EXPLOITING TAG INFORMATION FOR SEARCH AND PERSONALIZATION

Von der Fakultät für Elektrotechnik und Informatik
der Gottfried Wilhelm Leibniz Universität Hannover
zur Erlangung des Grades

DOKTORIN DER NATURWISSENSCHAFTEN

Dr. rer. nat.

genehmigte Dissertation
von

Dipl.-Ing. Raluca Paiu

geboren am 2. Dezember 1980, in Bukarest, Rumänien

2009

Referent: Prof. Dr. Wolfgang Nejdl
Korreferent: Prof. Dr. Heribert Vollmer
Tag der Promotion: 18. November 2009

ABSTRACT

With the increasing popularity of Web 2.0 sites, the amount of content available online is multiplying at a rapid rate, at the same time becoming also more diverse in terms of content types – pictures, music, Web pages, *etc.* – and quality. Professional and user-generated content are quite tightly merged together, such that for users it becomes difficult to spot only the high quality items perfectly matching their interests or current needs. On the other hand, collaborative tagging has become an increasingly popular means for sharing and organizing Web resources, leading to a huge amount of user generated metadata. Some previous works started to make use of this metadata for various purposes, though for improving users' access to information it is not yet obvious whether and how these tags or subsets of them can be used. In this thesis we investigate these questions in detail and, based on the outcomes of this analysis, propose a number of applications of tags for supporting search and personalization.

We start with an in-depth study of tagging behaviors and motivations for different kinds of resources and systems – Web pages (*Del.icio.us*), music (*Last.fm*) and images (*Flickr*) – being thus the first to present a thorough analysis of tag distributions and characteristics across multiple tagging environments. We analyze the implications of tags for search applications and show which types of tags are mostly employed for tagging and searching and which are the most easily remembered by users. Based on these observations, we propose a number of methods for automatically identifying the most valuable types of tags for search, evaluation results indicating the high potential of these methods in enabling further improvement of systems making use of social tags.

We continue discussing the use of tags for personalization applications and tackle two different aspects: personalized music recommendations and personalized Web search. For the former aspect we touch, we make use of collaboratively created user tags, while for the latter, expert annotations extracted from the ODP online catalog are employed. Extensive experiments analyzing both approaches show them to yield improved results over collaborative filtering and regular Google search, respectively.

Finally, we exploit tags for automatically inferring valuable knowledge about the resources tags are attached to. We focus on the multimedia domain and propose three algorithms relying on tags and other social information and aiming at identifying different features of multimedia resources. The three scenarios we discuss target to identify: (1) songs' moods and themes; (2) potential music hits; and (3) landmark pictures. The results of the algorithms' evaluations we performed are promising and provide new insights into the potential such methods have in enabling easier access to content and improving multimedia retrieval.

Keywords: *Web 2.0, Information Retrieval, Personalization*

ZUSAMMENFASSUNG

Mit der zunehmenden Popularität von Web 2.0 Seiten multipliziert sich die Menge der online verfügbaren Daten rasant. Gleichzeitig werden die Web-Daten immer vielfältiger im Hinblick auf Inhalt, wie z.B. Bilder, Musik, Web-Seiten, und Qualität. Professionell sowie nicht professionell erzeugte Inhalte sind so eng miteinander verschmolzen, dass es für Benutzer schwierig wird, nur die hochwertigen Inhalte zu finden, die ihren Interessen entsprechen oder derzeitige Anforderungen erfüllen. Auf der anderen Seite, hat sich kollaboratives Tagging zu einem zunehmend beliebten Mittel zum Austausch und zur Organisation von Web-Inhalten entwickelt, wodurch eine sehr große Menge von Metadaten entstanden ist. Einige von den früheren Studien haben angefangen, diese Metadaten für verschiedene Zwecken zu nutzen. Jedoch ist noch unklar in wieweit diese Tags oder Teilmengen davon zur Verbesserung des Zugangs des Nutzers zu Daten benutzt werden können. In dieser Dissertation untersuchen wir diese Fragen im Detail und schlagen als Ergebnis dieser Analyse vor, Such- und Personalisierungs-Methoden durch Verwendung von Tags zu verbessern.

Wir beginnen mit einer ausführlichen Studie des Verhaltens und der Motivation von Nutzern, Metadaten zu erstellen ("kollaboratives tagging"), bezogen auf verschiedene Arten von Ressourcen und Systemen, wie z.B. Web-Seiten (*Del.icio.us*), Musik (*Last.fm*) und Bilder (*Flickr*). Somit sind wir die ersten, die eine ausführliche Analyse der Verteilung und Eigenschaften von Tags über mehrere Tagging-Umgebungen darstellen. Wir analysieren die Auswirkungen von Tags für Suchanwendungen und zeigen, welche Arten von Tags am häufigsten für Annotation und Suche eingesetzt werden und welche Tag-Typen am leichtesten für die Benutzer zu erinnern sind. Anhand dieser Beobachtungen schlagen wir eine Reihe von Methoden vor, die die besten Tags für Such-Algorithmen automatisch ermitteln können. Die Evaluierungsergebnisse zeigen das große Potenzial dieser Methoden, die Funktionalität von Systemen, die Tags verwenden, zu verbessern.

Wir diskutieren den Einsatz von Tags für Personalisierungsanwendungen und betrachten zwei unterschiedliche Aspekte: personalisierte Musik-Empfehlungen und personalisierte Web-Suche. Für den ersten Aspekt, den wir analysieren, nutzen wir die kollaborativ erzeugten Benutzer Tags. Für den zweiten Aspekt dagegen werden die von Experten erstellten Annotationen aus dem ODP Online Katalog verwendet. Umfangreiche Experimente zeigen, dass beide Ansätze verbesserte Ergebnisse im Vergleich mit kollaborativem Filtering, beziehungsweise Google-Suche liefern.

Letztendlich nutzen wir Tags um wertvolle Erkenntnisse über Ressourcen, die mit den Tags assoziiert sind, zu gewinnen. Wir konzentrieren uns auf den Multimedia-Bereich und entwickeln drei verschiedene Algorithmen, basierend auf Tags und anderen sozialen Informationen, die als Ziel die Identifizierung verschiedenen Eigenschaften von Multimedia-Ressourcen haben. Die drei Szenarien, die wir analysieren, versuchen Stimmungen und Themen von Liedern zu identifizieren, potenzielle Musik-Hits vorherzusagen, sowie Bilder von Sehenswürdigkeiten zu finden. Die Evaluationsergebnisse von unseren Algorithmen sind vielversprechend und geben neue Einblicke in das Potenzial solcher Methoden zur Erleichterung des Zugangs zu Inhalten sowie zur Verbesserung des Multimedia Retrieval.

Schlagwörter: *Web 2.0, Information Retrieval, Personalisierung*

ACKNOWLEDGMENTS

First, I would like to thank my supervisor, Prof. Dr. Wolfgang Nejdl for giving me the opportunity of being part of L3S Research Center and Gottfried Wilhelm Leibniz University of Hannover. With his excellent guidance he taught me the key points of how excellent research must be pursued. I would also like to thank him for the continuous support, which allowed me to attend many interesting conferences and project meetings and thus helped me deepen my knowledge in this field.

I would also like to thank Prof. Dr. Heribert Vollmer, my second supervisor, for providing very useful comments on the draft of my thesis, as well as Prof. Dr. Gabriele von Voigt for agreeing to be part of my dissertation committee.

I am very grateful to Prof. Dr. Valentin Cristea, one of the best Professors I met at the Politehnica University of Bucharest, who believed in me and supported my arrival at L3S Research Center and Gottfried Wilhelm Leibniz University of Hannover.

I would also like to thank to the many colleagues I cooperated with, either from the Gottfried Wilhelm Leibniz University of Hannover, or from other universities and institutes, for their support and valuable comments not limited just to this thesis. Many thanks to the colleagues working in the administrative departments, especially to Anca Vais, Marion Wicht and Iris Zieseniss, for their support and help with many issues related to university administration.

I am also very grateful to Ionescu Aurelia for teaching me the German language and thus contributing to a much easier adaptation to the life in Germany. To Teodor Daneţ, a great Mathematics teacher, for believing in my intellectual abilities and contributing to the development of my analytical thinking.

A special thank to the European Commission for the IST work programme and its frameworks, which supported the research within my thesis, in particular the 6th Framework Programme, and the PHAROS IP project (IST Contract No. 045035).

Last, but definitely not last, I am forever grateful to my family. To my parents, for enduring the distance, for always supporting me in my initiatives and for the excellent guidance in my education and development. To my grandparents, who shaped my way and my intellectual skills starting from the

very early years of my childhood. To my uncle, Titel, a great person and exceptional Physics Professor, for influencing my decision on pursuing this Ph.D. study. To Thomas, for standing by me along this Ph.D., for his love and understanding, support and good advices.

FOREWORD

The algorithms presented in this thesis have been published at various conferences, as follows.

In Chapter 3 we describe contributions included in:

- *Can All Tags Be Used for Search?*. Kerstin Bischoff, Claudiu S. Firan, Wolfgang Nejdl, Raluca Paiu. In: Proceedings of the 17th ACM Conference on Information and Knowledge Mining, CIKM '08, Napa Valley, California, USA, October 26-30, 2008, pp. 193-202, ACM, 978-1-59593-991-3. [BFNP08]

- *Automatically Identifying Tag Types*. Kerstin Bischoff, Claudiu S. Firan, Cristina Kadar, Wolfgang Nejdl, Raluca Paiu. In: Proceedings of the 5th International Conference on Advanced Data Mining and Applications. ADMA'09, Beijing, China, August 17-18, 2009. [BFK+09]

Chapter 4 presenting the use of tags for personalization applications is built upon the work published in:

- *The Benefit of Using Tag-Based Profiles*. Claudiu S. Firan, Wolfgang Nejdl, Raluca Paiu. In: Proceedings of the 5th Latin American Web Congress. LA-WEB '07, October 31 - November 2 2007, Santiago de Chile. [FNP]

- *Using ODP metadata to personalize search*. Paul A. Chirita, Wolfgang Nejdl, Raluca Paiu, Christian Kohlschütter. In Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR '05, Salvador, Brazil, August 2005. [CNPK05]

Finally, in Chapter 5 we structure the presentation around the following papers:

- *Deriving Music Theme Annotations from User Tags*. Kerstin Bischoff, Claudiu S. Firan, Raluca Paiu. In: Proceedings of the 18th International World Wide Web Conference. WWW '09, Madrid, Spain, April 20-24, 2009. [BFP09]

- *How Do You Feel about "Dancing Queen"? Deriving Mood & Theme Annotations from User Tags.* Kerstin Bischoff, Claudiu S. Firan, Wolfgang Nejdl, Raluca Paiu. In: Proceedings of the Joint Conference on Digital Libraries. JCDL '09, June 15-19, 2009, Austin, Texas, USA. [BFNP09]

- *Exploiting Flickr Tags and Groups for Finding Landmark Photos.* Rabeeh Abbasi, Sergey Chernov, Wolfgang Nejdl, Raluca Paiu, Steffen Staab. In: Proceedings of the 31st European Conference on Information Retrieval. ECIR '09, April 6-9, Toulouse, France. [ACN+09]

- *Social Knowledge-Driven Music Hit Prediction.* Kerstin Bischoff, Claudiu S. Firan, Mihai Georgescu, Wolfgang Nejdl, Raluca Paiu. In: Proceedings of the 5th International Conference on Advanced Data Mining and Applications. ADMA'09, Beijing, China, August 17-18, 2009. [BFG+09]

During the early stages of the Ph.D. studies I have also published a number of papers investigating the use of metadata for improving desktop search. This aspect is not touched in this thesis due to space limitation, but the complete list of publications follows:

- *Leveraging Personal Metadata for Desktop Search – The Beagle++ System.* Enrico Minack, Raluca Paiu, Stefania Costache, Gianluca Demartini, Julien Gaugaz, Ekaterini Ioannou, Paul A. Chirita, Wolfgang Nejdl. In: Journal of Web Semantics. To appear (2009). [MPC+09]

- *Personalizing PageRank-Based Ranking over Distributed Collections.* Stefania Costache, Wolfgang Nejdl, Raluca Paiu. In: Proceedings of the 19th International Conference on Advanced Information Systems Engineering. CAiSE '07, June 2007, Trondheim, Norway. [CNP07]

- *The Beagle++ Toolbox: Towards an Extendable Desktop Search Architecture.* Ingo Brunkhorst, Paul A. Chirita, Stefania Costache, Julien Gaugaz, Ekaterini Ioannou, Tereza Iofciu, Enrico Minack, Wolfgang Nejdl, Raluca Paiu. In: Proceedings of the Semantic Desktop and Social Semantic Collaboration Workshop at the International Semantic Web Conference, ISWC '06, November 2006, Athens, GA, USA. [BCC+06]

- *Beagle++: Semantically Enhanced Searching and Ranking on the Desktop.* Paul A. Chirita, Stefania Costache, Wolfgang Nejdl, Raluca Paiu. In: Proceedings of the 3rd European Semantic Web Conference. ESWC '06, June 2006, , Budva, Montenegro. [CCNP06]

- *Peer-Sensitive ObjectRank - Valuing Contextual Information in Social Networks.* Andrei Damian, Wolfgang Nejdl, Raluca Paiu. In: Proceedings of the 6th International Conference on Web Information Systems Engineering. WISE '05, November 2005, New York, NY, USA. [DNP05]

- *Keywords and RDF Fragments: Integrating Metadata and Full-Text Search in Beagle++.* Tereza Iofciu, Christian Kohlschütter, Wolfgang Nejdl, Raluca Paiu. In: Proceedings of the Workshop on the Semantic Desktop - Next Generation Personal Information Management and Collaboration Infrastructure at the International Semantic Web Conference. ISWC '05, Galway, Ireland, November 2005. [IKNP05]

- *Semantically Enhanced Searching and Ranking on the Desktop.* Paul A. Chirita, Stefania Costache, Wolfgang Nejdl, Raluca Paiu. In Proceedings of the International Semantic Web Conference Workshop on the Semantic Desktop - Next Generation Personal Information Management and Collaboration Infrastructure. ISWC '05, Galway, Ireland, November 2005. [CCNP05]

- *Semantically Rich Recommendations in Social Networks for Sharing, Exchanging and Ranking Semantic Context.* Stefania Costache, Wolfgang Nejdl, Raluca Paiu. In: Proceedings of the 4th International Semantic Web Conference, ISWC '05, Galway, Ireland, November 2005. [CNP05b]

- *Desktop Search - How Contextual Information Influences Search Results & Rankings.* Wolfgang Nejdl, Raluca Paiu. In Proceedings of the 2nd SIGIR Workshop on Information Retrieval in Context (IRiX), Salvador, Brazil, August 2005. [NP05a]

- *Semantically Rich Recommendations in Social Networks for Sharing and Exchanging Semantic Context.* Stefania Costache, Wolfgang Nejdl, Raluca Paiu. In: Proceedings of the 2nd European Semantic Web Conference Workshop on Ontologies in P2P Communities, ESWC '05, Greece, May 2005. [CNP05a]

- *Activity Based Metadata for Semantic Desktop Search.* Paul A. Chirita, Stefania Costache, Rita Gavriloaie, Wolfgang Nejdl, Raluca Paiu. In: Proceedings of the 2nd European Semantic Web Conference, ESWC '05, Crete, Greece, May 2005. [CGG+05]

- *I know I stored it somewhere - Contextual Information and Ranking on our Desktop.* Wolfgang Nejdl, Raluca Paiu. In: Proceedings of the 8th International Workshop of the DELOS Network of Excellence on Digital Libraries on Future Digital Library Management Systems (System Architecture & Information Access), March - April 2005, Dagstuhl, Germany. [NP05b]

Contents

List of Figures

Chapter 1

Introduction

The amount of data available on the Web, in organizations and enterprises is multiplying at a rapid rate and as a result users often find themselves overwhelmed by the excess of information. During the past decade, the Web has become a universal repository of human knowledge and culture, which allows sharing of resources and ideas at an unprecedent scale. Any user can create her own Web documents and make them point to any other pages, without any restriction. Moreover, with the increasing popularity of social Web sites (*e.g. Del.icio.us*[1], *Facebook*[2], *YouTube*[3], *etc.*), blog publishing tools (*e.g.* Apache Roller[4], Blosxom[5], LiveJournal[6], *etc.*) and free blog hosting sites (*e.g.* Blogger[7], LifeLogger[8], *etc.*), more and more user generated content becomes available, in addition to the so far dominant professional content. These are key aspects, which turn the Web into a new publishing medium accessible to anybody.

Regarded as the next generation of the World Wide Web, Web 2.0 – a term which became popular starting with 2004 – does not refer to an update to any technical specifications of the Web, but rather to cumulative changes in the ways software developers and end-users utilize the Web, according to Tim O'Reilly. Web 2.0 websites allow users to do more than just retrieve information. They can build on the interactive facilities of "Web 1.0" to provide "Network as platform" computing, allowing users to run software-applications entirely through a browser.

[1] Delicious. http://delicious.com
[2] Facebook. http://www.facebook.com
[3] YouTube. http://www.youtube.com
[4] Apache Roller. http://roller.apache.org
[5] Blosxom. http://blosxom.sourceforge.net
[6] LiveJournal. http://www.livejournal.com
[7] Blogger. https://www.blogger.com
[8] LifeLogger. http://www.lifelogger.com

Because of its accessibility, this universe with almost no frontiers – Web 2.0 – has attracted from the very beginning the attention of millions of users. Furthermore, it also changed the way people use computers and perform their daily tasks. For instance, if until recently, before booking a hotel users would have asked their friends or colleagues about their experiences and impressions regarding that hotel, nowadays they read online opinions about the hotel, made available by a multitude of other Web users. Last but not least, the Web is slowly turning into an extension of the users' own desktop environment: personal photos are published online for making them available to friends or to the public at large (*e.g. Flickr*[9], *Picasa*[10]), bookmarks are managed on the Web, classified and labeled with the aid of tags and shared with other users (*e.g. Del.icio.us*), travels are planned and shared online (*e.g. Tripit*[11]).

