Napa Valley, California, USA*, pages 1429–1430. ACM, 2008.

[62] C. Bizer, T. Heath, and T. Berners-Lee. Linked data - the story so far. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 5(3):1–22, 2009.

[63] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. DBpedia - A crystallization point for the Web of Data. *Web Semantics: Science, Services and Agents on the World Wide Web*, July 2009.

[64] U. Bojars and J. G. Breslin. SIOC Core Ontology Specification. Namespace document, DERI, NUI Galway, http://rdfs.org/sioc/spec/, January 2009.

[65] D. Brickley and R. Guha. RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation, World Wide Web Consortium (W3C), February 2004.

[66] D. Brickley and A. Miles. SKOS Core Vocabulary Specification. W3C working draft, W3C, June 2008.

[67] D. Brickley and L. Miller. FOAF Vocabulary Specification 0.91. Namespace document, FOAF Project, http://xmlns.com/foaf/0.1/, November 2007.

[68] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30:107–117, April 1998.

[69] A. Z. Broder, R. Lempel, F. Maghoul, and J. Pedersen. Efficient pagerank approximation via graph aggregation. *Journal of Information Retrieval*, 9:123–138, March 2006.

[70] J. Broekstra, A. Kampman, and F. van Harmelen. Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. In I. Horrocks and J. Hendler, editors, *Proceedings of the First Internation Semantic Web Conference (ISWC '02), Sardinia, Italy*, volume 2342 of *Lecture Notes in Computer Science*, pages 54–68. Springer, June 2002.

[71] C. H. Brooks and N. Montanez. Improved annotation of the blogosphere via autotagging and hierarchical clustering. In *Proceedings of the 15th international conference on World Wide Web (WWW '06)*, pages 625–632, New York, NY, USA,

2006. ACM.

[72] V. Bush. As we may think. *The Atlantic Monthly*, 176(1):101–108, 1945.

[73] A. Byde, H. Wan, and S. Cayzer. Personalized tag recommendations via tagging and content-based similarity metrics. In *Proc. of the Int. Conf. on Weblogs and Social Media (ICWSM)*, March 2007.

[74] Y. Cai and Q. Li. Personalized search by tag-based user profile and resource profile in collaborative tagging systems. In *Proceedings of the 19th ACM international conference on Information and knowledge management (CIKM '10)*, pages 969–978, New York, NY, USA, 2010. ACM.

[75] R. Cailliau. A Little History of the World Wide Web. Technical report, World Wide Web Consortium (W3C), 1995.

[76] F. Carmagnola and F. Cena. User identification for cross-system personalisation. *Information Sciences: an International Journal*, 179(1-2):16–32, 2009.

[77] C. Cattuto, A. Baldassarri, V. D. P. Servedio, and V. Loreto. Vocabulary growth in collaborative tagging systems. *CoRR*, abs/0704.3316, 2007.

[78] T. Çelik and K. Marks. rel="tag". Draft specification, Microformats.org, January 2005. http://microformats.org/wiki/rel-tag.

[79] T. Çelik and B. Suda. hCard 1.0. Specification, Microformats.org, April 2010. http://microformats.org/wiki/hcard.

[80] M. Cha, A. Mislove, and K. P. Gummadi. A Measurement-driven Analysis of Information Propagation in the Flickr Social Network. In *Proceedings of the 18th International World Wide Web Conference (WWW '09)*, Madrid, Spain, April 2009.

[81] J. Chen, W. Geyer, C. Dugan, M. Muller, and I. 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, 2009. ACM.

[82] P.-A. Chirita, J. Diederich, and W. Nejdl. MailRank: Using Ranking for Spam Detection. In *Proceedings of the 14th ACM international conference on Information and knowledge management (CIKM '05)*, pages 373–380, New York, NY, USA, 2005. ACM.

[83] P.-A. Chirita, C. Firan, and W. Nejdl. Personalized Query Expansion for the Web. In GAGA, editor, *Proceedings of the 30th Int. ACM SIGIR conference on Research and Development in Information Retrieval (SIGIR '07)*, pages 7–14, New York, NY, USA, 2007. ACM.

[84] S.-O. Choy and A. K. Lui. Web information retrieval in collaborative tagging systems. In *Proceedings of the 2006 IEEE/WIC/ACM International Conference*

on Web Intelligence (WI '06), pages 352–355, Washington, DC, USA, 2006. IEEE Computer Society.