1.1 Problems Addressed in this Thesis

Despite so much success, both the Web and Web 2.0 have introduced problems on their own. Finding useful information in such a vast amount of content is often a very tedious and difficult task. When searching very large collections, such as the Web, it is often the case that there are several thousands or even millions of documents matching the goal of the users' searches. Users are thus frequently faced with the situation of having to inspect several dozens of items before finding the right object matching their information need. The main obstacle is the absence of a well defined data model for the Web, which implies that information definition and structure is frequently of low quality [MRS08]. Moreover, the increasing share of multimedia content poses additional challenges on search, multimedia content being much more difficult to process than textual resources. Last but not least, due to the ever increasing amount of online available user generated content, which strongly merges with content published by professionals, for users it becomes more and more difficult to quickly select only the high quality items, also matching their search requests. Because of all these issues, for many users search tasks often become so complicated that they even frustrate all their efforts.

For solving these problems additional techniques are needed, namely *Ranking* and *Personalization*. Ranking is needed for putting a meaningful order in the long lists of results lists, such that the most relevant items occur within the first returned results and are easily accessible by the users. Personalization at the other end is necessary in order to best serve the users' information needs according to their profiles or to the task at hand. This thesis will thus aim at solving the following problems:

Problem 1. *How to support users discover information through advanced search and ranking mechanisms?*

[9] Flickr. http://www.flickr.com/
[10] Picassa. http://picasaweb.google.com
[11] Tripit. http://www.tripit.com/

The mass publishing of information available online is essentially useless, unless this wealth can be discovered and consumed by other users. Early attempts at making Web information "discoverable" fell into two broad categories: (1) full-text index search engines such as Altavista[12], Excite[13] and Infoseek[14] and (2) taxonomies populated with Web pages in categories, such as Yahoo! Directory[15]. The former presented the user with a keyword search interface supported by inverted indexes and ranking mechanisms. The latter allowed the user to browse through a hierarchical tree of category labels. While this is at a first glance a convenient and intuitive method for finding Web pages, it has a number of drawbacks: first, accurately classifying Web pages into taxonomy tree nodes is for the most part a manual editorial process, which is difficult to scale with the size of the Web. Arguably, we only need to have "high-quality" Web pages in the taxonomy, with only the best Web pages for each category. However, just discovering these and classifying them accurately and consistently into the taxonomy entails significant human effort. Furthermore, in order for a user to effectively discover Web pages classified into the nodes of the taxonomy tree, the user's idea of what sub-tree(s) to seek for a particular topic has to match that of the editors performing the classification. This quickly becomes challenging as the size of the taxonomy grows – *e.g.* the Yahoo! taxonomy tree surpassed 1,000 distinct nodes fairly early on.

Can we offer better solutions for these problems?

Problem 2. *How to provide personalized access to information?*

Personalization has been an increasingly popular approach during the recent years, and much research and development effort has been expended. Researchers from different communities have developed systems with the ability to adapt their behavior to the goals, tasks, interests, and other features of individual users and groups of users. By doing so, personalization becomes a useful tool in the selection and filtering of information for the users, facilitating navigation and increasing the speed of access as well as the likelihood that the users' searches are successful. Unfortunately, current personalization techniques are far from being perfect. For example, many of the available personalization tools require a lot of interaction with their users in order to be able to provide useful personalized features. Providing less intrusive personalization tools which require less interaction and can still offer good results to their users is thus crucial. Moreover, especially when interacting with multimedia content, many users do not have specific queries in mind, but rather prefer to receive personalized recommendations of content relevant to their profiles.

Can we provide such personalization methods?

[12] Altavista. http://www.altavista.com
[13] Excite. http://www.excite.com
[14] Infossek. http://gocee.com/eureka/infoseek.htm
[15] Yahoo! Directory. http://dir.yahoo.com

Problem 3. *How to improve users' access to multimedia content?*

The explosion of multimedia content in databases, broadcasts, streaming media, *etc.* has generated new requirements for more effective access to these global information repositories. Content extraction, indexing, and retrieval of multimedia data continue to be some of the most challenging and fastest-growing research areas. A consequence of the growing consumer demand for multimedia information is that sophisticated technology is needed for representing, modeling, indexing, and retrieving multimedia data [BdVBF07]. Still, current available techniques for multimedia content extraction, indexing and retrieval are very expensive and not mature enough.

Can we offer better solutions addressing the shortcomings of the existent technologies?

1.2 Proposed Solution: Tags

Our proposed solutions to the above mentioned problems are based on *tags* – short textual descriptions attached to content objects, such as Web pages, pictures, videos, *etc.* and voluntarily provided by users. The popularity of Web 2.0 did not only bring more content and more diversity in terms of content types, but also the means to bring some structure into place. Collaborative tagging systems enhance to some extent the initial work of professional humans to categorize Web content (*e.g.* Yahoo! Directory or Open Directory Project). The multitude of tags provided voluntarily by users represent in fact categorizations of the resources along certain dimensions. Even if not all users are equally proficient in assigning textual labels to content, over time certain characteristic patterns (*i.e.* Power Law distributions [HA99]) occur inside tagging systems and the most valuable and suitable tags become predominant.

In this thesis we will analyze in detail the characteristics of different tagging systems and propose *novel methods using tags for supporting search and ranking, personalization and improved access to multimedia content.* Research for efficient ranking and personalization algorithms is necessary for quite a lot of application environments, *e.g.* the World Wide Web, Enterprise Networks, Digital Libraries, Social Networks, Multimedia Repositories, *etc.* For all these and especially for the domain of multimedia search, current ranking and personalization algorithms are still rather poor or even inexistent, although at the same time they are more and more required due to the rapidly growing amount of data stored and searched for each of these particular scenarios.

All algorithms proposed in this thesis for improving ranking and personalization rely on tags and are applicable for either Web or multimedia environments. In the case of Web textual resources, tags usually represent summarizations of the content they annotate and can serve as enhancement of the linkage information for ranking, or for personalization. In the case of multimedia resources, tags are even much more valuable, since processing multimedia content is by far more expensive than index-

ing and retrieving textual items. Besides, for multimedia objects the only textual descriptions are either manual metadata usually entered by content producers and therefore often missing or even incomplete, or automatically produced metadata created by different available multimedia annotators. Unfortunately, currently available multimedia annotators are still in their infant stage, and therefore the produced annotations are not very reliable. The available tags are thus a rich source of information in this context, enhancing the still error-prone automatic multimedia annotations and at the same time, offering the basis for knowledge discovery.

The contributions of this thesis are manifold: (i) Firstly, we provide a detailed analysis of tags' characteristics and tagging systems and discuss the benefit of tags for search applications in general; (ii) We propose advanced algorithms for personalized ranking and recommendations; and (iii) We present novel algorithms for knowledge discovery in the context of multimedia resources, which thus indirectly support multimedia search and retrieval.

1.3 Thesis Structure

In **Chapter 2** we start by introducing general notions in the context of Web 2.0 and we describe some of the general characteristics of collaborative tagging systems, essential for understanding the rest of the dissertation. More detailed reviews of related work are included in each of the next three chapters, which are centered around the three problems we aim to solve:

The first problem (*Problem 1*) is addressed in **Chapter 3**, where we start with a detailed analysis of different tagging systems and tags' characteristics across different domains. We thus go beyond current research in the area, which so far investigated these issues only with respect to single domains, *i.e.* previous studies inspected characteristics of tags attached to either Web page resources or pictures, but made no cross-domain analyses. We start with a review of relevant literature in Section 3.2 and then the data used for this analysis is introduced in Section 3.3. We inspect the usage of tags in three different tagging systems (Section 3.4) and analyze the implications of tags for search (Section 3.5). Based on the findings in previous sections, in Section 3.6 we propose a number of methods for automatically identifying the most valuable types of tags for search. We evaluate the methods we introduce and we discuss the results of the evaluation in Section 3.6.3.

Chapter 4 tackles another aspect, namely personalization (*Problem 2*). After introducing the reader into the topic (Section 4.1), a detailed review of the literature follows (Section 4.2). Two different aspects are considered in this chapter: personalized music recommendations (Section 4.3) and personalized Web ranking (Section 4.4). We introduce new algorithms for both domains, evaluate their performance and conclude the section with a discussion of the results we obtain.

In **Chapter 5** we aim at improving the access to multimedia content (*Problem 3*)

and propose a number of methods based on tags for discovering information related to multimedia items. A detailed review of relevant articles tackling the aspects we also address in this chapter is included in Section 5.2. Next, we focus on knowledge discovery based on tags, and exemplify within three scenarios: (1) inferring music mood and theme annotations (Section 5.3); (2) identifying potential music hits (Section 5.4); and (3) identifying landmark pictures (Section 5.5). For each of these sections, we present the datasets used, we introduce the methods we propose and finally evaluate them and discuss the results.

Chapter 6 concludes the thesis with an enumeration of the contributions, while also discussing possible future research directions and open challenges associated with these topics.

Chapter 2

General Background

Web 2.0 refers to what it is perceived as the next generation of Web development and Web design and is characterized as facilitating communication, information sharing, interoperability, and collaboration on the World Wide Web [Wik]. The term was introduced by Darcy DiNucci in 1999 in her article "Fragmented Future" [DiN99], but the term is nowadays closely associated with Tim O'Reilly due to the O'Reilly Media Web 2.0 conference from 2004. Examples of Web 2.0 applications include social networking sites, video sharing sites, wikis, blogs or collaborative tagging systems.

Even if the term "Web 2.0" suggests a technical enhancement of the Web, it actually refers to the cumulative changes in the ways software developers and end-users utilize the Web. An important characteristic of Web 2.0 sites is that they allow users to do much more than just retrieve information. They can build on the interactive facilities of "Web 1.0" to provide "Network as platform" computing, allowing users to run software-applications entirely through a browser [O'R05]. Users can own data within Web 2.0 sites and exercise control over it. Moreover, another important characteristic is that Web 2.0 sites through their functionalities encourage the users to update their content and improve the applications as they interact with them. According to David Best [Bes06], the characteristics of Web 2.0 are: rich user experience, user participation, dynamic content, metadata, Web standards and scalability. Further characteristics, such as openness, freedom and collective intelligence by way of user participation, can also be viewed as essential attributes of Web 2.0.

The popularity of the term Web 2.0 increased along with the increasing usage of blogs, wikis and social networks. However, in this chapter we will not concentrate on these types of Web 2.0 applications, but rather on *Collaborative Tagging Sites*.

2.1 Collaborative Tagging Systems - A General Characterization

Web-based tagging systems allow users to annotate a particular resource (be it a Web page, a blog post, image, podcast, spreadsheet, *etc.*) with a set of freely selectable keywords – tags. These annotations describe most of the times characteristics of the resources they are attached to, are often in a highly structured form and therefore facilitate information access and organization.

Before the advent of collaborative tagging systems, annotations were created in principle solely by dedicated professionals. For example, catalogers create metadata, often in the form of Machine-Readable Cataloging (MARC) records for books and other intellectual creations, and this is the basis of most Online Public Access Catalogs (OPAC) in libraries and other institutions. This often requires serious education and training [Mat04]. Professionally created annotations are of high quality, nevertheless it is very costly in terms of time and effort to produce them. This makes this type of annotations difficult to scale up with the rapidly growing amount of content becoming available especially on the World Wide Web. The Dublin Core Metadata Initiative has been introduced in order to solve exactly this scalability problem: original creators of the intellectual material provide also metadata along with their creations. This approach solves the scalability problems to some extent, but both approaches suffer from the same basic problem: the intended and unintended eventual users of the information are disconnected from the process.

With the growing popularity of social tagging sites, much of the annotation effort has been taken up by community users, who collaboratively and voluntarily attach keyword descriptions to digital content. Social tagging systems allow users to share their tags for particular resources. In addition, tags serve as links among resources tagged the same way by several users. Because of their lack of predefined taxonomic structure, social tagging systems rely on shared and emergent social structures of the community users. Based on this observation, tags in social tagging systems have recently been termed as *"folksonomies"*, the term *"folksonomy"* being coined by Thomas Vander Wal [Wal05] and resulted as a combination of the terms *folk* and *taxonomy*.

Folksonomies refer to the bottom-up classifications that emerge from social tags, *i.e. user taxonomies*. As folksonomies arise in Internet-mediated social environments, users can discover who used a given tag for a particular resource and explore what other tags this user also employed. In this way, users can discover the tag sets of another user who tends to interpret and tag content in a way that makes sense to them. The result can be a rewarding gain in the user's capacity to find related content (a practice known as "pivot browsing"). Part of the appeal of folksonomies is their inherent subversiveness: when faced with the choice of the search tools that Web sites provide, folksonomies can be seen as a rejection of the search engine status quo in

favor of tools that are created by the community.

Despite their popularity, folksonomies have been also strongly criticized, mainly because of their lack of a controlled vocabulary, causing sometimes unreliable or inconsistent results. Because tags are freely chosen (instead of taken from a given vocabulary), synonymy (multiple tags for the same concept), homonymy (same tag used with different meanings) and polysemy (same tag with multiple related meanings) are likely to arise, thus lowering the efficiency of content indexing and searching. Other reasons for noise are the lack of stemming (normalization of word inflections) and the heterogeneity of users and contexts. The lack of a hierarchical or systematic structure for the tagging system makes the terms relevant to what they are describing, but often fails to show their relevancy or relationship to other objects of the same or similar type.

Figure 2.1 presents a conceptual model for social tagging systems [MNBD06b]. In

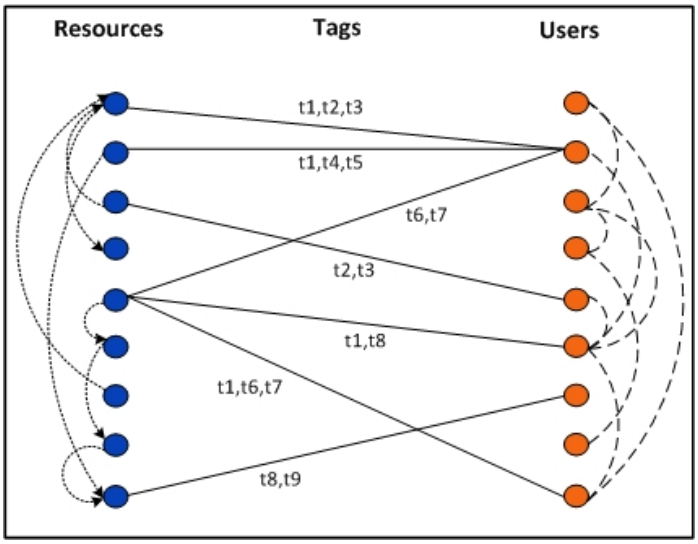

Figure 2.1 A model of collaborative tagging system

this model, users assign tags to different types of resources and tags are typed edges connecting users and resources. Resources can be connected with each other as well, for example through hyperlinks in the case of Web pages. Connections can also exist among users, through participation inside social networks, or sets of affiliations (*e.g.* users working for the same company). Variations of this model are also possible: links between resources can be absent, and likewise links among users can be missing. However, even with these missing links, one can still observe the implicit connections between users, resources and tags – *i.e.* connections among resources exist through the users who tag them and similarly, users are connected through the resources they annotate.

Formally, a tagging system S is represented as a quadruple of the form:

$$S = (U, T, R, Y) \tag{2.1}$$

modeling the relations between users, tags and resources. In Equation 2.1, U represents the set of users, T is the set of tags, R is the set of resources and $Y \subseteq U \times T \times R$ is a ternary relation over U, T and R. If a user $u \in U$ used tag $t \in T$ to annotate a resource $r \in R$, then there is a relation $(u, t, r) \in Y$. This is also called tag assignment [HJSS06b].

These three different types of entities, part of any tagging system, and depicted in the model from Figure 2.1 have been studied independently in the past, usually in the context of Web-based systems. For example, in the case of *resources*, much research has been done in the area of link analysis, PageRank [PBMW98] being one of the most prominent link analysis methods. For *users*, analysis of social ties and social networks, as subfields of sociology, have received a lot of attention both from physicists, economists and computer scientists. Finally, for tags, the aggregation and the semantic aspects of tags have been recently discussed at length [GH06].

In the present thesis we adopt a unitary view (*user – tag – resource*) of this model and our research focuses on a so-far less investigated aspect: we show how tags can be used to improve search and personalization applications. In the next chapters we present the details of our analyses.

Chapter 3

Tags' Characteristics and Implications for Search

3.1 Introduction

Web 2.0 tools and environments have made collaborative tagging very popular – any user can assign freely selectable words, in the form of keywords or category labels, to shared content to describe and organize these resources. Several of these tagging systems have been acquired by search engine companies – *Flickr* and *Del.icio.us* by Yahoo!, *YouTube* by Google – which now also extend search to these communities.

One of the earliest Web 2.0 applications, *Flickr*, is currently the most popular photo sharing website and online community platform. *Flickr* asks photo submitters to describe images using tags, to allow searchers to (re-)find pictures using place name, subject matter, or other aspects of the picture. For the music domain, *Last.fm*[1] is the world's largest social music platform, with over 20 million active users based in more than 230 countries. Since August 2005, *Last.fm* supports tagging of artists, albums, and tracks to create a site-wide folksonomy of music. Tags can describe genre ("garage rock"), mood ("chill"), artist characteristic ("baritone"), or any other form of user-defined classification ("seen live"). *Del.icio.us* is the premiere social bookmarking Web site for storing, sharing, and discovering Web bookmarks, where users can tag each of their bookmarks with freely chosen keywords. A combined view of everyone's bookmarks with a given tag is available and users can view bookmarks added by similar-minded users.

However, all tags available is these various tagging platforms represent quite of few different aspects of the resources they describe and it is not obvious whether and how

[1] Last.fm. http://www.last.fm

these tags or subsets of them can be used for search. Moreover, users' motivations for tagging resources, as well as the types of assigned tags differ across systems. Prior studies, which started to investigate these aspects identified that the factors which induce these differences refer to the type of resources subject to be annotated, or to the systems design choices, such as functionality of displaying other peoples tags or not. As a result, different tag types and distributions of tags in these categories emerge inside tagging systems and their potential to improve search remains unclear, despite initial investigations.

First studies have started to investigate tagging motivations and patterns, usually for one specific collection, including some initial work on how to support the tagging process and improve information retrieval algorithms in general using tags. There are no studies so far investigating these questions across different collections, and there is only limited research regarding the usefulness of tags for search. *Do tags provide new information about the content they annotate, or do they just replicate what is already available from content or other metadata? What kinds of tags are used, and which types can improve search most? Can we automatically identify valuable tags?* These are some of the questions we target to answer in this chapter and we analyze tag data from three very different tagging systems: *Del.icio.us*, *Flickr* and *Last.fm*, as well as anchor texts from a Stanford Web crawl.

We analyze and classify sample tags from these systems, for getting an insight into what kind of tags are used for different types of resources, and for providing statistics on tag distributions in all tagging environments considered. We also measure the informative potential of tags by checking the overlap they have with content or with metadata assigned from experts or originating from other sources. Last but not least, we analyze the potential of different types of tags for improving search, by comparing them with the user queries posted to search engines or gathered through a user survey. After this in depth analysis and after identifying the most useful types of tags for search, we propose a set of methods for automatically classifying tags and so automatically spotting the high quality ones. The algorithms presented in this chapter thus focus on addressing *Problem 1* presented in Chapter 1, namely supporting users discover information through advanced search mechanisms.

The rest of this chapter is structured as follows: in Section 3.2 we present some preliminary existing work in the different areas we address, namely tagging motivations and types of tags, using tags for search and automatic tag classification. Next, in Section 3.3 we present the datasets which we analyzed and the different characteristics that we identify across several tagging systems are included in Section 3.4. We explore the implications tags have on search-oriented applications in Section 3.5 and finally propose a number of methods to automatically classify tags into a verified taxonomy (Section 3.6). Finally, a discussion on the results and methods presented in this chapter is included in Section 3.7.

3.2 Specific Background

Recent scientific work has started examining tagging behaviors, tag types and automatic tag classification, though many studies focus only on one specific collaborative tagging system [GH06, HRS07, SLR+06, AN07, HKGM07] or only provide first qualitative insights across collections from very small samples [MNBD06a, Zol07]. The next paragraphs give an overview over existing work, from which we started our investigations.

3.2.1 Tagging Motivations and Types of Tags

Analyses of collaborative tagging systems indicate that incentives for tagging are quite manifold and so are the kinds of tags used. [XFMS06, GH06, MNBD06a] are some of the first approaches trying to organize the otherwise flat set of tags into hierarchies. Marlow *et al.* introduce in [MNBD06a] two different organizational taxonomies for tagging systems, one capturing system design properties and attributes and one referring to the users' incentives for tagging. The former taxonomy encompasses seven classes – Tagging rights, Tagging support, Aggregation, Type of object, Source of material, Resource connectivity, and Social Connectivity – and basically describes the different design choices of tagging systems which might later on have a strong impact on the ways users tag and interact inside such platforms. The latter taxonomy described in [MNBD06a] aims at organizing tags along the different motivations of users for employing tags. The authors suggest six such possible classes – Future retrieval, Contribution and sharing, Attract attention, Play and competition, Self presentation, Opinion expression – however, the taxonomy is not verified in any way, nor compared to the more popular taxonomy introduced in [GH06]. Golder and Huberman propose in [GH06] seven tag classes, capturing the possible functions of tags. Nevertheless, this taxonomy is introduced in the context of *Del.icio.us* and the authors do not verify its applicability inside other tagging platforms. Our approach of classifying tags also makes use of this taxonomy, and extends its usefulness to other tagging systems and as well to other resources (apart from Web pages). In [HHLS05], the authors of *Connotea*[2] provide a two-dimensional taxonomy, where the first two facets of the the first dimension represent the identity of taggers – "tag user" and "content creator". Both facets can be classified in a second dimension as either "self" or "others". Other categorization that the authors offer divides the space of tagging systems according to the "audience" (scholarly or general) and the "type of object stored in the system" (URL vs. actual content). This classification is of course bound to the *Connotea* system and not easily extendable to any other platform or resource types. In [XFMS06], the authors introduce a taxonomy of five tag classes: Content, Context, Attribute, Subjective and Organizational tags, that they observe inside the *My Web*

[2] Connotea. http://www.connotea.org

2.0[3] system. Their taxonomy is also not verified and is mainly used in the context of tag suggestions, as basis for suggesting "good" tags to the users. However, no precise measures of the quality of the suggested tags is included in the paper. None of the articles mentioned above made any attempts to automatic tag type classification.

Regarding users' motivations for tagging, [MNBD06a] identified that organizational motivations for enhanced information access and sharing are predominant, though also social motivations can be encountered, such as opinion expression, attraction of attention, self-presentation [GH06, MNBD06a], or providing context to friends [AN07]. Which of those incentives is most characteristic for a particular system seems to vary, depending on tagging rights, tagging support, aggregation model, *etc.*– all influencing why certain tags are used or not. [Zol07] and [GH06] indicate that in free-for-all tagging systems like *Last.fm*, opinion expression, self-presentation, activism and performance tags become very frequent, while in self-tagging systems like *Flickr* or *Del.icio.us* users tag almost exclusively for their own benefit of enhanced information organization. It is suggested that such subjective, socially motivated tags are recognizable by their length and the (high) number of words they consist of [Zol07] – therefore, we also make use of this information for automatically identifying tags from the corresponding subjective categories.

Despite these different tagging motivations and behaviors, stable structures do emerge in collaborative tagging systems [GH06, HRS07, HJSS06b]. The evolving patterns follow a scale-free power law distribution, indicating convergence of the vocabulary to a set of very frequent words, coexisting with a long tail of rarely used terms [HRS07, HJSS06b]. Studying the evolution of tagging vocabularies in the MovieLens system, [SLR+06] uses controlled experiments with varying system features to prove how such design decisions heavily influence the convergence process within a group, *i.e.* the proportions "Factual", "Subjective" and "Personal" tags will have. According to these results, being able to display automatically identified "Factual" tags only would lead to even more factual and interpersonally useful tags.

In this chapter we will investigate all these issues for and across several large collections in more detail, comparing and discussing their common characteristics as well as their differences, plus the implications of tagging behavior in different collections on search.

3.2.2 Tags Supporting Search

Based on the idea that tags in bookmarking systems usually provide good summaries of Web pages they annotate and that they indicate the popularity of a page, [BXW+07] investigated the use of tags for improving Web search. The proposed SocialSimilarityRank measures the association between tags and SocialPageRank accounts for the popularity among taggers in terms of a frequency ranking. In [HJSS06b]

[3] My Web 2.0. http://myweb.yahoo.com

the authors suggest an adapted PageRank-like algorithm, FolkRank, to improve efficient searching via personalized and topic-specific ranking within the tag space. This can be used to recommend interesting users, resources and related tags to increase the chance of "serendipitous encounters".

In music retrieval, tags have been used as an alternative or additional possibility to find songs: In [FNP] *Last.fm* songs are not only recommended based on tracklists (song and artist) of similar users, but also by considering (descriptive) tags. Their experimental results showed that tag-based search algorithms provide better and faster recommendation results than traditional track-based collaborative filtering methods.

In [HKGM07], the authors try to answer the question whether social bookmarking data can be used to augment Web search. Their analysis of a *Del.icio.us* dataset shows that tags tend to gravitate toward certain domains and that tags occur in over 50% of the pages they annotate. Only in 20% of the cases tags do not occur in page text, back-link text, or forward-link page text of the pages they annotate. We extend these results, by investigating in detail several different datasets containing tagged data, covering pictures (*Flickr*), music (*Last.fm*) and bookmarks (*Del.icio.us*), and by investigating the potential of different kinds of tags for improving search. As another proof of the usefulness of tags for Web search, recently some tag-based search platforms have become available: Tagvy [TAG] structures the search results around different sources containing tagged resources and matching the corresponding queries of the users. Thus, for a query containing the tag "newyork", a user will get pictures matched in *Flickr*, textual results coming from *Technorati* blog posts and tagged with "newyork", URLs from *Del.icio.us* and news from *Google News*. Similarly, Quitura [Qui] was developed as an alternative search platform that is centered around tag clouds for navigation. Initial results returned for a user's query can be further refined with tags presented as a tag cloud on the left side of the results' list. Recently, tag-search functionality has been included also inside the Firefox browser – the Search Cloudlet [Sea] plugin, developed by the International Software and Productivity Engineering Institute. This plugin adds a click-on tag cloud to Google and Yahoo! searches, helping users to find deep-seated terms and phrases to refine search results.

A different approach is presented in [DEFS06], where tags are used to enhance the quality of enterprise search applications. In addition to collecting user annotations directly from users through a browser toolbar, the authors also propose several strategies for obtaining implicit annotations from search engine query logs. The collected annotations are integrated into a search engine index and used during search, preliminary experiments on the IBM intranet demonstrating that annotations can help to improve the search quality.

Anchor text (AT) or link label is the visible, clickable text in a hyperlink and is also a special kind of tag. AT are broadly and successfully used in Web search engines [Bri98]; the idea of attaching AT to the linked object dates back to the first

Web search engines [McB94] in 1994. Because *AT* accurately describe the content of a linked object [Dav00, Bri98], they can also be used for computing similarity between objects and in particular Web pages [Kan04, SSYC06] or for query refinement [KZ04]. Given that anchor text has been investigated in detail, we compare our results on tags characteristics with those of *AT*.

3.2.3 Automatic Classification of Tags

So far, in the literature there have been only few studies trying to automatically categorize user tags. However, they all focus solely on specific domains and make no statements about the generalizability of their approaches to other areas apart from the original ones. Focusing on the domain of pictures, [RGN07] tries to extract event and place semantics from tags assigned to *Flickr* photos - making use of location (geographic coordinates) and time metadata (time stamp: upload or capture time) associated with the pictures. The proposed approach relies on bursts analysis: tags referring to event names are expected to exhibit high usage patterns over short time periods (sometimes also periodically, like "Christmas"), while tags related to locations show these kinds of patterns in the spatial dimension. The approach yields high precision values especially for identifying place tags from highly popular tags. Still, there are some systematic errors which seem to be introduced by sparse, wrong, or missing data.

In [SvZ08], different tag categories used by users to annotate their pictures in *Flickr* are analyzed automatically. Using the WordNet [Mil95] lexical database the authors are able to classify 52% of their sample tags into the WordNet categories: Location (28%), Artefact/Object (16%), Person/Group (13%), Action/Event (9%), Time (7%) or Other (27%). However, tag classification is not the main focus of the paper, the authors being rather interested in recommending tags to users for supporting them in the annotation process.

Given a set of *Del.icio.us* bookmarks and a set of tags assigned by users, [HRGM08] investigates the predictability of social tags for individual bookmarks. The proposed classification algorithms make use of the page's textual content, anchor text, surrounding hosts, as well as other tags already applied to the URL. With this approach, most tags seem to be easily predictable, page text providing the superior attributes for classification.

In contrast to previous work, we present a general approach to tag type classification demonstrating the performance of our algorithms on collections containing different kinds of resources.

3.3 Datasets

In the following we present the datasets we used for our studies, the methods applied for gathering the data and some basic statistics on tag distributions.

3.3.1 Datasets' Crawling Methods

Last.fm

For our analysis, we have crawled an extensive subset of the *Last.fm* website, a UK-based Internet radio and music community website, founded in 2002 and now owned by CBS Interactive. Statistics of the site claim that 21 million users in more than 200 countries are streaming their personalized radio stations provided by *Last.fm*. The crawl was performed in May 2007, focusing on pages corresponding to tags, music tracks and user profiles. We obtained information about a total number of 317,058 tracks and their associated attributes, including track and artist name, as well as tags for these tracks plus their corresponding usage frequencies. Starting from the most popular tags, we found a number of 21,177 different tags, which are used on *Last.fm* for tagging tracks, artists or albums. For each of these tags we extracted the number of times each tag has been used, the number of users who used the tag, as well as lists of similar tags together with their similarity scores.

Flickr

For comparison with *Flickr* characteristics, we took advantage of data crawled by some of our research partners (University Koblenz/Landau and Tagora Project[4]) during January 2004 and December 2005. The crawling was done by starting with some initial tags from the most popular ones and then expanding the crawl based on these tags. We used a small portion of the first 100,000 pictures crawled, associated with 32,378 unique tags assigned with different frequencies.

Del.icio.us

The *Del.icio.us* data for our analysis was kindly provided by research partners as well (Knowledge and Data Engineering/Bibsonomy at the University of Kassel). This data was collected during July 27 and July 30, 2005 by gathering a first set of nearly 6,900 users and 700 tags from the start page of *Del.icio.us*. These were used to download more data in a recursive manner. Additional users and resources were collected by monitoring the *Del.icio.us* start page. A list of several thousands usernames was collected and used for accessing the first 10,000 resources each user had tagged. From the collected data we extracted resources, tags, dates, descriptions, usernames,

[4]Tagora. http://www.tagora-project.eu/

etc., the resulted collection comprising 323,294 unique tags associated to 2,507,688 bookmarks.

Web Anchor Texts

Although the text in HTML anchors (<a>) is not part of an explicit collaborative tagging system, it represents a similar annotation mechanism. Since most web search engines already use AT to improve results, we compare it with the collaborative tagging systems investigated in this paper. Our dataset consisted of 8,453,043 Web pages parsed from a Stanford WebBase[5] crawl of the Web from January 2006. We extracted 10,348,807 different AT ignoring case. 7,902,047 AT were links to a page in the same domain (*internal*), while 2,756,377 were links to pages in different domains (*external*). Interestingly, only a very small portion, 3% of all AT, were used for both internal and external links.

3.3.2 Tags' Distribution Across Systems

Figure 3.1 presents a comparison of the collaborative tagging systems we analyzed. Usage of tags basically follows a power law distribution for each system. We observe a sharp drop at the end for the *Flickr* and *Last.fm* curves, due to the crawling mechanism which focused more on popular tags.

Disregarding the exact number of tags (this was dependent on each system's architecture and the crawling methods), we analyzed the slopes of the different systems. A more abrupt slope shows that popular tags are being used more often while tags in the tail have less weight. A more gradual inclination indicates a more even use of tags throughout the collection. The most evenly distributed system is *Flickr* where people almost always tag only their own pictures, not much influenced by others. For *Del.icio.us*, influence of others is more visible as the slope gets steeper. *Last.fm* shows the steepest slope, with a few very popular tags and 60% of the top 100 representing genre information. *Last.fm* covers a very specific domain – music – which explains why tags are more restricted than in *Flickr*, where images can include everything and than in *Del.icio.us*, which has an even broader range of topics (in the ODP[6] catalog, about 4 million Web sites are filed into more than 590,000 different categories).

The AT distribution plot shows two visibly different parts with different slopes. The head (top 750 external AT; top 2,000 internal AT) is more even than for all three collaborative tagging systems, while the tail for the external AT is comparable to *Del.icio.us* tags, and for external AT is not a perfect power law distribution. We think this is mainly due to the fact that in our analyzed sample these top AT point to a small set of very popular Web pages. These are external AT sites like Web pages of search engines, important news sites and portals, as well as internal AT links to

[5] WebBase crawls available at http://www-diglib.stanford.edu/~testbed/doc2/WebBase/
[6] "DMOZ" Open Directory Project. http://www.dmoz.org

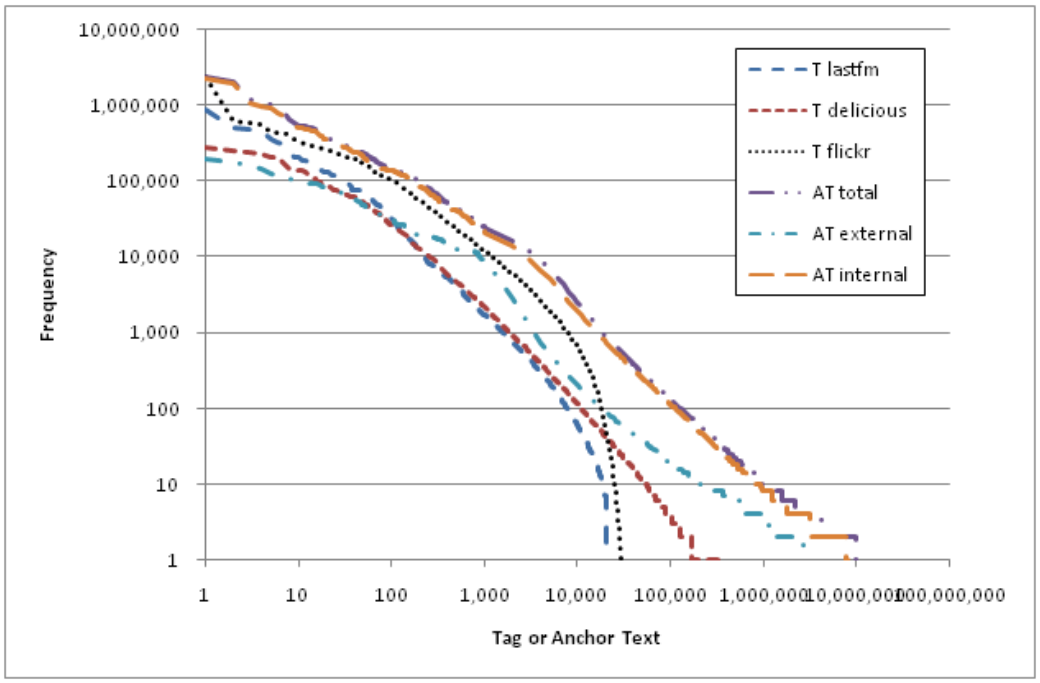

Figure 3.1 Frequency distribution log scale plot comparing tagging systems and anchor text usages

key pages for the Web site like table of contents, site map and home pages. As Figure 3.1 shows, all our datasets exhibit power law characteristics, so that even if the sizes of the collections differ, results are still consistent and comparable.

3.4 Tag Usage in Different Tagging Systems

The following section presents and discusses the results of our comparative investigations of tag usage in *Last.fm*, *Del.icio.us*, *Flickr*, and in *AT*, providing answers to relevant questions and aspects. Looking at the usage of different types of tags, we first identify and quantify the distinctions occurring in users' tagging behavior. Most of the tags are potentially useful for search, though not all kinds of tags are equally valuable. We further investigate reliability of tags and added value, by comparing how well tags correspond to metadata assigned by experts and by looking into the amount of new, non-redundant information provided by tags.

3.4.1 How are Tags Used?

Tags serve various functions based on system features like resource type, tagging rights, *etc.* [MNBD06a], and not all these tags are equally useful for the community or for interpersonal retrieval [GH06]. For being able to improve tag based search, we

first need to know how tags are used and which types of annotations we can expect to find along with resources. For this purpose, we propose and use an extended tag taxonomy appropriate for different tagging systems. This builds on and extends previous work, which has discussed classification schemes for tags, restricted however to only a single tagging system or based on very small data samples. We then investigate tag distributions in different collections, based on our tag classification scheme, and also provide measures on the accuracy of our taxonomy.