[85] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[86] F. Dawson and T. Howes. vCard MIME Directory Profile. Request for comments, Internet Engineering Task Force (IETF), Network Working Group, September 1998.

[87] M. Dean and G. Schreiber. OWL, Web Ontology Language. W3C Recommendation, World Wide Web Consortium (W3C), February 2004.

[88] K. Dellschaft and S. Staab. An epistemic dynamic model for tagging systems. In P. Brusilovsky and H. C. Davis, editors, *Proceedings of the 19th ACM Conference on Hypertext and Hypermedia (Hypertext '08), Pittsburgh, PA, USA*, pages 71–80. ACM, 2008.

[89] M. Dorigo and G. Di Caro. The ant colony optimization meta-heuristic. *New ideas in optimization*, pages 11–32, 1999.

[90] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the Web. In *Proceedings of the 10th international conference on World Wide Web (WWW '01)*, pages 613–622, New York, NY, USA, 2001. ACM.

[91] N. Eiron, K. S. McCurley, and J. A. Tomlin. Ranking the web frontier. In *Proceedings of the 13th international conference on World Wide Web (WWW '04)*, pages 309–318, New York, NY, USA, 2004. ACM.

[92] F. Eisterlehner, A. Hotho, and R. Jäschke, editors. *ECML PKDD Discovery Challenge 2009, Bled, Slovenia*, September 2009.

[93] R. Fagin, R. Kumar, and D. Sivakumar. Comparing top k lists. In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms (SODA '03)*, pages 28–36, Philadelphia, PA, USA, 2003. Society for Industrial and Applied Mathematics.

[94] C. Fellbaum and I. NetLibrary. *WordNet: an electronic lexical database*. MIT Press USA, 1998.

[95] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, and T. Berners-Lee. Hypertext Transfer Protocol–HTTP/1.1 (RFC 2616). Request For Comments, Internet Engineering Task Force (IETF), June 1999.

[96] R. T. Fielding and R. N. Taylor. Principled design of the modern web architecture. In *Proc. of the 22nd Int. Conf. on Software Engineering (ICSE '00)*, pages 407–416, New York, NY, USA, 2000. ACM Press.

[97] C. S. Firan, W. Nejdl, and R. Paiu. The Benefit of Using Tag-based Profiles. In *Proc. of 2007 Latin American Web Conference (LA-WEB '07)*, pages 32–41,

Washington, DC, USA, 2007. IEEE Computer Society.

[98] M. Frank. Applikationsübergreifende Wiederverwendung von Empfehlungsfunktionalitäten im Web. Master thesis, L3S Research Center, 2009.

[99] S. Geisser. The predictive sample reuse method with applications. In *Journal of the American Statistical Association*, pages 320–328. American Statistical Association, June 1975.

[100] J. Gemmell, A. Shepitsen, M. Mobasher, and R. Burke. Personalization in folksonomies based on tag clustering. In S. S. Anand, B. Mobasher, A. Kobsa, and D. Jannach, editors, *Proceedings of the 6th Workshop on Intelligent Techniques for Web Personalization and Recommender Systems*, July 2008.

[101] A. Gertner, J. Richer, and T. Bartee. Contact recommendations from aggegrated on-line activity. In B. Mobasher, D. Jannach, and S. S. Anand, editors, *Proceedings of the 8th Workshop on Intelligent Techniques for Web Personalization & Recommender Systems (ITWP '10) co-located with International Conference on User Modeling, Adaptation and Personalization (UMAP), Hawaii, USA*, June 2010.

[102] S. A. Golder and B. A. Huberman. The Structure of Collaborative Tagging Systems. *CoRR*, abs/cs/0508082, 2005.

[103] S. A. Golder and B. A. Huberman. Usage patterns of collaborative tagging systems. *Journal of Information Science*, 32(2):198–208, 2006.

[104] I. P. Goldstein. The genetic graph: a representation for the evolution of procedural knowledge. *International Journal of Man-Machine Studies*, 11(1):51 – 77, 1979.

[105] T. Gruber. Where the Social Web Meets the Semantic Web. In I. F. Cruz, S. Decker, D. Allemang, C. Preist, D. Schwabe, P. Mika, M. Uschold, and L. Aroyo, editors, *International Semantic Web Conference*, volume 4273 of *Lecture Notes in Computer Science*, page 994. Springer, 2006.