Defining Tag Types

We started with an exploratory analysis of existing taxonomies (see [GH06, SLR+06, XFMS06]), as well as possible attribute fields for the different resources to be considered. As a resource can be characterized by different attributes, tag types shed light on what distinctions are important to taggers [GH06]. We kept and refined the most fine-grained scheme presented by Golder & Huberman [GH06], adding the classes *Time* and *Location*, in order to make it applicable to systems other than *Del.icio.us*, which only focuses on Web page annotation. We went through several iterations to improve the scheme by classifying sample tags and testing for agreement between multiple raters as described in Section 3.4.1. Our final taxonomy comprises eight classes, presented together with example tags from our datasets in Table 3.1. Table 3.2 presents an approximate mapping between our taxonomy and other existing tag classification schemes.

Nr.	Category	*Last.fm*	*Flickr*	*Del.icio.us*	*AT*
1	Topic	*love* *revolution*	*people* *flowers*	*webdesign* *linux*	*health* *security*
2	Time	*80s*	*2005, july*	*daily* *current*	*previous years* *tomorrow*
3	Location	*england* *african*	*toronto* *kingscross*	*slovakia* *newcastle*	*great lakes region* *nederlands*
4	Type	*pop, acoustic*	*portrait, 50mm*	*movies, mp3*	*pdf, books*
5	Author/ Owner	*the beatles* *wax trax*	*wright*	*wired* *alanmoore*	*musicmoz.org* *elcel technology*
6	Opinions/ Qualities	*great lyrics* *yum*	*scary* *bright*	*annoying* *funny*	*mobile essentials*
7	Usage context	*workout, study* *lost*	*vacation, birthday* *science*	*review.later, work* *traveling*	*event planning, research* *entertainment*
8	Self reference	*albums i own* *seen live*	*me* *100views*	*sonstiges* *frommyrssfeeds*	*about us* *home page*

Table 3.1 Tag classification taxonomy, applicable to different tagging systems (music resources, pictures, Web pages)

Topic is probably the most obvious way to describe an arbitrary resource, describing what a tagged item is about. For music, *Topic* was defined to include theme (e.g. "love"), title and lyrics. The *Topic* of a picture refers to any object or person displayed. While such *Topic* information can partially be extracted from the content of textual resources [HKGM07], it is not easily accessible for pictures or music. Tags

Nr.	Our Category	Golder et al. [GH06]	Xu et al. [XFMS06]	Sen al. [SLR+06]
1	Topic	What or who it is about	Content-based	Factual
2	Time	*replaced Refining categories*	Context-based	
3	Location			
4	Type	What it is	Attribute	
5	Author/Owner	Who owns it		
6	Opinions/Qualities	Qualities & Characteristics	Subjective	Subjective
7	Usage context	Task organization	Organizational	Personal
8	Self reference	Self reference		

Table 3.2 Mapping between tag classification schemes

in the *Time* category add contextual information about month, year, season, or other time related modifiers. This includes the time a picture was taken, a music piece or Web page was produced. Similarly, *Location* is an additional retrieval cue, providing information about sights, country or town, or the origin of a musician. Tags may also specify the *Type*, which mainly corresponds to file, media or Web page type ("pdf", "blog", *etc.*). In music this category comprises tags specifying format as well as instrumentation and music genre. For pictures, this includes camera settings and photographic styles like "portrait" or "macro". Yet another way to organize resources is by identifying the *Author/Owner* who created the resource (author, artist) or owns it (a music and entertainment group like Sony BMG or a *Flickr* user). Tags can also comment subjectively on the quality of a resource, expressing opinions based on social motivations typical for free-for-all-tagging systems, or are simply used as rating-like annotations for easing personal retrieval. *Usage context* tags suggest what to use a resource for, or the context/task the resource was collected in and grouped by. These tags (*e.g.* "jobsearch", "forProgramming", *etc.*), although subjective, may still be a good basis for recommendations to other users. Last, *Self reference* contains highly personal tags, mostly helpful for the tagger herself. For comparison, we applied this tag classification scheme also to our AT collection – defining *Self reference* in terms of site internal and system-reference comprising frequent navigational AT pointing to pages within the domain or sections of a Web page.

Distribution of Tag Types Across Systems

To make manual tag classification feasible we had to sample our data – we manually investigated 1200 tags in total. For the three different tagging systems as well as for our AT collection, we took three samples of 100 tags each to be manually classified. These three samples per system included the top 100 tags, 100 tags starting from 70% of probability density (based on absolute occurrences), and 100 tags beginning from 90%. These different samples based on rank percentages were chosen based on the results of prior work [HRS07] which suggested that different parts of the power law curve exhibit distinct patterns. Our goal was to provide descriptive statistics about

tag type usage depending on popularity to formulate appropriate hypotheses based on relative frequencies of distinct tag types. Some samples range over slightly more than 100 tags, as some tags had to be skipped as they were completely unreadable or not understandable. Also, since our data sets have different sizes, the long tail is cut off at different points, which may lead to slight shifts in ranks. However, as the long tail consists mostly of idiosyncratic tags with very low usage frequencies, the influence of this adjustment should be negligible.

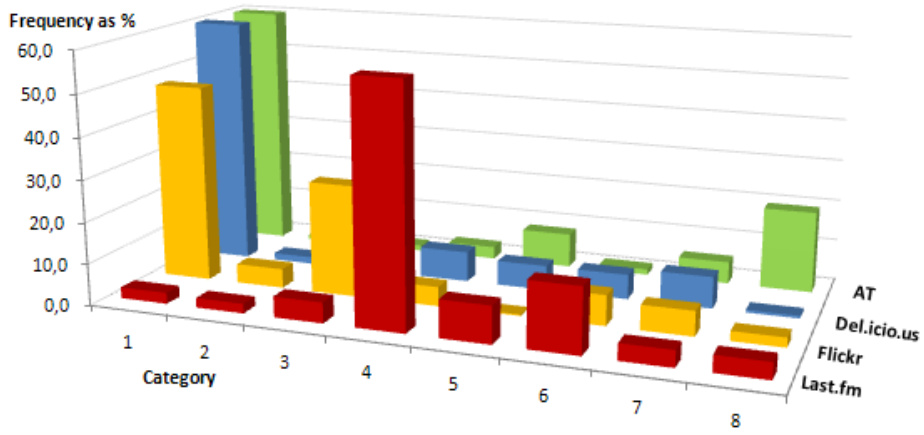

Figure 3.2 Tag type distributions across systems

We classified the different types of tags across different systems. The resulting distributions are shown in the Figure 3.2. The most obvious general conclusion is that tag types are very different for different collections. Specifically, the most important category for *Del.icio.us* and *Flickr* is *Topic*, while for *Last.fm*, the *Type* category is the most prominent one, due to the abundance of genre tags, which fall into this class. Obviously, genre is the easiest way of characterizing and organizing music – one of the rare exceptions was for the theme "love" and some parts of the lyrics/title. In contrast, a similar dominance can be observed for *Topic* in case of Web resources and pictures. *Type* is also common in *Del.icio.us*, as it specifies whether a page contains certain media. As *Flickr* is used only for pictures, *Type* variations only include fine grained distinctions like "macro" – most users do not seem to make such professional annotations. For pictures only, *Location* plays an important role. *Usage context* seems to be more used in *Del.icio.us* and *Flickr*, while *Last.fm* as a free-for-all-tagging system (with lower motivation for organization) exhibits a significantly higher amount of subjective/opinion tags. *Time* and *Self reference* only represent a very small part of the *Flickr* tags studied here. *Author/Owner* is a little more frequent, though very rarely used in *Flickr* due to the fact that people mainly tag their own pictures [MNBD06a]. For *AT*, specifying the *Topic* is the main functional category. External *AT* are mostly titles of pages or very similar to titles. *Self* (or system) *reference* is the second most important function for *AT*; *AT* for internal site navigation falls into this category. *Time*, *Location* and *Opinions/Qualities* are rare

for *AT*.

To better understand how importance of tag categories varies with tag popularity, Figure 3.3 shows the distributions for all systems across all samples.

Again, we observe *Type* as the predominant tag category for music, while for URLs and pictures it is *Topic* – mostly increasing across samples. For the long tail of the *Last.fm* sample, usage of *Type* category decreases, and opinion expression and artist labeling (*Author/Owner*) get more important. For *AT*, internal *Self* or system *reference* decreases in importance for less frequent *AT*. This is probably related to the fact that the vocabulary for many navigational *AT* is highly standardized ("home", "top") and so highly ranked. The same argument holds for types of linked resources.

The type distribution between systems shows a clear tendency of preferred tag functions that do not depend much on the popularity of the tags. With respect to search, it is encouraging to see, that most tags – *Topics* and resource *Type* in general, *Topic* and *Location* for pictures, and to a certain degree *Type* for music – are factual in nature, verifiable and thus potentially relevant to the community and other users. Subjective and personal tags (categories 6, 8) are only a minor part (except for category 8 in *AT*). Similar to results reported in [Zol07], *Opinions/Qualities* are only characteristic for social, free-for-all music tagging systems (like *Last.fm*), possibly because for young people (exposing) music taste is one important aspect in forming one's own personal identity.

Accuracy of Tag Classification

Clearly, such classification schemes only represent one possible way of categorizing things. Quite a few tags are ambiguous due to homonymy (especially for *Flickr* and *Del.icio.us*, e.g. "apple", "shannon") and therefore it is difficult to decide in which of the categories they would fit best. We based our decision on the most popular resource(s) tagged. In general, it was often necessary to check co-occurring tags and associated resources to clarify tag meaning, especially for the very technical *Del.icio.us* bookmarks. During classification we even found some tags considered as 'factual' difficult to classify directly. For example, "vacation" can be considered as the *Topic* of a Web resource, as well as a personal tag of type *Usage context* grouping resources for the next holidays. Similarly "zoo" or "festival" may be depicted in a picture or used as context attributes not directly inferable from the resource. Depending on the intended usage such tags fit into more than one category. This problem of concise category boundaries also applies to the other categorization schemes presented in section 3.4.1. Again, our classification decisions were based on popularity of associated resources.

These observations prompt the interesting questions of how much accuracy, *i.e.* consistency, can be obtained for such a tag classification scheme. Hence, we evaluated inter-rater agreement, to get a quantitative measure on possible accuracy. From our initial sample we selected 75 tags per system (25 randomly chosen tags per range) plus 75 per anchor tags and had them also assessed by students unfamiliar with the

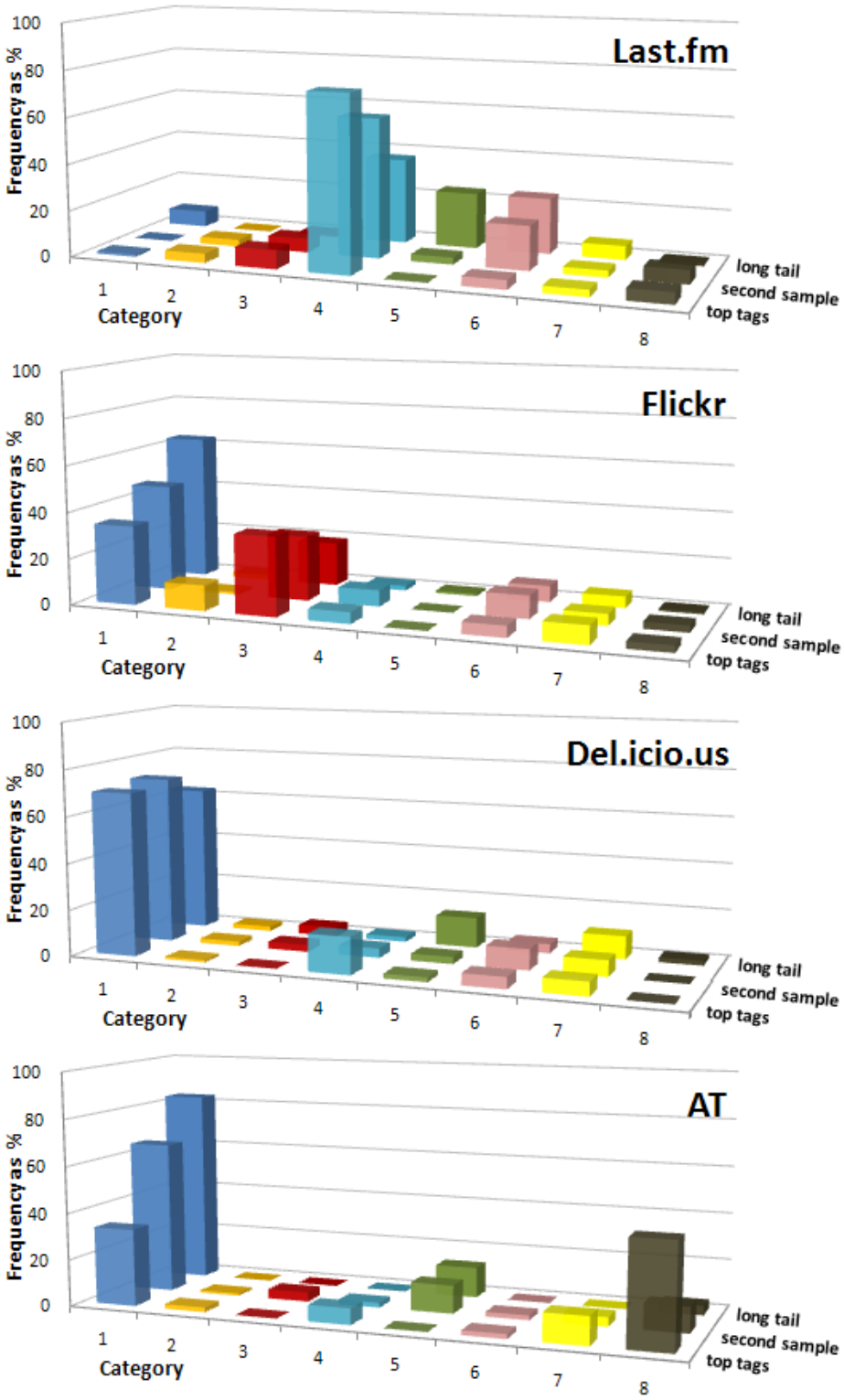

Figure 3.3 Tag type distributions across systems and samples

tag categorization scheme. We computed Cohen's Kappa (κ) [Coh60] which indicates the achieved inter-rater agreement beyond-chance, as the standard measure to assess concordance for our nominal data.

Our raw agreement value for the κ calculation is about 0.79 given the sum of 0.77 for the by chance expected frequencies, resulting in a κ of 0.71 – considered as good and substantial inter-rater reliability [Coh60]. Looking more closely at the values for the individual systems, we found the classification for *Last.fm* the most consistent with actual agreement of 0.85 and κ value of 0.74 (*Flickr*: 0.8 and 0.69; *Del.icio.us*: 0.67 and 0.46; *AT*: 0.83 and 0.7). We observe that for more constrained systems concordance seems to be higher. The partial disagreement observed may be caused by ambiguity of the classification scheme, but also because the *Topic* category is so prevalent: Since this category is very frequently assigned (264 out of 300 ratings) chance agreement (expected frequency) is enormously high which leads to a reduced κ value independent of the actual agreement rate [EG04]. To account for the ambiguity in tag meaning and tag function for certain resources, we gave the rater a chance to name a second category that would fit as well. Taking into account this second possible category for a tag, our κ improved considerably to 0.80 – 0.76 for *Last.fm*, 0.9 for *Flickr*, 0.59 for *Del.icio.us* and 0.75 for *AT* respectively.

It is interesting to investigate how existing tag categorization schemes including ours can be improved further. The confusion matrix created for the κ calculation reveals several prominent confusion patterns for the *Del.icio.us* tags – always involving the 'default' *Topic* category. Specifically, in several cases we found disagreement on whether a tag indicated the *Topic* or *Type*, *Author/Owner* or *Usage context* respectively.

3.4.2 Reliable Metadata Generators: Experts or Users?

Given the huge amount of metadata created through collaborative tagging, another interesting question concerns its reliability: is it worth using tags for search, or should we use instead annotations produced by experts? To answer this question, we compared metadata created by experts against metadata produced by communities of users. The music domain is very suitable for this kind of analysis, since there are a lot of online available music reviews for albums, tracks, and artists, produced by human experts. At the same time, on the *Last.fm* portal, we can find most metadata in the form of tags, assigned to the same kinds of entities (tracks, albums and artists).

Tags in music reviews. In this experiment, we analyzed the overlap between tags assigned to *Last.fm* tracks and music reviews extracted from Google results for the same set of tracks. From the 317,058 *Last.fm* tracks in our original dataset, we randomly selected 8,130 tracks, for which we tried to find music reviews by sending queries in the form [*"artist" "track" music review -lyrics*] to Google – the same query as used in [KPSW07]. For each of the selected tracks we considered the top 100 Google results, and extracted the text of the corresponding pages to create one single

document inside which we searched for the tags corresponding to the track. The tag distribution found was linear and 73.01% of the track tags occurred inside review pages. This overlap is rather high, and probably caused by the fact that most of the *Last.fm* tags represent genre names, which also occur very often in music reviews.

Second, we investigated how many of the tags assigned to tracks occurred in the manually created reviews from www.allmusic.com. We randomly selected music tracks from our *Last.fm* dataset and crawled the Web pages corresponding to their AllMusic reviews. If no review was available for one track, we tried to find the review Web page of the album featuring that track. The resulting dataset consisted of 3,600 reviews. Following the same procedure as for the previous experiment, with reviews crawled from Google results, we found that 46.14% of the tags belonging to a track occurred on the AllMusic review pages. Again the tag distribution we found is linear. We hypothesize that the lower number of matches is due to the fact that AllMusic reviews are created by a relatively small number of human experts, which use a more homogeneous and thus restricted vocabulary than found in arbitrary reviews on the Web. A graphical representation of tag distributions for both Web and AllMusic reviews is given in Figure 3.4.

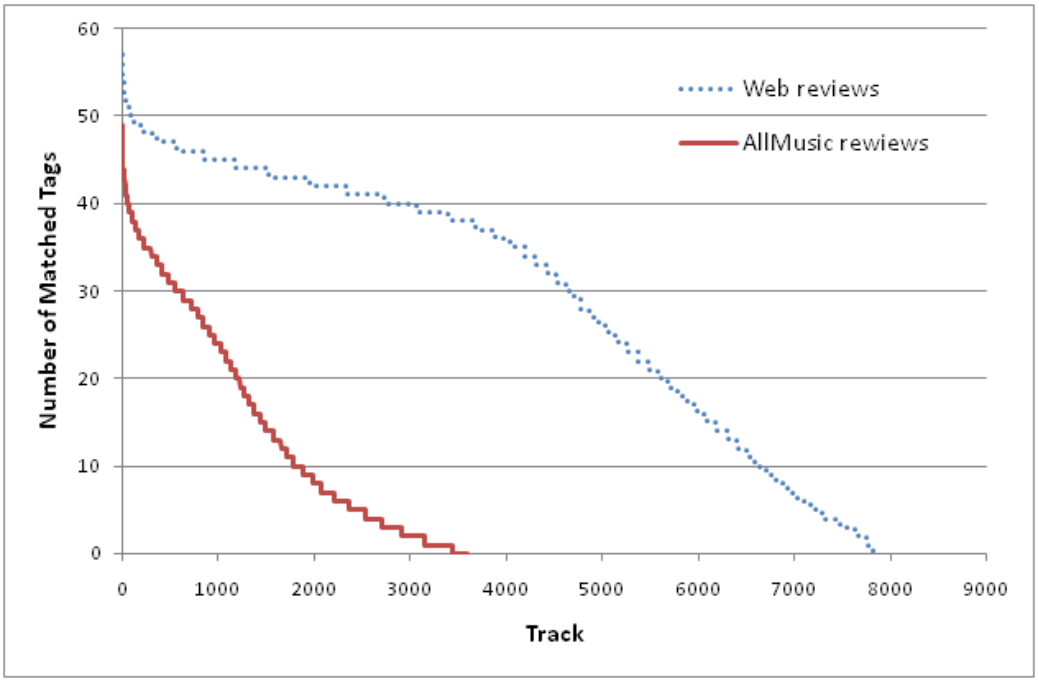

Figure 3.4 Tags Distribution in Web and AllMusic Reviews

3.4.3 Can We Find Tags in Original Content?

It is also interesting to investigate the added value of tags: do they provide new information on the content they annotate, or just replicate what is already available from the content itself?

Tags and *AT* in Web pages. From the *Del.icio.us* crawl we extracted 20,911 URLs for which we had the full HTML page in the WebBase crawl. For these we counted how many tags appear in the Web page text they annotate and found that this is the case for 44.85% of the selected *Del.icio.us* tags. This result is close to the value of 50% found in [HKGM07] and thus confirms their finding. Comparing how many *AT* are present in pages they link to, we analyzed 8,614,990 *AT* and found that 44.7% of external *AT* and 81.24% of internal *AT* are already present in the linked page text. Results are thus similar to those of *Del.icio.us* tags as only external *AT* should be regarded as a collaborative annotation scheme comparable to *Del.icio.us*.

We also computed the overlap between Bookmarks from *Del.icio.us* and the URLs from the Web crawl, and found 77,756 URLs present in both analyzed datasets. When manually comparing the *tags* to the *AT* of the corresponding pages it became obvious that most *AT* look like page titles, while tags relate to page content descriptions. We computed text matches between tags and *AT*. As *Del.icio.us* tags consist only of one word, we found a very low rate, of 4.71%, of the URLs that have at least one exact match between a tag and an *AT*, and 42.52% of the URLs that have at least one partial match, i.e., the tag was contained in the *AT*. For the same overlapping dataset we computed Pearson's correlation coefficient between the frequency of tags and that of *AT*. We found that these frequencies are uncorrelated (for internal *AT* $p = 0.0002$, for external *AT* $p = 0.0411$) although there is a slight, yet insignificant, increase in correlation for external *AT*.

Tags in track lyrics. To get an indication of how often tags are used to describe the theme of songs, we computed the overlap between track tags and track lyrics. The dataset used in this experiment consisted of the intersection of our *Last.fm* collection and a crawl of the site www.lyricsdownload.com. The intersection of the two sets consisted of 77,498 tracks, for which attributes, such as lyrics, name of the track, album featuring the track, and tags assigned by *Last.fm* listeners are available.

To analyze how many of the tags assigned by the users describe what the songs are about, we took all tags corresponding to the tracks and tried to find them in the track lyrics. The distribution of the number of tags, which occur in the text of the corresponding tracks' lyrics, follows a power law distribution: the maximum number of tags which also appeared in the lyrics text was 11, which was the case for only one track; approximately 3,000 tracks had more than 1 tag occurring in the corresponding tracks' lyrics; around 10,300 tracks had only 1 tag that could be exactly matched inside lyrics text and for the rest of about 63,000 tracks none of the tags was found in this "original" content. On average, 1.54% of the tracks' tags occurred in the lyrics – which is in line with our manual tagging classification results (see Section

3.4.1).

3.4.4 Results: Tag Usage Implications on Search

The results presented in Section 3.4.1 show that the tag distributions depend on the resource domain: pictures and Web pages can contain objects referring to any topic, whereas music resources are very restricted in content, leading to a much more focused set of top tags. Analyzing tag types, we were able to show that more than 50% of the tags in *Del.icio.us*, *Flickr* and *AT* are *Topic*-related keywords. As non-subjective annotations, these tags are usable for search by all users, not just the tagger. A probable motivation for using these tags is that Web pages and pictures can belong to any topic category, thus classifying these resources with topics is a very natural way to organize them.

In contrast, for *Last.fm* the *Type* category is predominant: most of the tags correspond to music genres. In *Last.fm* we also find more opinion related tags, whose top tags might be useful for a majority of users, but not for people disagreeing with popular opinion. *Opinion/Quality* and *Author/Owner* are the second and third most used classes for tagging music resources.

Regarding added informational value of tags we observe that *Del.icio.us* tags are, like *AT*, present in 45% of the pages they annotate. Only 43% of *Del.icio.us* tags are included in *AT* for the same URL they point to. This means that over 50% of tags bring new information to items they annotate or describe. In contrast, *Last.fm* tags are usually not contained at all in lyrics (the only textual original content available): the percentage of new tags is 98.5%. Regarding music reviews (another source of information about music, manually created by human experts), at least one *Last.fm* tag occurs in the review texts for almost all analyzed tracks. This proves tags to be a reliable source of metadata about songs, created more easily by a much higher number of users.

3.5 Exploring Tags for Search

Extending and complementing our final discussion in the previous section, we also explored how users' search and tagging behaviors compare. We analyze how much a query log overlaps with tags and conduct a user study which shows what tag types users consider most useful for search and which ones they remember best - being thus easily available to be used as retrieval cues in search.

3.5.1 Do Web Users Search Like They Tag?

In this experiment, we investigated how much current Web queries overlap with tags. We used the AOL query logs [PCT06] to calculate overlap between Web queries and

tags, and contrasted tag and query classes.

First, we counted what percentage of queries consist of tags used in our three systems. Regarding *Del.icio.us*, 71.22% of queries contain at least one *Del.icio.us* tag, while 30.61% of queries consist entirely of *Del.icio.us* tags. This confirms the findings in [HKGM07] that due to the significant overlap *Del.icio.us* tags may help finding Web resources matching queries to tags. For *Flickr* and *Last.fm* the numbers are 64.54% and 12.66%, and 58.43% and 6%, respectively. Here we have to take into account that our tag vocabulary contains 323,294 *Del.icio.us* tags, while we only have 32,378 *Flickr* tags and 21,177 *Last.fm* tags. Nevertheless, we notice that *Del.icio.us* tags (for tagging general resources) appear much more often in queries than *Flickr* or *Last.fm* tags (images or music related tags). Also, tags describing images are used almost twice as much in queries than music related tags.

For our comparative analysis of tags and queries we tried to find the tag classes established before within queries – investigating which kind of tags could best answer a given query. We built a frequency sorted list of all queries in the AOL log and took three samples, as in Section 3.4.1. For comparing system specific behavior, we similarly sampled 300 queries for music and 300 for image queries, by filtering the query log for queries containing a keyword (like "music", "song", "picture" *etc.*) or having a click on *Last.fm* or *Flickr*. The resulting queries were classified into our eight categories, with queries belonging to multiple classes in case they consisted of terms corresponding to different functions. The results are shown in Figure 3.5. Not

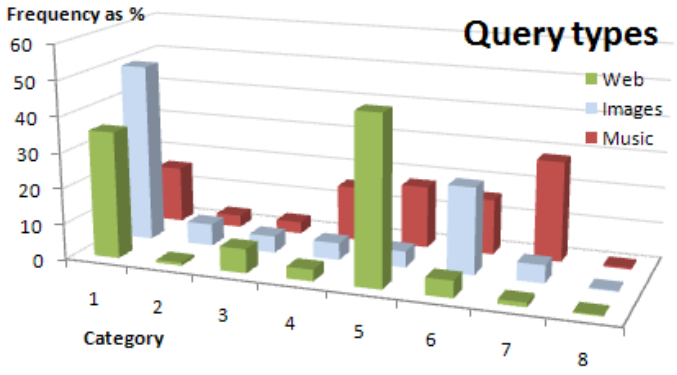

Figure 3.5 Distribution of query types for different resources

quite surprisingly, general Web queries often name the *Topic* of a resource – just like tags in *Del.icio.us* do to an even larger extent. The query distribution pattern seems to fit to tag types except for a clear difference regarding category 5 (*Author/Owner*). *Usage context* is more often used for tag based information organization than for search. For obvious reasons, *Self reference* is not a useful query type for public Web resources.

For images, our tag type distribution almost perfectly corresponds to the query type patterns. As figure 3.5 shows, *Topic* accounts for about half of the queries, as

well as of the tags in *Flickr*. Slight differences exist for *Location*, used more for tagging than for searching and *Author/Owner* being somewhat more important for queries than for tagging. Interestingly, there seem to be many more subjective queries beside the *Topic* asking for *Opinions/Qualities* like "funny", "public" or "erotic" pictures. This however may also be influenced by our samples which often contained queries for adult pictures. With decreasing popularity of tags this category becomes somewhat less important – with increasing emphasis on *Topic* and *Location*.

The biggest deviation between queries and tags occurs for music queries. While our tags in *Last.fm* are to a large extent genre names, user queries belong to the *Usage context* category (like "wedding songs" or "graduation songs", or songs from movie or video games, category 7). Also, users search for known music by artist (category 5) and title or theme (category 1). This difference may be due to information value considerations: as artist and title are already provided in *Last.fm* as formal metadata there is no need to tag resources with this information. In the less frequent tags of *Last.fm* these become more important, so our sampling of popular tags for this system may underestimate their importance. Lyrics are not frequently searched for. An interesting and surprising observation is that searching by genre is rare: Users intensively use tags from this category, but do not use them to search for music. One reason for this might be the fact that many music pieces get tagged with the same genre and thus search results for genre queries would contain far too many hits. Categorizing tracks into genre is also subjective to a certain extent, as it depends on the annotator's expertise. The amount of subjective qualities asked for or tagged is comparable for the *Last.fm* system, with about 16% each.

For music we did not find any structural differences across the three samples that could be generalized. We also remark that we found similar morphological variations and spelling errors in queries as in tags across all resource types.

3.5.2 Which Tags Are Useful and Easily Remembered?

On the Web, users' searching behavior might be different than when searching directly inside a tagging system. We therefore also conducted a user survey on how users describe and remember resources, based on the methodology described in [NHW+04]. Specifically, we wanted to discover:

- Which of the tag categories we proposed is considered most useful for searching;
- Which of the tag categories are most used by users for describing resources;
- Which of the categories are best remembered by users.

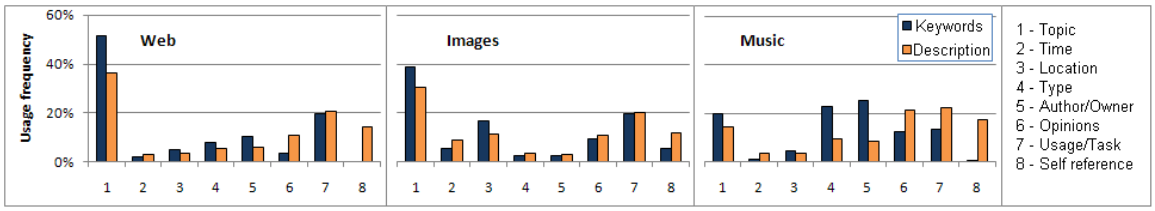

Figure 3.6 User study results: Comparison of category frequencies for keywords and descriptions

User Study – Setup

We had 30 participants in our user study: all of them are researchers and PhD students in Computer science, of different nationalities, 23 men and 7 women, with ages ranging from 23 to 40. The experiment consisted of two different parts: for the first part participants were asked to mentally recall 6 desktop items – 2 pictures, 2 songs and 2 URLs from the users' bookmark list – which they did not access for a long time. For the case that study participants did not have some of the requested items on the personal desktop, we asked them to recall 2 photos which they once saw, 2 songs which they like hearing (e.g. on the radio) and 2 URLs of pages which they once visited and found interesting. Users had to write textual descriptions for each of them, BUT without looking at the pictures/Web pages and without listening to the music. They were requested to write descriptions as detailed as they could. Besides, for each of these resources, users had to provide a set of keywords best describing them.

In the second part of the experiment, participants were provided with the tag category descriptions and examples of tags for pictures, music and Web pages corresponding to each category. They were then asked to answer 3 questions:

- *How useful do you think each of the 8 categories are for searching your own resources?*
- *How useful do you think each of the 8 categories are for searching other peoples' resources?*
- *How well do you remember each of the 8 tag categories?*

For all questions, users rated on a 5-point Likert scale ranging from 0 – meaning "not useful at all" (for the first two questions) / "not remembering at all" (for question 3) – to 4 – corresponding to "very useful for search" and respectively "very good remembering".

User Study – Independent Recall of Tag Types

In the first part of the experiment we aimed at identifying which of the 8 tag categories were used by the users for describing recalled resources - and thus are implicitly most salient and best remembered.

For this part we could only use the data of 24 participants, since for the remaining ones either descriptions or keywords were missing. In total, we had descriptions and keywords for 145 resources (48 pictures, 49 songs and 48 URLs). We identified 1,307 concepts in either keywords or descriptions – Pictures: 291 in descriptions and 208 in keywords; Songs: 243 and 178; URLs: 223 and 164. On average, a picture had 6.06 *concepts* attached in the description and 4.33 in keywords. For songs the numbers are: 4.96 and 3.36 and for URLs: 4.65 and 3.42. The personal pictures elicited more memories written down in detail. This is probably partially due to their personal nature – in contrast to rather public songs and web pages. In general descriptions contained many details about the *Usage context* and *Self reference*.

Figure 3.6 shows the relative frequencies of the different categories for descriptions and keywords. We observe that for URLs *Topic* and *Usage context* are most used in both descriptions and keywords; due to its factual nature *Topic* reaches over 50% for keywords. *Self reference* and *Opinions/Qualities* are also very frequently appearing in descriptions, but not as often in keywords. Looking at the descriptions, this is caused by people telling quite detailed stories about these resources and their associations. The category distribution for pictures is very similar to URLs except for a small drop in *Topic* and an increase in *Location* and *Time*. As we have already seen in earlier sections, the music domain is quite different. When describing songs, people tend to use much more *Opinions/Qualities*, *Usage context* and *Self reference* concepts then when using only keywords. Vice versa keywords are used more for *Type* and *Author/Owner*.

The keywords assigned by our participants thus exhibit similar characteristics found in the analysis of *Flickr*, *Del.icio.us* and *Last.fm* presented in section 3.4.1. For web pages and pictures, actual relative frequencies deviate a little but ordering of category importance is almost the same except for the swap of *Usage context* and *Location* in *Flickr*. For music, we find more significant deviations: while in *Last.fm* *Type* is by far the most important category, the keywords are more often *Author/Owner* than *Type* and *Topic*. This is explained by *Last.fm*'s system features, as artist and title are already provided as formal metadata. Independent of the resource type, *Usage context* is a very well remembered category, which certainly could be exploited and supported more in current tagging systems. Especially for pictures it provides new and only partially subjective information (e.g. "CHI2007", "Universiteit Twente"). Also for music, we found them useful as inter-personal recommendations or associations (e.g. "salsa course").

Except for the type frequencies, their appearance order is also an indicator of importance, as well as of easiness in recalling details pertaining to the different cate-

gories. In our study the participants almost always named *Topic* first in descriptions and keywords. The exception is music where for keywords *Author/Owner* was usually first, followed by *Topic*.

User Study – Assisted Recall and Usefulness of Tag Types

In the second part of the user study, we wanted to get an indication of the users' perception regarding the usefulness of different tag categories for searching both personal and non-personal resources (pictures, music files or Web pages). We also investigated which kinds of tags are best cues in order to recall a resource.

Figure 3.7a presents a detailed comparison of the 30 users' ratings for usefulness and remembering of tag types for images. Ratings are very similar across the different activities of searching personal or public pictures and remembering – except *Time* and *Type*. Our participants remembered *Time* very well for their pictures and found it equally useful to search for them, but for public resources it is less valued as a retrieval cue. Often users do simply not know it. For *Type* it is opposite: Similar to the results of [NHW+04] people seem to find *Type* more useful for searching others' pictures and also remember them, but they do not use it to search their own items – since they do not annotate or describe their pictures with such (semi-)professional photographic aspects. Differences are even smaller for most of the categories for Web resources and music.

As the ratings are very similar across the different activities of searching personal or public resources and remembering (the pairwise Pearson correlation coefficients between the three activities range from 0.85 to 0.97), Figure 3.7b compares the 30 users' ratings only on usefulness for searching public resources across Web pages, images, and music. Values vary across resources. *Topic* is the most useful and best remembered type of information for Web pages, followed by *Usage context*, *Author/Owner* and *Type*. *Self reference*, *Time*, and *Location* are judged neither useful nor well remembered. For pictures on the other hand, while *Topic* is still the most valuable category, the next ones are *Location* and *Time*. *Usage context* and *Type* are judged least important, probably due to perceived subjectivity of context and low (semi-)professional photography knowledge. For music, best ratings were given for *Author/Owner*, *Type*, and *Topic*, the others receiving a rather mediocre or low importance value. *Opinions/Qualities* is considered more useful for searching songs you do not have in your collection than it is for searching your favorite songs. As a surprise, it seems that people assume quite some agreement on subjective characteristics and opinions. We also found this tendency for URLs and pictures, though less pronounced.