[106] T. Gruber. Collective knowledge systems: Where the Social Web meets the Semantic Web. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(1):4–13, 2008.

[107] T. R. Gruber. A translation approach to portable ontology specifications. *Journal of Knowledge Acquisition – Special issue: Current issues in knowledge modeling*, 5:199–220, June 1993.

[108] I. Guy, I. Ronen, and E. Wilcox. Do you know?: recommending people to invite into your social network. In *Proceedings of the 13th international conference on Intelligent user interfaces (IUI '09)*, pages 77–86, New York, NY, USA, 2009. ACM.

[109] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen. Combating web spam with trustrank. In *Proceedings of the Thirtieth international conference on Very large*

data bases (VLDB '04), pages 576–587. VLDB Endowment, 2004.

[110] H. Halpin, V. Robu, and H. Shepherd. The Complex Dynamics of Collaborative Tagging. In *Proc. of 16th Int. World Wide Web Conference (WWW '07)*, pages 211–220, New York, NY, USA, 2007. ACM Press.

[111] E. Hammer-Lahav. The OAuth 1.0 Protocol. Request for comments, Internet Engineering Task Force (IETF), April 2010.

[112] T. H. Haveliwala. Topic-Sensitive PageRank: A Context-Sensitive Ranking Algorithm for Web Search. *IEEE Transactions on Knowledge and Data Engineering*, 15(4):784–796, 2003.

[113] T. Heath and E. Motta. Revyu.com: a Reviewing and Rating Site for the Web of Data. In J. Golbeck and P. Mika, editors, *Semantic Web Challenge*, volume 295 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2007.

[114] D. Heckmann, T. Schwartz, B. Brandherm, M. Schmitz, and M. von Wilamowitz-Moellendorff. Gumo - the general user model ontology. In *Proc. of the 10th Int. Conf. on User Modeling*, pages 428–432, Edinburgh, UK, 2005.

[115] M. Heckner, M. Heilemann, and C. Wolff. Personal information management vs. resource sharing: Towards a model of information behavior in social tagging systems. In E. Adar, M. Hurst, T. Finin, N. S. Glance, N. Nicolov, and B. L. Tseng, editors, *Proceedings of the Third International Conference on Weblogs and Social Media (ICWSM '09), San Jose, California, USA*. The AAAI Press, May 2009.

[116] J. Hendler, N. Shadbolt, W. Hall, T. Berners-Lee, and D. Weitzner. Web Science: an interdisciplinary approach to understanding the Web. *Communications of the ACM*, 51(7):60–69, 2008.

[117] I. Herman, H. Halpin, S. Hawke, E. Prud'hommeaux, D. Raggett, and R. Swick. W3C Semantic Web Activity, 2010. http://www.w3.org/2001/sw/.

[118] P. Heymann, D. Ramage, and H. Garcia-Molina. Social tag prediction. In S.-H. Myaeng, D. W. Oard, F. Sebastiani, T.-S. Chua, and M.-K. Leong, editors, *SIGIR*, pages 531–538. ACM, 2008.

[119] I. Horrocks, B. Parsia, P. F. Patel-Schneider, and J. A. Hendler. Semantic web architecture: Stack or two towers? In F. Fages and S. Soliman, editors, *Proceedings of the 3rd International Workshop on Principles and Practice of Semantic Web Reasoning (PPSWR '05), Dagstuhl, Germany*, volume 3703 of *Lecture Notes in Computer Science*, pages 37–41. Springer, September 2005.

[120] A. Hotho, D. Benz, R. Jäschke, and B. Krause, editors. *ECML PKDD Discovery Challenge 2008, Antwerp, Belgium*, September 2008.

[121] A. Hotho, R. Jäschke, C. Schmitz, and G. Stumme. BibSonomy: A Social Book-

mark and Publication Sharing System. In *Proc. First Conceptual Structures Tool Interoperability Workshop*, pages 87–102, Aalborg, 2006.

[122] A. Hotho, R. Jäschke, C. Schmitz, and G. Stumme. Emergent Semantics in BibSonomy. In C. Hochberger and R. Liskowsky, editors, *Informatik 2006: Informatik für Menschen*, volume 94(2) of *LNI*, Bonn, October 2006. GI.