Summarizing the results, substantial differences in perceived value of tag types exist only between resource types, each resource type having its own noticeable categories (e.g. *Location* for images). Concerning the activity (remembering or searching for own or other people's resources), there was only minor impact for *Time* or *Type*

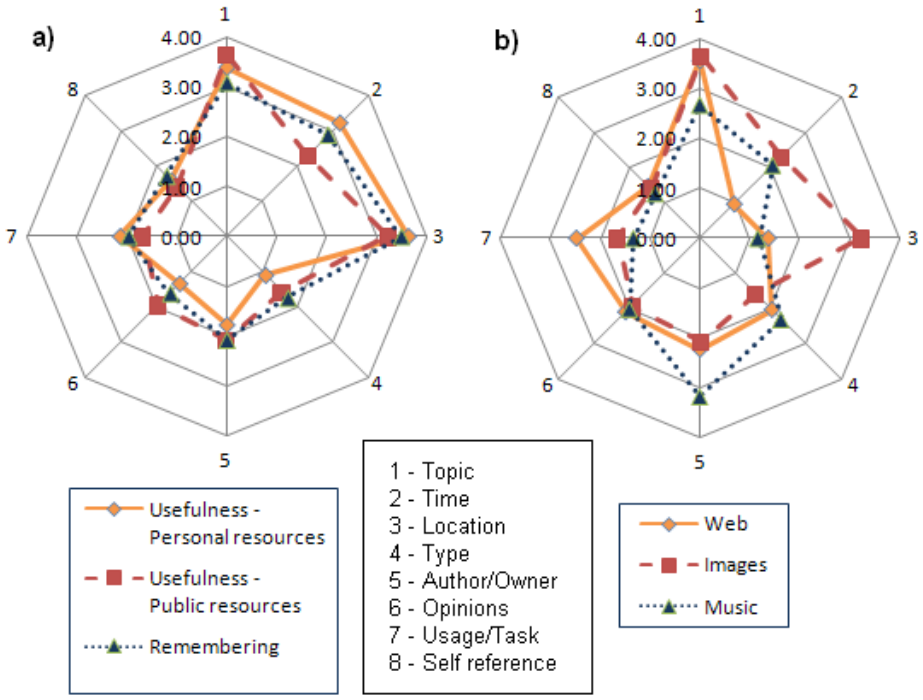

Figure 3.7 User study results: a) Usefulness for personal, usefulness for public, and level of remembrance of images; b) Usefulness for public resources for different resource types

for pictures and *Opinions/Qualities* for music. However, for all resource types users rated 'factual' categories, especially *Topic*, very high. On the other hand, *Usage context* and *Opinions/Qualities* were valued higher than we had intuitively expected.

3.5.3 Results: Usefulness of Tags in Search

Comparing categories of tags and queries as well as perceived usefulness provides some interesting insights: Figure 3.8 summarizes the differences and commonalities of how our eight categories are used for the different resource types. It compares the usage of tags (as described in Section 3.4.1) with the usage of queries (as described in Section 3.5.1) and the perceived level of usefulness by the users for public resources (as described in Section 3.5.2) for each of the categories.[7]

Most of the general Web queries are *Topic*-related queries (as most of the tags for *Del.icio.us*, *Flickr* and *AT*) and this category is also considered by far the most useful for search in particular for Web pages. Except for *Topic* users do not voice major differences in usefulness of the eight categories. However, we observe that some categories are more useful than others. For Web resources *Topic* tags are very useful,

[7]Note that usefulness scores have been scaled from a 0-4 scale to sum up over all categories to 100% in order to be suitable for this comparison.

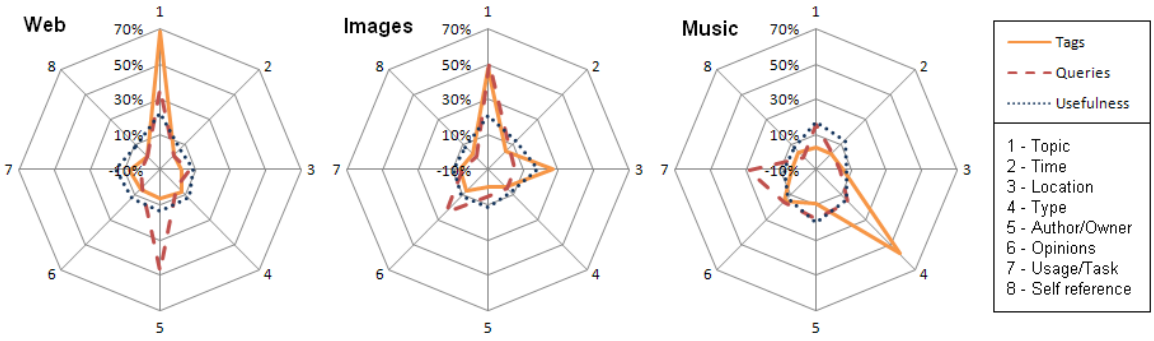

Figure 3.8 Comparison of category usage for tags and queries, and user usefulness assessment

as over 30% of the queries target this category; but we also see that although users query the *Author/Owner* category, they usually do not tag in this way. For images the *Topic* category is considered by experiment participants very useful for search, and tags and queries of this kind are equally present. For images many queries are about *Opinions/Qualities* but users tend to add more *Location* tags than the needed *Opinions/Qualities*. So, even if users actually like to search for "funny" or "strange" pictures and judge them explicitly as partially useful for search, they often do not tag them in this way. As for the music domain, tags generally fall into the *Type* (i.e., genres) class, although more tags from *Usage context* and *Topic* categories would be needed (*Author/Owner* is already present).

3.6 Automatically Identifying Valuable Tag Types

As we have seen in the previous sections, not all tags are equally useful for search: some of the categories are representative for some sets of users, while at the same time, they might not have any positive impact on the search results of others - sometimes, they might even introduce noise. Being able to distinguish between the types of tags associated to resources is thus highly beneficial for search engines in order to best support users in their search needs by extending search to the corresponding tag categories.

We propose two basic approaches to automatically classify sample tags from *Flickr*, *Del.icio.us* and *Last.fm*, depending on the category that needs to be identified. Though some tags could be assigned to more than one functional type, we will categorize each tag according to its most popular type, mainly to make evaluation of automatic classification, *i.e.* human assessment of a ground truth data set feasible. For this, we use both straight forward matching rules against regular expressions and table look-ups in predefined lists, as well as more complex model-based machine learning algorithms. We next describe these methods in detail.

3.6.1 Rule-based Methods

Five of the eight tag type categories (*Time*, *Location*, *Type*, *Author/Owner* and *Self reference*) can be identified by using simple rules, implemented as regular expressions, or table look-ups in predefined lists.

Time. Spotting time-tags was done with the help of both several date/time regular expressions and by using lists of weekdays, seasons, holiday names, *etc.* The same predefined lists where used for all three systems: *Last.fm*, *Del.icio.us* and *Flickr*. This approach can easily capture most time tags – since time vocabulary of the predominately English tags is rather restricted. Other less trivial approaches, *e.g.* detecting time related tags as bursts over short time periods [RGN07], on the other hand, require time related metadata (like upload time) that is not present for all tags in all tagging systems. In total we made use of 19 complex regular expressions containing also 106 predefined time-related expressions (*e.g.* "January", "Thanksgiving", "monthly").

Location. For identifying location tags in *Last.fm*, *Flickr* and *Del.icio.us*, we made use of the extensive knowledge provided by available geographic thesauri, so called gazetteers. From the open source system GATE[8], a tool for Natural Language Processing, a total of 31,903 unique English, German, French and Romanian location related words were gathered. These terms comprised various types of locations: countries (with abbreviations), cities, sites, regions, oceans, rivers, airports, mountains, *etc.* For *Del.icio.us*, the list needed to be slightly adapted by excluding some extremely common words (*e.g.* "java", "nice", "church") in order to assure better accuracy. After this step, the final list of location names contained 31,782 unique entries.

Type. Since the Type category, denoting what kind of resource is tagged, is system/resource dependent, separate lists were used for the three systems. Candidate tags were matched against predefined lists containing common music genres (for *Last.fm*), file and media formats (for *Del.icio.us*) or photographic techniques (for *Flickr*), respectively. A list of 851 music genres gathered from AllMusic[9] was used in order to identify type tags in the *Last.fm* data set. Though it is hard to find agreement on the perfect genre taxonomy, this inventory of genres is highly popular and also used in id3-tags of mp3-files. Without any doubt, the type category is the most numerous in the music portal. As music is only a (small) part of resources tagged in *Del.icio.us*, we gathered a list of 83 English and German general media and file format terms, *e.g. document, pdf, foto, mpg* or *blog, messenger, shortcut*. For *Flickr*, type or genre tags cover besides file formats mostly photographic techniques and styles. Thus, the type list used here included 45 items that describe picture types (like *portrait, landscape* or *panoramic*), photographic techniques (like *black and white, close-up* or *macro*) or other more camera-related words (like *megapixel, shutter*

[8] Gate. http://gate.ac.uk/
[9] AllMusic. http://www.allmusic.com/

or *mm*).

Author/Owner. Identifying whether a tag names the author or owner of a resource, is a relatively easy task for *Last.fm*. From the available online information, *i.e.* the tracks collected, a huge catalog of artist names resulted, against which candidate tags were matched. In case of *Del.icio.us* with its wide variety of Web pages bookmarked, finding the author or owner is not trivial. Since processing of a page's content and possibly extraction of named entities seems a costly procedure, we made use of an inexpensive heuristic assuming domain owners/authors to appear in a Web page's URL. With the help of regular expressions, we could search for the owner or the author of the resource inside the corresponding URL, *i.e.* check if URLs of the resources are formed such as http://xyz.author.com. Most tags from this category where used to ease navigation – *e.g.* clicking the tag "google" to go to the bookmark for http://maps.google.com instead of typing the URL into the browser directly. For *Flickr* classifying tags into the Author/Owner category was not possible, as pictures are mostly personal and no user-related information was included in our data set.

Self reference. For identifying self reference tags from *Last.fm*, we created an initial list of 28 keywords, containing references to the tagger herself in different languages (*e.g.* "my", "ich" or "mia") and her preferences (*e.g.* "favo(u)rites", "love it", "listened"). For *Del.icio.us* we adapted the list to this particular system, and the resulted list contained 31 items including also structural elements of a Web site (like "homepage", "login" or "sonstiges") that do not appear in the music tagging portal. Finally, for *Flickr* the list was adapted to include some personal background references, like "home" or "friends". Each of the list items was searched inside the collection of tags as a separate word with the help of regular expressions.

Such easily handcrafted lists are per se never complete, but automatic extension of the seeding set can be achieved by taking into account similar tags, *e.g.* based on second order co-occurrence as described in the next section. All rule-based methods are run over all tags to be classified, thus enabling machine learning algorithms to discriminate among the residing tags with respect to *Topic*, *Opinion* and *Usage context* (the remaining three of our tag classes).

3.6.2 Model-based Methods

Model-based methods include the family of algorithms using complex machine learning classification techniques. Building a reasonably comprehensive topic register is impossible, since the list of possible topics is practically inexhaustible. The same argument holds for usage context terms and for opinion expressions, in particular if phrases are allowed. Therefore, machine learning techniques are necessary for solving this task. To find the Topic, Usage context and Opinion tags, different binary classifiers were built for each category. They were trained to decide, based on given tag features, whether a tag belongs to the respective tag class or not. Here, we used

classifiers available in the open source machine learning library Weka[10].

Classification features.

For all three systems *Last.fm*, *Del.icio.us* and *Flickr*, we extracted the same features to be fed into the binary classifiers:

- Number of users or tag frequency, respectively
- Number of words
- Number of characters
- Part of speech
- Semantic category membership

Number of users is an external attribute directly associated to each tag, measuring prevalence in the tagging community, and thus indicating a tag's popularity, relevance and saliency. For *Flickr*, we used the absolute usage *frequency* since our data does not contain the necessary user-tag tuples and it can be considered to be an equally useful, though different, indicator of popularity.

Since it has been suggested that, often highly subjective opinion tags in *Last.fm*–like "lesser known yet streamable artists" – exhibit both a higher *number of words* and *number of characters* [Zol07], we used these intrinsic tag features as well for training our classifiers.

Similarly, many of these opinion tags are adjectives, while topic tags are mostly nouns [GH06]. Thus, we included *part of speech* as additional feature. For determining word class, we employ the lexical database WordNet 2.1[11]. In form of a derived tree of hypernyms for a given word sense, WordNet also provides valuable information about the semantics of a tag. The three top level categories extracted from here complete our tag feature set.

For *Last.fm* with its multi-word tags, we collected the latter two features for each word in the tag, *i.e.* we matched all terms individually if the phrase as a whole did not have a WordNet entry.

Other possible resource features, like lyrics of a song, title, description or text of a bookmarked Web page are not considered here as we do not distinguish tag meanings on a per resource basis.

[10] Weka. http://www.cs.waikato.ac.nz/ml/weka/
[11] WordNet. http://wordnet.princeton.edu/

Sense disambiguation and substitution.

For exploiting tag information like part of speech and WordNet category during machine learning, choosing the right sense/meaning of a tag is critical - especially since in English multiple senses of a word like "rock" are spread across different word classes like verb and noun. While statistical or rule-based part-of-speech tagging can be used to partially disambiguate candidate senses for words in sentences, our *Del.icio.us* and *Flickr* sample tags contain just one word – only *Last.fm* supports phrases as tags. Thus, we decided for a different strategy of word sense disambiguation making use of the rich semantic information provided implicitly through tags co-occurring with a given tag in the system.

For the *Last.fm* and *Del.icio.us* sample tags we extracted all co-occurring tags with the corresponding frequencies. However, instead of using such strongly related tags directly, we further wanted to narrow down potential relations by computing second order co-occurrence. For all sample tags, we thus determined similarity with all other tags by calculating pairwise the cosine similarity over vectors of their top-1000 co-occurring tags. A very high similarity indicates that two tags are almost synonymous because they are so frequently used together in the same context / with the same set of tags – the two tags themselves rarely appearing together directly but being mutually exchangeable [CBHS08]. Given an ambiguous tag, we now search for the newly identified similar tags in the definitions, examples and synset words in WordNet. If this does not decide for one meaning, then by default the sense returned by WordNet as most popular is chosen. Since some tags will not be found in WordNet at all, we make further use of these similar tags by taking the most similar one that has a match in WordNet, as a substitute for the original tag.

For *Flickr*, neither disambiguation nor substitution of a tag by its most similar tag could be applied, as we miss the necessary coccurrence relationships between tags. Here, the most popular meaning was used for getting both part of speech and semantic category information from WordNet.

To build models from these features that enable finding Topic, Usage context and Opinion tags from our sample tags, we experimented with various machine learning algorithms Weka offers: Naïve Bayes, Support Vector Machines, C4.5 Decision Trees, *etc.* For each, we moreover used different combinations of the basic features described. As the Weka J48 implementation of C4.5 yielded the best results, only the results obtained with this classifier are presented in the following section on evaluation results.

3.6.3 Results and Discussion

Before discussing the results for both the rule-based and model-based methods in detail, we describe the underlying evaluation procedure and ground truth.

Ground Truth and Evaluation

For evaluating the proposed algorithms, we built a ground truth set containing sample tags from each system that were manually classified into one of the eight categories by a human rater. To make manual tag categorization feasible a subset of 700 tags per system was assessed; thus, we intellectually analyzed 2,100 tags in total.

The samples per system included the top 300 tags, 200 tags starting from 70% of probability density (based on absolute occurrences), and 200 tags beginning from 90% – prior work suggests that different parts of the power law curve exhibit distinct patterns [HRS07].

Clearly, such classification schemes only represent one possible way of categorizing things. Quite a few tags are ambiguous due to homonymy or depending on the intended usage for a particular resource they can fall into more than one category. We based our decision on the most popular resource(s) tagged. For this scheme and method, we had achieved a good and substantial inter-rater reliability – a Cohen's Kappa value, κ, of 0.71. In general, it was often necessary to check co-occurring tags and associated resources to clarify tag meaning, especially for the very technical *Del.icio.us* bookmarks (for details please refer to [BFNP08]).

	Class	Features	Accuracy	% Man.	% Auto.
Del.icio.us	Topic	POS,Cat	81.46	67.14	76.00
	Time	RegEx,List	100.00	0.86	0.86
	Location	List	97.71	3.86	3.86
	Type	List	93.71	8.00	5.14
	Author	RegEx	70.20	6.29	2.14
	Opinion	Num,POS,Cat	93.40	5.14	0.00
	Usage	POS,Cat	89.66	7.86	0.14
	Self ref.	List	99.00	0.86	0.29
	Unknown				*11.57*
Flickr	Topic	Freq,POS,Cat	79.39	46.07	45.92
	Time	RegEx,List	98.86	4.72	4.15
	Location	List	86.70	26.18	21.89
	Type	List	95.99	5.29	1.72
	Author	N/A		0.14	
	Opinion	Num,POS,Cat	93.21	7.44	5.87
	Usage	Num,POS,Cat	85.48	7.58	4.58
	Self ref.	List	97.85	2.58	0.43
	Unknown				*15.45*
Last.fm	Topic	Freq,Num	90.32	2.43	0.00
	Time	RegEx,List	99.14	1.29	1.29
	Location	List	97.43	8.29	7.71
	Type	List	77.14	51.14	33.71
	Author	List	88.65	8.14	3.29
	Opinion	Freq,Num,POS,Cat	74.73	17.71	18.43
	Usage	POS,Cat	79.57	6.43	5.29
	Self ref.	List	98.71	4.57	3.71
	Unknown				*26.57*

Table 3.3 Best results for rule-based and model-based methods. *(Features: POS=part of speech, Cat=WordNet categories, Freq=tag frequency, Num=number of words and characters, RegEx=regular expressions, List=list lookup)*

The ground truth sets of 700 sample tags are then classified by first running the rule-based and then the model-based methods as described in the previous section. For measuring the performance of our tag type classification algorithms we use classification accuracy. For the model-based methods we perform a 10-fold cross-validation on the samples, and for the rule-based method we compute the accuracy by determining the number of true/false positives/negatives. Table 3.3 summarizes results structured per system for all classes. It shows the best performing features, the achieved accuracy, and the percentage of tags (*i.e.* the sample of 700 tags) belonging to a certain category: both the real, manual value ("Man.") and the predicted, automatic value ("Auto."). A graphic representation of the accuracies per class and tagging system is given in Figure 3.9.

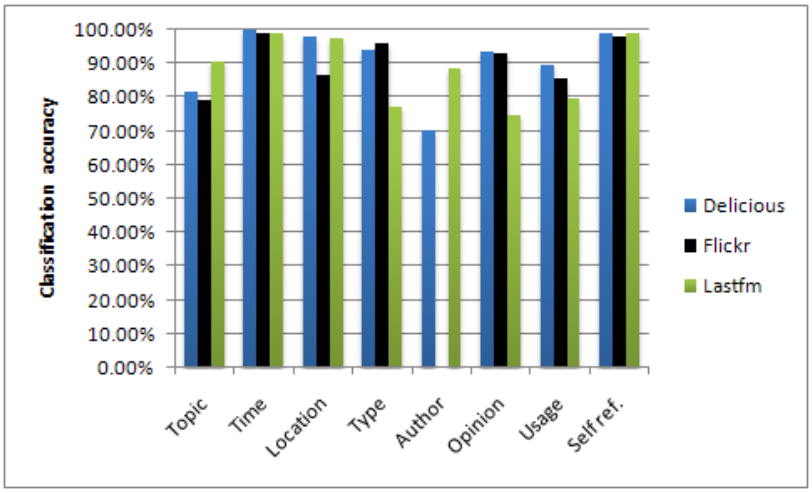

Figure 3.9 Classification accuracy per class and systems

Performance of Rule-based Methods

The regular expressions and table look-ups performed very well in predicting the five categories: Time, Location, Type, Author/Owner and Self-reference. With about 98% accuracy, performance was especially satisfactory for the highly standardized Time tags as well as for Self reference tags. However, accuracy is considerably lower for Type in *Last.fm*. This is mainly due to the used lists not being exhaustive enough. For example, the list of genres did not contain all potential sub-genres, newly emerging mixed styles or simply spelling variants and abbreviations. Its quota decreased progressively with the "difficulty" of the data set, *i.e.* the less frequent and more idiosyncratic the tags became. As a solution to this general problem, extending the list by adding tags similar to the initial set – based on second order co-occurrence – is planned.

Similarly, our artist database did not contain all naming variants for a band or a singer and it had wrong entries in the artist's rubric. Loosening matching cri-

teria, on the other hand, results in predicting a much larger proportion of tags to denote Author/Owner than in the ground truth. For the heuristic employed to find Author/Owner tags in *Del.icio.us* the same argument holds. While the regular expressions-based method just found a portion of the tags of interest, more rules *e.g.* including named entity recognition would probably lead to many false positives.

Last but not least, system specific design choices influence heavily the accuracy (of 86.70%) for spotting locations in *Flickr*. Due to the rule that does not allow users to enter space characters in the tags, many compound names could not be recognized by the method: "deadsea", "newyorkcity", "sanfrancisco", "seattlepubliclibrary". At the same time, some location names may range over multiple tags (*e.g.* "new" and "york").

Performance of Model-based Methods

The C4.5 decision tree yielded extremely good results for tag classification into Topic, Opinion and Usage tags. From the different intrinsic and extrinsic tag attributes used as features, part of speech and the semantic category in WordNet were present for all best performing classifiers, except for Topic in *Last.fm*. Here, number of users and number of words and characters alone achieved the best results. The number of words and characters obviously helped identifying Opinion tags in all three system, as well as Usage tags in *Flickr*. However, as a consequence of the relatively small training set of 700 tags as well as highly unbalanced 'natural' distribution of tags over the three categories, robustness needs to be improved. For *Del.icio.us* and *Flickr* the rate of false negatives is very high for Opinion and Usage tags. Thus, none (for *Del.icio.us*) or only part of the true Opinion and Usage tags are found.

In contrast, almost all true Topic tags are correctly identified, but at the same time the number of predicted Topic tags overestimates the real proportion in the ground truth for *Del.icio.us* and *Flickr*. The opposite happens for *Last.fm*. The classifiers learn well to reject non-Topic and non-Usage tags, but they also miss more than half of the true positives. Thus, our classifiers reinforce the tendencies to focus on one particular tag type depending on the system.

Nevertheless, the average accuracy is good, lying between 82% and 88%. As shown in Table 3.3 and Figure 3.10, except for Opinion in *Del.icio.us* and Topic in *Last.fm*, the machine learning algorithms perform well in predicting tag type shares per system correctly. For example, the Opinion classifier matches 18.43% of the tags in *Last.fm*, compared to 17.71% by human rating. *Last.fm*– being a free-for-all tagging system – exhibits a significantly higher amount of subjective tags than the other systems. It seems that users, especially youngsters looking for a way of expressing themselves, enjoy labeling the songs with Opinion tags, like "addicting", "guilty pleasures", "chick music", or "songs that totaly rule", *etc.*

Word Sense Disambiguation

Extracting similar tags by computing second order co-occurrence, *i.e.* calculating cosine similarity between two tags based on their co-occurring tags, and exploiting them during learning, improves classification performance on average by only 2% for *Last.fm*, while there is no noticeable difference for *Del.icio.us*. Although some meaningful disambiguations can be performed using this method, it does not have a big influence on the classification accuracy. Some positive examples for similar tags for *Del.icio.us* capturing synonyms, translations in other languages or simply singular/plural variations would be: "gasonline" and "cheap.gas", "flats" and "Home.Rental", "Daily.News" and "noticias" or "technique" and "techniques". For *Last.fm*, we could find pairs like "relaxing" and "calm", "so beautiful" and "feelgood tracks", "favorite tracks" and "favs" or "brit rock" and "british rock". However, quite some of the similar tags found seem not to be synonymous, in particular for the less popular, idiosyncratic tags. Still, for disambiguation, the strategy proved successful as (almost) synonymous and strongly related words usually explain the meaning of a word. For example, "rock" or "pop", two ambiguous tags with most popular senses other than musical were correctly disambiguated and the musical meaning was chosen in the case of *Last.fm*.

Overall Results

The linear average of all accuracies is 89.93%, while a more meaningful average, weighted by the real (*i.e.* manual) percentages of tags for each class, is 83.32%. The weighted average values per system are of: *Del.icio.us* - 83.93%, *Flickr* - 85.07%, *Last.fm* - 81.08%.

As initially shown in [BFNP08] for a smaller sample, tag class distributions vary significantly across the different systems. We observe that vocabulary and tag distribution depend on the resource domain, *e.g.* images and Web pages can refer to any topic, whereas music tracks are more restricted in content, thus leading to a more restraint and focused set of top tags. The most numerous category for *Del.icio.us* and *Flickr* is Topic, while for *Last.fm* Type is predominant, followed by Opinion. A portion of tags could not be classified with reasonably confidence, the percentage for the "Unknown" tag type varying between 12% and 27%. Our methods overestimate the occurrences of Topic tags for *Del.icio.us* at the expense of Opinion tags. Similary, not all Type and Author tags could be identified succesfully in *Last.fm*. However, apart from this, our methods predict comparable class shares as the human raters in the overall distribution (Figure 3.10).

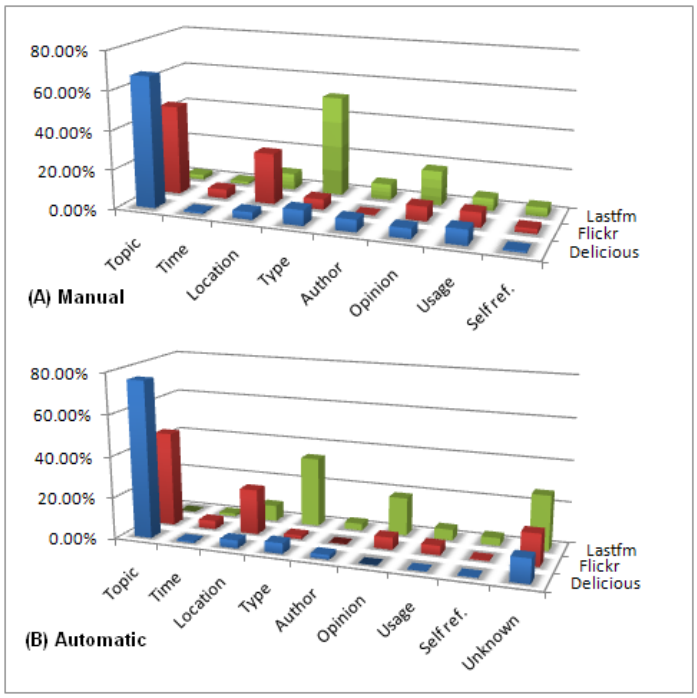

Figure 3.10 Tag distribution per tagging systems: (A) manual assignment, (B) automatic assignment

3.7 Discussion

Tag usage is rapidly increasing on the Web, providing potentially interesting information to improve search. To tap that potential, we extended previous preliminary work with a thorough analysis of the use of tags for different collections and in different environments. For the three popular and quite different collaborative tagging systems *Del.icio.us*, *Flickr* and *Last.fm*, as well as for Anchor Text (*AT*), we investigated the kinds of tags used, their distribution, and their suitability for improving search.

Our analysis provided evidence for the usefulness of a common tag classification scheme for different collections, thus allowing us to compare the kinds of tags used in different tagging environments. We have shown that the distributions of tag types strongly depend on the resources they annotate: For *Flickr*, *Del.icio.us* and *AT* *Topic*-related tags are appearing in more than 50% of the cases, while for *Last.fm* the *Type* category is the most prominent one.

Other interesting findings refer to the added value of tags to existing content: More than 50% of existing tags bring new information to the resources they annotate and for the music domain, this is the case for 98.5% of the tags. A large amount of tags is accurate and reliable, for the music domain for example 73.01% of the tags also occur in online music reviews.

Regarding search, our studies show that most of the tags can be used for search,

and that in most cases tagging behavior exhibits approximately the same characteristics as searching behavior. We also observed some noteworthy differences: For the music domain, *Usage context* is very useful for search, yet underrepresented in the tagging material. Similar, for pictures and music *Opinions/Qualities* queries occur quite often, although people tend to neglect this category for tagging. Clearly, supporting and motivating tags within these categories could provide additional information valuable for search.

These observations motivated us to develop methods for automatically classifying tags into the eight categories building our tag taxonomy. We introduced two types of methods for achieving this goal – rule-based, relying on regular expressions and predefined lists, as well as model-based methods, employing machine learning techniques. Experimental results of an evaluation against a ground truth of 2,100 manually classified sample tags show that our methods can identify tag types with 80-90% accuracy on average, thus enabling further improvement of systems exploiting social tags.

Chapter 4

Tags Supporting Personalization

4.1 Introduction

We have seen in the previous chapter some characteristic features of tags and tagging systems and how tags can be successfully used to support many different kinds of searches. In the present chapter we will investigate the use of tags for personalization, *i.e.* we will focus on solving *Problem 2* from Chapter 1.

As the amount of available data has grown at an unprecedented scale, the *information overhead* problem of users being flooded with far too much information, is becoming more and more critical. The need for powerful personalization techniques, which help focusing the retrieved results to match the user-specific retrieval context, is thus also turning crucial.

Even if the content available online has diversified both in terms of sources of provenance and in terms of nature – the share of multimedia content (*e.g.* pictures, videos, music, *etc.*) is rapidly increasing – the default way of searching is still using textual queries. In the cases of "traditional" media[1], *e.g.* articles or news, keyword queries represent a natural means to locate information. However, when attempting to locate information in form of multimedia items, still text-based queries are employed. This situation occurs mainly because of three reasons: (1) due to its simplicity and easiness of use, simple keyword-based search has been established as the default way of searching for the majority of users; (2) there are no search engines supporting descriptive queries; and (3) currently, no publicly available multimedia search engines support queries-by-example. Despite the simplicity, keyword-based search suffers from the ambiguity of the words building the queries. For example, a user interested in music and issuing the query *"alternative"* has to browse through

[1]With "traditional" media we refer to online content available in electronic textual form.

a long list of results containing among music-related hits also documents about "alternative energy", "alternative medicine" or "alternative transportation". This kind of situations occur on a daily basis and the need for personalization techniques is tremendous. With powerful personalization algorithms in place, the user would be first provided with the suitable set of results matching both her topic profile and the purpose of her search. Moreover, it is quite often the case (especially when interacting with the "new" media – music, videos, blogs, *etc.*) that users do not have a particular query in mind, but rather want to browse the available content or even more convenient, receive recommendations tailored to their interests.

User studies presented by Search Engine Watch [Sul04] indicate that actually 81% of the survey's participants would prefer to receive personalized content, but only around 64% of the interviewed users are willing to provide insight into their preferences for products of content. However, practicaly this number is even much lower, as most users do not want to spend much time on completing registration forms or stating their topics of interests, or do not even have the knowledge to accurately specify them. Imagine for example a user who is interested in "electronic" music, but has already listened over and over again all music files that she has on her PC. If she decides to use an online music recommender system, first she will most likely have to complete a registration form with personal details and some preferred music artists. Only after this step is completed, the user is able to listen to music and most of the times, first recommended items are rather poor in matching her tastes. In order to improve the quality of the recommendations, users usually need to explicitly provide feedback to the system in the form of ratings for the music tracks they listen (both positive and negative rates). Nevertheless, this is a very time consuming process and many users abandon frustrated this kind of systems.

The algorithms presented in this chapter aim to solve this type of problems and are based on tag information. Two types of tags will be considered: (1) collaboratively created tags produced by Web 2.0 users and (2) annotations assigned by human experts and gathered from publicly available taxonomies, such as the Open Directory Project (ODP).

For the first proposed algorithms we use Web 2.0 tags extracted from a music portal and we discuss the benefit of tags for producing personalized music recommendations. In this context, we show how to build tag-based user profiles, as opposed to traditional track-based profiles and demonstrate how to use the tag-based profiles to recommend personalized songs. We define several new recommender and search algorithms, and investigate their behavior, comparing it to classical collaborative filtering based on track-based profiles as a base-line. The search-based methods are the best performing ones, in addition being also much faster than collaborative filtering and not suffering from the cold start problem. Moreover, these methods can be applied not only in the case of recommendation scenarios, but also for personalizing users' music search results.

The second set of proposed algorithms aims at personalizing Web ranking and

relies on expert annotations extracted from the ODP taxonomy. We show how to generalize personalized search inside catalogs such as ODP, but not limited to it, beyond current available search features which offer the possibility to restrict searches to specific categories. The precision of our personalization method significantly surpassed Google's in a set of experiments on topic-related searches. Since it can be argued that the effort made by human experts in manually classifying Web pages into the ODP predefined categories is infinitesimal compared to the size of the Web, we also propose methods to automatically extend these categories to the whole Web. We build on the idea that sets of ODP or other directory entries can be used to bias PageRank appropriately, and thus to implicitly extend such annotations to the rest of the Web. We specifically investigate when biasing on such a set actually makes a difference to non-biased PageRank, presenting experiments with various kinds of biasing sets (*i.e.*, including different kinds of entries). We then use these results to analyze biasing sets from the ODP 2001 crawl used in [Hav02] and show that all biasing sets we investigated (up to four levels deep) can be successfully used for biasing.

The rest of the chapter is structured as follows: after Section 4.2, where we discuss related work in the context of personalized ranking and recommendations, we continue in Section 4.3 presenting a set of seven algorithms (Section 4.3.3) for providing personalized music songs to the users. We next present a set of experiments focusing on the evaluation of the proposed algorithms and discuss the results in Section 4.3.4. Section 4.4 follows and is structured into two parts: one discussing about the use of ODP tags for personalizing Web ranking (Section 4.4.1) and one proposing a method to automatically extend the ODP tags to the whole Web (Section 4.4.2). We conclude this chapter with a discussion on the results and contributions introduced here (Section 4.5).

4.2 Specific Background

Personalization has attracted a lot of attention during the last few years and several interesting approaches have been proposed. In this section we will review the most relevant ones and we will analyze existing work from the two domains which we also target to improve with the algorithms proposed in this chapter, namely personalized (music) recommendations and personalized Web ranking.

4.2.1 Personalized Recommendations

The interest in the area of recommender systems is immense from the perspective of both industry and academia, mainly because of the abundance of practical applications in this domain. Examples of such applications include recommending books, CDs/DVDs, music, movies, *etc.* However, despite of the work that has already been done to serve some of these application scenarios, the current generation of recom-

mender systems still requires further improvements to make recommendation methods more effective and applicable to an even broader range of real-life applications. These improvements include better methods for representing user behavior and information about the items to be recommended, more advanced recommendation modeling methods, incorporation of various contextual information into the recommendation process and last but not least development of less intrusive methods.

The recommendation problem is usually reduced to the problem of estimating ratings for the items that have not yet been seen by the users and intuitively this estimation is most of the times done based on the users' previously rated items. Once the estimates are computed, the items with the highest estimated ratings can be recommended to the users.

In the literature, recommendation algorithms can be basically classified into three categories:

- *Content-based algorithms* – the user will receive recommendations based on the similarity of the recommended items with his previously rated content.
- *Collaborative filtering algorithms* – the user will be recommended items that people with similar tastes and preferences liked in the past.
- *Hybrid algorithms* – combinations of content-based and collaborative filtering methods.

Even if in practice most algorithms proved to perform well, they all suffer from a number of limitations. For example content-based techniques are known to suffer from the limited content analysis, overspecialization and new user problems. Basing their recommendations on content analysis, the content-based algorithms are limited by the features that are explicitly associated with the objects they use as input. Therefore, in order to have sufficient features, it must be ensured that the content is in a form that can allow automatic parsing (*e.g.* textual form) or that the features will be assigned manually. While information retrieval techniques perform quite well in extracting features from textual content, some other domains have serious problems with automatic feature extraction. The multimedia domain, is a very good example – recommending music, movies or pictures is much more challenging, since using only basic information such as title, author, performing artists, *etc.* often produces unsatisfactory results. The algorithms we propose in this chapter aim to solve exactly this situation, providing valuable personalized music recommendations.

Beside the content analysis problem, overspecialization also represents a drawback of the content-based recommendation methods. Because this kind of systems recommend those items which are most similar with the users' previously highly rated items, they run into the danger of boring their users with too similar content, instead of providing them also with some content diversity. In some certain situations, the content should not even be recommended if it is too similar to the users' items (*e.g.*

when reading a news article, the user would most likely not like to get the same information, just written a bit differently).

Like content-based methods, collaborative filtering algorithms have their own problems, *i.e.* new users and new items problems, as well as sparsity. Providing recommendations to new users is difficult, mainly because the system must first learn the users' preferences from the ratings that they provide. Acquiring ratings into the system, obviously requires time and multiple interactions of the users with the system. Similarly, new items cannot be recommended until they have been rated by a substantial number of users. Last but not least, it is known that most users are not very keen on explicitly providing feedback on the received recommendations, unless they are very bad or very good [AT05], therefore collaborative filtering-based systems also suffer from data sparsity. This means that the number of available ratings is much smaller than the number of ratings that need to be predicted.