[123] A. Hotho, R. Jäschke, C. Schmitz, and G. Stumme. FolkRank: A Ranking Algorithm for Folksonomies. In *Proc. of Workshop on Information Retrieval (FGIR)*, Germany, 2006.

[124] A. Hotho, R. Jäschke, C. Schmitz, and G. Stumme. Information retrieval in folksonomies: Search and ranking. In *Proc. of the 3rd European Semantic Web Conference*, volume 4011 of *LNCS*, pages 411–426, Budva, Montenegro, June 2006. Springer.

[125] T. Iofciu, P. Fankhauser, F. Abel, and K. Bischoff. Identifying users across social tagging systems. Technical report, L3S Research Center, 2010.

[126] A. Jameson. Modelling both the Context and the User. *Personal and Ubiquitous Computing*, 5:29–33, 2001.

[127] A. Jameson. Adaptive interfaces and agents. *The HCI handbook: fundamentals, evolving technologies and emerging applications*, pages 305–330, 2003.

[128] R. Jäschke, L. B. Marinho, A. Hotho, L. Schmidt-Thieme, and G. Stumme. Tag recommendations in folksonomies. In *Proc. 11th Europ. Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, pages 506–514, 2007.

[129] G. Jeh and J. Widom. SimRank: A Measure of Structural-Context Similarity. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '02)*, pages 538–543, New York, NY, USA, 2002. ACM.

[130] G. Jeh and J. Widom. Scaling personalized web search. In *Proceedings of the 12th international conference on World Wide Web (WWW '03)*, pages 271–279, New York, NY, USA, 2003. ACM.

[131] T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '02)*, pages 133–142, New York, NY, USA, 2002. ACM.

[132] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. 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.

[133] J. Kahan and M.-R. Koivunen. Annotea: an open RDF infrastructure for shared Web annotations. In *Proceedings of the 10th international conference on World*

Wide Web (WWW '01), pages 623–632, New York, NY, USA, 2001. ACM.

[134] S. Kamvar, T. Haveliwala, C. Manning, and G. Golub. Exploiting the Block Structure of the Web for Computing PageRank. Technical Report 2003-17, Stanford InfoLab, 2003.

[135] P. Kärger, D. Olmedilla, F. Abel, E. Herder, and W. Siberski. What do you prefer? using preferences to enhance learning technology. *IEEE Transactions on Learning Technologies (TLT)*, 1(1):20–33, 2008.

[136] L. Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18:39–43, 1953. 10.1007/BF02289026.

[137] M. G. Kendall. A New Measure of Rank Correlation. *Biometrika*, 30(1/2):81–93, June 1938.

[138] M.-C. Kim and K.-S. Choi. A Comparison of Collocation-based Similarity Measures in Query Expansion. *Information Processing and Management*, 35(1):19–30, 1999.

[139] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[140] G. Klyne and J. J. Carroll. Resource Description Framework (RDF): Concepts and Abstract Syntax. W3C Recommendation, World Wide Web Consortium (W3C), February 2004. http://www.w3.org/TR/rdf-concepts/.

[141] A. Kobsa. User Modeling: Recent Work, Prospects and Hazards. *Adaptive User Interfaces: Principles and Practice*, 1993.

[142] A. Kobsa and W. Wahlster, editors. *User models in dialog systems*. Springer-Verlag New York, Inc., New York, NY, USA, 1989.

[143] C. Kohlschütter, F. Abel, and D. Skoutas. A Novel Interface for Exploring, Annotating and Organizing News and Blogs Articles. In E. Agichtein, M. Hearst, I. Soboroff, and D. Tunkelang, editors, *3rd Annual Workshop on Search in Social Media (SSM '10), co-located with the ACM WSDM 2010 Conference on Web Search and Data Mining, New York, USA*, February 2010.

[144] C. Kohlschütter, P.-A. Chirita, and W. Nejdl. Efficient Parallel Computation of PageRank. In M. Lalmas, A. MacFarlane, S. Rüger, A. Tombros, T. Tsikrika, and A. Yavlinsky, editors, *Advances in Information Retrieval*, volume 3936 of *Lecture Notes in Computer Science*, pages 241–252. Springer Berlin / Heidelberg, 2006.

[145] Y. Koren. Collaborative filtering with temporal dynamics. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '09)*, pages 447–456, New York, NY, USA, 2009. ACM.

[146] C. Körner, D. Benz, A. Hotho, M. Strohmaier, and G. Stumme. Stop thinking, start tagging: tag semantics emerge from collaborative verbosity. In M. Rappa,

P. Jones, J. Freire, and S. Chakrabarti, editors, *Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010*, pages 521–530. ACM, 2010.

[147] G. Koutrika, F. A. Effendi, Z. Gyöngyi, P. Heymann, and H. 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.

[148] R. Krestel, P. Fankhauser, and W. Nejdl. Latent Dirichlet Allocation for Tag Recommendation. In *Proceedings of the 2009 ACM Conference on Recommender Systems (RecSys 2009)*, New York, NY, USA, October 23-25 2009. ACM.

[149] A. N. Langville and C. D. Meyer. *Google's PageRank and Beyond: The Science of Search Engine Rankings*. Princeton University Press, Princeton, NJ, USA, 2006.

[150] S. Lawrence. Context in web search. *IEEE Data Engineering Bulletin*, 23:25–32, 2000.

[151] S. O. K. Lee and A. H. W. Chun. Automatic tag recommendation for the web 2.0 blogosphere using collaborative tagging and hybrid ANN semantic structures. In *Proc. 6th WSEAS Int. Conf. on Applied Computer Science*, pages 88–93, 2007.

[152] E. Leonardi, F. Abel, D. Heckmann, E. Herder, J. Hidders, and G.-J. Houben. A flexible rule-based method for interlinking, integrating, and enriching user data. In B. Benatallah, F. Casati, G. Kappel, and G. Rossi, editors, *Proceedings of 10th International Conference on Web Engineering (ICWE 2010), Vienna, Austria, July 5-9, 2010*, volume 6189 of *Lecture Notes in Computer Science*, pages 322–336. Springer, 2010.

[153] K. Lerman and L. Jones. Social browsing on flickr. *CoRR*, abs/cs/0612047, 2006.

[154] K. Lerman, A. Plangprasopchok, and C. Wong. Personalizing image search results on flickr. *CoRR*, abs/0704.1676, 2007.

[155] X. Li, L. Guo, and Y. E. Zhao. Tag-based social interest discovery. In *Proc. of the 17th Int. World Wide Web Conference (WWW'08)*, pages 675–684. ACM Press, 2008.

[156] G. Linden, B. Smith, and J. York. Amazon.com Recommendations: Item-to-Item Collaborative Filtering. *IEEE Internet Computing*, 7:76–80, 2003.

[157] D. Liu, X.-S. Hua, L. Yang, M. Wang, and H.-J. Zhang. Tag ranking. In *Proceedings of the 18th international conference on World Wide Web*, WWW '09, pages 351–360, New York, NY, USA, 2009. ACM.

[158] C. Marlow, M. Naaman, D. Boyd, and M. Davis. HT06, tagging paper, taxonomy, flickr, academic article, to read. In *Proc. of the 17th Conf. on Hypertext and Hypermedia*, pages 31–40. ACM Press, 2006.

[159] H. A. Martens and P. Dardenne. Validation and verification of regression in small data sets. In *Chemometrics and Intelligent Laboratory Systems*, pages 99–121. Elsevier, December 1998.

[160] T. Vander Wal. Explaining and showing broad and narrow folksonomies. *http://www.personalinfocloud.com/2005/02/explaining_and_.html*, February 2005.

[161] T. Vander Wal. Folksonomy. *http://vanderwal.net/folksonomy.html*, July 2007.

[162] B. Mehta. Learning from what others know: Privacy preserving cross system personalization. In C. Conati, K. F. McCoy, and G. Paliouras, editors, *User Modeling*, volume 4511 of *Lecture Notes in Computer Science*, pages 57–66. Springer, 2007.

[163] B. Mehta. *Cross System Personalization: Enabling personalization across multiple systems*. VDM Verlag, Saarbrücken, Germany, 2009.

[164] B. Mehta, C. Niederee, and A. Stewart. Towards cross-system personalization. In *International Conference on Universal Access in Human-Computer Interaction, Las Vegas, Nevada, USA (UAHCI '05)*. Lawrence Erlbaum Associates, 2005.

[165] Q. Mei and K. Church. Entropy of search logs: how hard is search? with personalization? with backoff? In *Proceedings of the international conference on Web search and web data mining (WSDM '08)*, pages 45–54, New York, NY, USA, 2008. ACM.

[166] P. Merholz. Metadata for the masses. *Adaptive Path*, October 2004. http://www.adaptivepath.com/publications/essays/archives/000361.php.

[167] E. Michlmayr and S. Cayzer. Learning User Profiles from Tagging Data and Leveraging them for Personal(ized) Information Access. In *Proc. of the Workshop on Tagging and Metadata for Social Information Organization, 16th Int. World Wide Web Conference (WWW '07)*, May 2007.

[168] E. Michlmayr, S. Cayzer, and P. Shabajee. Add-A-Tag: Learning Adaptive User Profiles from Bookmark Collections. In *Proc. of the 1st Int. Conf. on Weblogs and Social Media (ICWSM '06)*, March 2007.

[169] P. Mika. Ontologies Are Us: A unified model of social networks and semantics. In *Proceedings of the International Semantic Web Conference (ISWC 2005)*, pages 522–536, November 2005.

[170] V. Milicic. Case study: Semantic tags. W3C Semantic Web Case Studies and Use Cases, December 2008. http://www.w3.org/2001/sw/sweo/public/UseCases/Faviki/.

[171] R. Newman. Tag Ontology. Namespace document, http://www.holygoat.co.uk/projects/tags/, December 2005.

[172] M. G. Noll and C. Meinel. Web search personalization via social bookmarking and tagging. In *Proceedings of the 6th international Semantic Web and 2nd Asian*

conference on Asian Semantic Web conference (ISWC'07/ASWC'07), pages 367–380, Berlin, Heidelberg, 2007. Springer-Verlag.

[173] B. Nowack. OpenSocial/RDF. Namespace document, http://web-semantics.org/ns/opensocial, December 2008.

[174] E. Okon. Entwurf und Implementierung einer Programmbibliothek zur semantischen Erweiterung von RESTful Web Services. Technical report, L3S Research Center, September 2008.

[175] T. O'Reily. What is web 2.0? - design patterns and business models for the next generation of software, September 2005.

[176] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Technical report, Stanford Digital Library Technologies Project, 1998.

[177] A. Passant. LODr – A Linking Open Data Tagging System. In J. Breslin, U. Bojars, A. Passant, and S. Fernández, editors, *First Workshop on Social Data on the Web (SDoW2008)*, volume 405. CEUR Workshop Proceedings, 2008.

[178] A. Passant and P. Laublet. Meaning Of A Tag: A collaborative approach to bridge the gap between tagging and Linked Data. In *Proceedings of the WWW 2008 Workshop Linked Data on the Web (LDOW2008), Beijing, China*, Apr 2008.

[179] J. Pitkow, H. Schütze, T. Cass, R. Cooley, D. Turnbull, A. Edmonds, E. Adar, and T. Breuel. Personalized search. *Communications of the ACM*, 45(9):50–55, 2002.

[180] E. Prud'hommeaux and A. Seaborne. SPARQL Query Language for RDF. W3c recommendation, W3C, January 2008.

[181] F. Qiu and J. Cho. Automatic identification of user interest for personalized search. In *Proceedings of the 15th international conference on World Wide Web (WWW '06)*, pages 727–736, New York, NY, USA, 2006. ACM.

[182] T. Rattenbury, N. Good, and M. Naaman. Towards automatic extraction of event and place semantics from Flickr tags. In *Proceedings of the 30th International ACM SIGIR Conf. on Information Retrieval (SIRIR '07)*, pages 103–110, New York, NY, USA, 2007. ACM Press.

[183] D. Recordon and D. Reed. Openid 2.0: a platform for user-centric identity management. In *DIM '06: Proceedings of the second ACM workshop on Digital identity management*, pages 11–16, New York, NY, USA, 2006. ACM.

[184] T. Reenskaug. Thing-Model-View-Editor: an Example from a Planning System. Technical note, Xerox PARC, May 1979.

[185] S. Rendle, L. Balby Marinho, A. Nanopoulos, and L. Schmidt-Thieme. Learning optimal ranking with tensor factorization for tag recommendation. In *Proceedings*

of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '09), pages 727–736, New York, NY, USA, 2009. ACM.

[186] S. Rendle and L. Schmidt-Thieme. Pairwise interaction tensor factorization for personalized tag recommendation. In *Proceedings of the third ACM international conference on Web search and data mining (WSDM '10)*, pages 81–90, New York, NY, USA, 2010. ACM.

[187] E. Rich. Users are individuals: individualizing user models. *International Journal of Man-Machine Studies*, 18(3):199 – 214, 1983.

[188] E. Rich. User modeling via stereotypes. *Readings in intelligent user interfaces*, pages 329–342, 1998.

[189] L. Richardson and S. Ruby. *RESTful Web Services*. O'Reilly, May 2007.

[190] S. E. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '94)*, pages 232–241, New York, NY, USA, 1994. Springer-Verlag New York, Inc.

[191] B. C. Russell, A. B. Torralba, K. P. Murphy, and W. T. Freeman. LabelMe: A Database and Web-based tool for Image Annotation. *International Journal of Computer Vision*, 77(1-3):157–173, 2008.

[192] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[193] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web (WWW '01)*, pages 285–295, New York, NY, USA, 2001. ACM.

[194] L. Sauermann, R. Cyganiak, and M. Völkel. Cool URIs for the Semantic Web. Technical Memo TM-07-01, DFKI GmbH, Kaiserslautern, February 2007.

[195] J. B. Schafer, J. Konstan, and J. Riedl. Recommender systems in e-commerce. In *Proceedings of the 1st ACM conference on Electronic commerce (EC '99)*, pages 158–166, New York, NY, USA, 1999. ACM.

[196] A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '02)*, pages 253–260, New York, NY, USA, 2002. ACM.

[197] S. Sen, J. Vig, and J. Riedl. Tagommenders: connecting users to items through tags. In *Proceedings of the 18th international conference on World Wide Web (WWW '09)*, pages 671–680, New York, NY, USA, 2009. ACM.

[198] A. P. Sheth, S. Staab, M. Dean, M. Paolucci, D. Maynard, T. W. Finin, and K. Thirunarayan, editors. *The Semantic Web - ISWC 2008, 7th International Semantic Web Conference, ISWC 2008, Karlsruhe, Germany, October 26-30, 2008. Proceedings*, volume 5318 of *Lecture Notes in Computer Science*. Springer, 2008.

[199] B. Sigurbjörnsson and R. van Zwol. Flickr tag recommendation based on collective knowledge. In *Proc. of 17th Int. World Wide Web Conference (WWW '08)*, pages 327–336. ACM Press, 2008.

[200] H. A. Simon. On a class of skew distribution functions. *Biometrika*, 42(3/4):425–440, 1955.

[201] K. Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. *Document retrieval systems*, pages 132–142, 1988.

[202] A. Stewart, E. Diaz-Aviles, W. Nejdl, L. B. Marinho, A. Nanopoulos, and L. Schmidt-Thieme. Cross-tagging for personalized open social networking. In C. Cattuto, G. Ruffo, and F. Menczer, editors, *Proceedings of the 20th ACM Conference on Hypertext and Hypermedia (Hypertext 2009), Torino, Italy*, pages 271–278. ACM, 2009.

[203] H. Story, B. Harbulot, I. Jacobi, and M. Jones. FOAF+SSL: RESTful Authentication for the Social Web. In *Proceedings of the First Workshop on Trust and Privacy on the Social and Semantic Web (SPOT '09)*, Heraklion, Greece, June 2009. CEUR-WS.org.

[204] M. Szomszor, H. Alani, I. Cantador, K. O'Hara, and N. Shadbolt. Semantic modelling of user interests based on cross-folksonomy analysis. In Sheth et al. [198], pages 632–648.

[205] L. Terveen and D. W. McDonald. Social matching: A framework and research agenda. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 12(3):401–434, 2005.

[206] J. Thom-Santelli, M. J. Muller, and D. R. Millen. Social tagging roles: publishers, evangelists, leaders. In *Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, CHI '08, pages 1041–1044, New York, NY, USA, 2008. ACM.

[207] E. Tonkin. Searching the long tail: Hidden structure in social tagging. *Proceedings of the 17th SIG Classification Research Workshop*, 2006.

[208] G. Tummarello, R. Delbru, and E. Oren. Sindice.com: Weaving the open linked data. In *Proceedings of the 6th International Semantic Web and 2nd Asian Semantic Web Conference (ISWC'07/ASWC'07)*, pages 552–565, Berlin, Heidelberg, 2007. Springer-Verlag.

[209] L. von Ahn and L. 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.

[210] E. M. Voorhees. Query expansion using lexical-semantic relations. In W. B. Croft and C. J. van Rijsbergen, editors, *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '94)*, pages 61–69, New York, NY, USA, 1994. Springer-Verlag New York, Inc.

[211] D. Winer. RSS 2.0 specification. Technical note, Berkman Center for Internet & Society, July 2003. http://cyber.law.harvard.edu/rss/rss.html.

[212] D. Winkler. Konzeption, Implementierung und Evaluierung von Visualisierungsstrategien zur Gruppierung von Inhalten. Bachelor thesis, L3S Research Center, 2008.

[213] H. Wu, M. Zubair, and K. 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, 2006. ACM Press.

[214] X. Wu, L. Zhang, and Y. Yu. Exploring Social Annotations for the Semantic Web. In *Proc. of 15th Int. World Wide Web Conference (WWW '06)*, pages 417–426, New York, NY, USA, 2006. ACM Press.

[215] B. Xiang, D. Jiang, J. Pei, X. Sun, E. Chen, and H. Li. Context-aware ranking in web search. In *Proceeding of the 33rd international ACM SIGIR conference on Research and development in information retrieval (SIGIR '10)*, pages 451–458, New York, NY, USA, 2010. ACM.

[216] S. Xu, S. Bao, B. Fei, Z. Su, and Y. Yu. Exploring folksonomy for personalized search. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '08)*, pages 155–162, New York, NY, USA, 2008. ACM.

[217] J. Yan, N. Liu, E. Q. Chang, L. Ji, and Z. Chen. Search result re-ranking based on gap between search queries and social tags. In *Proceedings of the 18th international conference on World Wide Web (WWW '09)*, pages 1197–1198, New York, NY, USA, 2009. ACM.

[218] D. Yin, Z. Xue, L. Hong, and B. D. Davison. A probabilistic model for personalized tag prediction. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '10)*, pages 959–968, New York, NY, USA, 2010. ACM.

[219] M. Yudelson, P. Brusilovsky, and V. Zadorozhny. A user modeling server for contemporary adaptive hypermedia: An evaluation of the push approach to evidence propagation. In *11th International Conference on User Modeling (UM '07)*, pages 27–36, 2007.

[220] N. Zang, M. B. Rosson, and V. Nasser. Mashups: Who? What? Why? In

M. Czerwinski, A. Lund, and D. Tan, editors, *Proceedings of Conference on Human Factors in Computing Systems on Human factors in computing systems (CHI '08)*, pages 3171–3176, New York, NY, USA, 2008. ACM.

List of Figures

List of Tables

Curriculum Vitae

Fabian Abel born in Hannover, Germany, July 8, 1980.

Research Career:

July 2000 *Abitur*, Hannah Arendt Gymnasium, Barsinghausen.

October 2002 – May 2003 *Lecturer*, Multi-Media Berufsbildende Schulen (MMBbS), Design College, Hannover.

July 2003 – September 2003 *Internship*, Philips Semiconductors GmbH, Hamburg.

October 2003 – February 2007 *Research Assistant*, Distributed Systems Institute (IVS) and L3S Research Center, Gottfried Wilhelm Leibniz University Hannover.

October 2004 *Bachelor of Science in Applied Computer Science*, Gottfried Wilhelm Leibniz University Hannover. Context-aware Information Providing in the Semantic Web.

February 2007 *Master of Science in Computer Science*, Gottfried Wilhelm Leibniz University Hannover. Thesis: An Agent-based approach to Distributed Reasoning.

March 2007 – October 2010 *PhD Student & Researcher*, Distributed Systems Institute (IVS) - Semantic Web Group and L3S Research Center, Gottfried Wilhelm Leibniz University Hannover.

November 2010 – now *Postdoc*, Delft University of Technology (TU Delft), the Netherlands. Employed as postdoc at the Web Information Systems group.

April 2011 *Ph.D. in Computer Science*, Gottfried Wilhelm Leibniz University Hannover. Thesis: Contextualization, User Modeling and Personalization in the Social Semantic Web.