Some non-intrusive methods of getting user feedback are presented in [MSDR04] and [SKR01]. [SKR01] suggests a taxonomy of several well known e-commerce recommendation applications, based on the type of input required from the users, the ways the recommendations are presented to the users, the technology used to produce the recommendations and the degree of personalization of the recommendations. Unfortunately the paper does not discuss any evaluation metrics of the applications considered. [MSDR04] proposes two experimental recommender systems, Quickstep and Foxtrot, for recommending academic research papers. Quickstep uses ontology inference and an underlying ontology (ODP related to computer science) to improve the accuracy of user profiling. Foxtrot represents an extension of Quickstep with visualization features enabling users to provide explicit feedback on their profiles. Experimental evaluation of the systems reveals a lot of space for improvement, the accuracy of both created user profiles and of the received recommendations being around 50%. Obviously these approaches are still inaccurate and cannot yet fully replace the explicit ratings provided by the users. In our approach we make use of the user created tags, which are provided totally voluntarily and thus reveal direct interest of the users in the corresponding topics.

Regarding the music recommendation domain that we also target to improve with our methods, most of the currently available music recommender systems are based on collaborative filtering methods, *i.e.* they recommend music to a user by considering some other users' rating for the same music pieces. This technique is quite widely utilized, including music shopping services like Amazon[2] or iTunes[3], and has proved to be effective. However, this recommendation method suffers from the cold start problem, is not very scalable or often offer poor variety of recommendation results [LKGO04].

A better technique is described in [CRH05], which gives an overview of the Foafing the Music system, which uses the Friend-of-a-Friend (FOAF) and Rich Site Summary

[2]Amazon. http://www.amazon.com
[3]iTunes. http://www.apple.com/itunes

(RSS) vocabularies for recommending music to a user, depending on her musical tastes. Music information, such as new album releases, related news about artists or available audio pieces, is gathered from RSS feeds from the Web, whereas FOAF documents are used to define user preferences, *i.e.* for building the user profiles. The approach we propose differs from [CRH05] by the fact that the user profile is inferred automatically from his desktop music data without any additional manual effort from the user's side. Another hybrid music recommendation method is presented in [YGK+06], which simultaneously considers user ratings and content similarity and is based on a three-way aspect model, so that it can directly represent substantial (unobservable) user preferences as a set of latent variables introduced in a Bayesian network. Then, probabilistic relations over users, ratings and contents are statistically estimated.

A totally different approach for producing music recommendations is presented in [PVV06]. Their method is applied to an interactive music system that generates playlists fitting the preferences indicated by the user. For automatically generating music playlists, their approach uses a local search procedure in the solution space, based on simulated annealing: the algorithm iteratively searches the solution space stepping from one solution to a neighboring solution, compares their quality and stops when a globally optimal solution is found. [CZZM07] also proposes a search-based method for producing music recommendations and is therefore similar with our approach. The authors point out the fact that a search-based method as a recommendation strategy is definitely much better in terms of scalability, compared to collaborative filtering techniques, or content-based methods. In the solution proposed in [CZZM07], a music piece is first transformed into a sequence of music signature representing timbral characteristics of the music piece. Based on this, an LSH-based method is proposed for indexing the music songs for enabling later efficient similarity search. In the recommendation phase, the representative signature sequences are extracted for some seed songs and are used to create a query, which is then used to retrieve the pieces with the most similar melodies from the indexed dataset. As relevance criteria, matching ratio, temporal order, term weight and matching confidence are considered. Even if the method seems to provide satisfactory results, the quality is still below Pandora's [Pan].

[WL02] describes a system that queries web search engines for pages related to artists, downloads the pages, extracting the text and natural language features, and analyzes these features in order to produce textual summary descriptions of each artist. These descriptions are then used to compute similarity between artists and can be further used for producing recommendations. However, the paper does not present any evaluation experiments regarding the quality of the recommendations received by using this technique. Additionally our approach differs from the one presented in [WL02] by the fact that they are searching the Web for finding similar artists, whereas we search the Web (in particular the *Last.fm* site) for building up a user profile. [CF00] is also similar to [WL02], in that it collects Web data with

the aid of spiders and uses this data as input for music collaborative filtering. The evaluation of the method shows that data collected by this spider can be nearly as effective for collaborative filtering as data collected from real users.

A similar approach to ours, which uses collaboratively created data from the Web for making recommendations is described in [BWC07]. However, their goal is generating personalized tag recommendations for users of social bookmarking sites such as *Del.icio.us*. Techniques for recommending tags do already exist and are based on the popularity of tags among all users, on time usage information, or on simple heuristics to extract keywords from the URL being tagged. Their approach complements these techniques and is based on recommending tags from URLs that are similar to the one in question, according to two variants of cosine similarity metrics.

4.2.2 Personalized Ranking

Personalizing Web search is a promising way to improve search quality by customizing search results for users with individual goals and the interest in the area brought the participation of both research and industry. Many existing papers addressing personalized Web ranking focus on the different possibilities of building user profiles and are less concerned about how to actually use these profiles to improve search results. A detailed study about user profiling and possible sources of information used for constructing them (*e.g.* items bookmarked, time spent visiting some resource, visiting frequency *etc.*) is presented in [Cha99]. However, in this section we will discuss existing work addressing the more interesting problem related to integrating the profile information into the ranking framework. Since many of the methods proposed to achieve personalized Web ranking and which we will also refer to, rely on PageRank [Bri98, PBMW98], we will first shortly review the algorithm.

PageRank review

PageRank is a link analysis-based algorithm used by Google[4], that assigns a numerical weighting to each element of a hyperlinked environment, such as the World Wide Web, with the purpose of measuring its relative importance within the set. The basic idea is that if a page u has a link to another page v, then the author of the page u is implicitly conferring some importance to page v. A hyperlink to a page thus counts as a vote of support. The PageRank of a page is defined recursively and depends on the number of all pages that link to it ("incoming links"), as well as on the PageRank of the pages that link to it. A page that is linked to by many pages with high PageRank receives a high rank itself. If there are no links to a web page there is no support for that page.

We consider N_u the number of outlinks of page u (outdegree of u) and $Rank(u)$, as the rank of page u. Then the link (u, v) transfers $Rank(u)/N_u$ importance to page

[4]Google. http://www.google.com

v. This simple idea leads to the following fix-point computation that yields the rank vector $\vec{Rank}^*$ over all of the pages on the Web. If N represents the total number of pages on the Web, we assign to each of them an initial rank value equal to $1/N$. If B_v represents the set of pages pointing to v, then the resulting rank of page v is computed iteratively as follows:

$$\forall v, Rank_{i+1}(v) = \sum_{u \in B_v} Rank_i(u)/N_u \tag{4.1}$$

The computation is done iteratively until the rank converges within some threshold. The rank of every page in the set is computed similarly, based on the Equation 4.1, and in the end, the vector $\vec{Rank}^*$ contains the corresponding rank values of all pages. Even if the computation of $\vec{Rank}^*$ is quite expensive, this is done only once offline, after the crawling step has finished. Later, the vector needs to be recomputed, as a number of new pages gets included in the initial set. The values from the vector $\vec{Rank}^*$ can be used to influence the ranking of the search results.

The same process can be also expressed as an eigenvalue computation for the Web matrix. Let M be the square, stochastic matrix corresponding to the directed graph G of the Web, assuming all nodes in G have at least one outgoing edge. If there is a link from page j to page i, then let the matrix entry m_{ij} have the value $1/N_j$, otherwise the value is 0. One iteration of the previous fix-point computation corresponds to the matrix-vector multiplication $M \times \vec{Rank}$. Repeatedly multiplying $\vec{Rank}$ by M yields the eigenvector $\vec{Rank}^*$ of the matrix M. This also means that $\vec{Rank}^*$ is the solution of the equation:

$$\vec{Rank} = M \times \vec{Rank} \tag{4.2}$$

Because M corresponds to the stochastic transition matrix over the graph G, PageRank can be viewed as the stationary probability distribution over pages induced by a random walk on the Web.

The convergence of PageRank is guaranteed only if M is irreducible (every node in G can be reached from any other node, *i.e.* G is strongly connected) and aperiodic (any node in G can be revisited in a multiple of $k = 1$ steps). The latter is guaranteed in practice for the Web, while the former is true if we add a damping factor $1 - \alpha$ to the rank propagation.

We define a new matrix M', in which we add transition edges of probability $\frac{\alpha}{N}$ between every pair of nodes in G:

$$M' = (1 - \alpha)M + \alpha[\frac{1}{N}]_{N \times N} \tag{4.3}$$

This modification improves the quality of PageRank by introducing a decay factor $1 - \alpha$ which limits the effect of rank sinks (Web pages with no outlinks), in addition to guaranteeing convergence to a unique rank vector. Substituting M' for M in Equation 4.2 we can express PageRank as the solution to:

$$\vec{Rank} = M' \times \vec{Rank} \tag{4.4}$$

$$\vec{Rank} = (1-\alpha)M \times \vec{Rank} + \alpha\vec{p} \tag{4.5}$$

with $\vec{p} = [\frac{1}{N}]_{N\times 1}$

PageRank-based personalization approaches

One of the earliest mentions of personalizing PageRank-based ranking can be found in [PBMW98]. Here the authors suggest the use of the $\vec{p}$ vector for personalizing the ranking computation. Intuitively this vector corresponds to a random surfer periodically jumping to a random page in the Web. This is however a very democratic choice for $\vec{p}$, since all Web pages are valued simply because they exist. Instead, if we want to personalize PageRank, we can include in the $\vec{p}$ vector only those pages corresponding to the interests of a particular user, such that the random jump is thus strongly biased toward this set of pages. The idea was however never fully explored. [RD02] extends this model, by using a more intelligent surfer who probabilistically hops from one page to another, depending on the content of the pages and the query terms the surfer is looking for. However, with this approach PageRank is tailored based on the query terms and not on individual users. In [AN04], the authors suggest a method for personalizing PageRank based on URL features such as Internet domains. The intuition behind this approach is that users might favor pages from a specific geographic region, as well as pages that are likely to be monitored by experts for accuracy and quality, such as pages published by academic institutions. The users need thus to specify their interest profiles as binary feature vectors where a feature corresponds to a DNS tree node or node set. The PageRank scores are pre-computed for each profile vector by assigning a weight to each URL based on the match between the URL and the profile features. A weighted PageRank vector is then computed based on URL weights and used at query time to rank results.

[Hav02] also explores the idea of transforming the original PageRank algorithm such that it becomes topic sensitive, and thus also avoids the problem of getting as search results highly ranked pages just because they are highly linked, but with only little relevance to the topic of the queries [DS99]. The authors of [Hav02] propose to compute a set of PageRank vectors, biased on a set of representative topics, such that these vectors more accurately capture the notion of importance with respect to a particular topic. They compute 16 such biased PageRank vector, each of them corresponding to one of top-level categories found inside the Open Directory Project (ODP) catalog. Then, at query time, they calculate the similarity of the query (and of the context of the query) to each of these topics. Afterwards, instead of using a single global ranking vector, the linear combination of the topic-sensitive vectors is taken, weighted based on the similarities of the query with each of the 16 topic vectors. Evaluation results of this approach showed that personalized PageRank scores can improve Web search, however the paper does not address any scalability issues or ways to improve it.

[JW03] proposes a method, which can achieve personalized rankings, without

having to store a Personalized PageRank Vector (PPV) for each user. With this approach the user needs to first select from a set of hub pages (H) a number of pages that are representative for her interests. As opposed to [Kle99], where hub pages represent Web pages that have many outlinks to high authoritative pages, here hub pages refer to Web pages having high PageRank scores. Once the user has selected her preferred set of pages, her resulting PPV-vector can be expressed as a linear combination of basis hub vectors associated with the preference vectors with a single non-zero entry corresponding to each of the pages selected by the user from the hub set – basis vectors. Computing and storing all possible hub vectors is however impractical. To solve this problem, the authors suggest to decompose the computation of the basis vectors into partial vectors and hubs skeletons. Intuitively, partial vectors encode the part of the corresponding hub vector which is unique to the hub page and is calculated at query time. The hub skeleton complements the partial vectors' information, in which it captures the interrelationships among hubs – common information to all hub pages and pre-computed offline. Computing partial vectors and hub skeletons, thus avoids a lot of redundant computation and storage.

The idea of efficiently computing PageRank is also investigated in [KHMG03]. Here the authors suggest to speed up the computation of PageRank by splitting the Web graph into blocks based on domain information. A Local PageRank vector is computed for each resulting block and then the relative importance of each of the blocks is estimated (*i.e. BlockRank*). After these steps, the Local PageRank of the blocks are weighted by the estimated BlockRanks and then concatenated to form a Global PageRank vector. The Global PageRank vector is then used as a starting vector for the computation of the standard PageRank. Personalization using this algorithm can be achieved by the fact that the BlockRank vectors will be biased towards the users' preferences. In this particular case, instead of being able to specify which page has higher chances to be revisited, the users can specify the host that is likely to be revisited during a random surf on the Web.

In [QC06], Qiu & Cho study the problem of automatically inferring user profiles from search engines' click-through data. The paper also discusses how to integrate the inferred user profiles to produce personalized rankings. The proposed approach aims to use for this purpose the ranking mechanism introduced by Haveliwala in [Hav02]. Thus, the personalized ranking for a page p, given query q issued by a user with topic preference vector T is calculated as a linear combination of: (1) the personalization factors based on the user's preferences; (2) of the probability distributions of the query terms given the preferred topics of the user; and, last but not least (3) of the topic-sensitive PageRank values of the page p given the interest topics of the user. The conducted user study, revealed up to 24% improvement of search results of the personalization algorithm over standard PageRank-based results.

Other personalization approaches

Researchers have also proposed ways to personalize Web search based on ideas other than PageRank. For example, [TM02] extends the well know HITS algorithm by artificially increasing the hub and authority scores of the pages marked relevant by the users as results of previous searches. [LYM04] suggest to filter out from the result list, those items that are known to be irrelevant or very likely to be irrelevant. The process of filtering is separated from the actual ranking mechanism and in the paper, the authors propose to restrict searches to a set of categories defined in ODP via Google Directory. However, the main contribution refers to the different ways to exploit the users' browsing history in order to learn their profiles.

Other approaches to personalized Web ranking focus on developing personalized ranking methods which are applied on a restricted set of Web pages, namely those pages returned by a particular search engine for a specific query. The returned pages are scored according to the user's personal preferences and the resulted scores are combined with the scores returned by the search engine for the same set of pages. In [DSW07] the authors use exactly this approach to achieve personalized Web ranking. The proposed personalization methods are categorized into two classes: person- and group-level methods, depending on where the information used for personalization originates. Five personalization strategies are evaluated in the paper: two click-based and three profile-based. However, the biggest achievement of the paper resides in the evaluation framework which allows a large-scale evaluation of the five algorithms. Here, the authors use a snapshot of MSN query logs, which is used as a baseline for comparing the ranking provided by MSN with the personalized ranking produced by the five methods introduced in this paper. Since for MSN the initial relevance scores of the items cannot be reproduced, the final ranking of the results, including the personalized score, is made based on Borda's ranking fusion method [DKNS01]. This approach is similar to our taxonomy-based personalized search algorithm, introduced in Section 4.4.

Other more recent approaches to personalized search refer to the family of methods relying on social information. [XBF+08] for example proposes a personalized search framework to utilize folksonomy for personalized search. In their framework, the rank of a Web page is decided not only by the term matching the query and the Web page's content, but also by the topic matching between the user's interests and the Web page's topics. A major contribution of the paper is represented by the methodology used for automatic evaluation of the proposed approach. Here, the evaluation is performed on *Del.icio.us* and *Dogear*[5] and relies on the assumption that users' bookmarking and tagging actions reflect their personal relevance judgments. Also depending on social data is the approach presented by Noll and Meinel

[5]Dogear is a project of IBM Watson Research Center, which allows people to bookmark pages in their intranet. http://domino.watson.ibm.com/cambridge/research.nsf/242252765710c19485256979004d289c/1c181ee5fbcf59fb852570fc0052ad75?OpenDocument

in [NM07]. Here the authors suggest also to re-rank the non-personalized search results by considering the users' social annotations and the collaborative annotations attached to the search results. However, in this case the evaluation of the method is performed on a much smaller scale, through a user survey.

4.3 Using Tags for Personalized Music Recommendations

More and more companies start offering personalized services toward their users and online music recommender systems are one prominent example. Pandora[6], *Last.fm*[7], Foafing the Music[8] or Yahoo! Music[9] are a few of the currently online available music recommendation systems. These systems employ different approaches for recommending music tracks to their users, ranging from content based and collaborative filtering techniques to hybrid methods. While clearly useful to their users, these more conventional recommendation techniques still suffer from a number of problems: in the case of collaborative filtering[10], musical pieces with no ratings cannot be recommended because recommendations are based on actual user ratings. Besides, artist variety in recommended pieces can be poor, making these recommendations less satisfactory than they could be. Recommending tracks that are similar to users' favorites in terms of content induces unreliability in modeling users' preferences and besides, content similarity does not necessarily reflect preferences. Hybrid recommendation techniques combine the advantages of the two approaches and are thus better. However, to our knowledge, the only usable music recommender system using hybrid techniques is Foafing the Music, which relies heavily on FOAF profiles created by the users - not an easy task for non-expert users. Last, but not least, though many of these community sites allow tagging, these tags are not used for recommendation or any form of advanced search.

For overcoming these problems, we propose novel methods relying on tag-based music user profiles and on tag-based ranking of music tracks in order to produce a set of recommended music items. To evaluate the performance of our approach, we compare our results with recommendations given by state-of-the-art collaborative filtering methods.

[6]Pandora. http://www.pandora.com
[7]Last.fm. http://www.last.fm
[8]Foafing the Music. http://foafing-the-music.iua.upf.edu
[9]Yahoo! Music. http://music.yahoo.com
[10]Method of making automatic predictions about the interests of a user by collecting preference information from many users.

4.3.1 Datasets

For this study we used *Last.fm* data, crawled in May 2007. The characteristics of this dataset have been already introduced in Section 3.3, though the focus was set on tracks, tags and other features related directly to these two entities. In addition to the entities already presented in the description of this dataset included in the above mentioned section, we also collected information related to users. As a place to congregate and share musical tastes, *Last.fm* gives to its users access to thousands of tracks from all musical genres. Before using the system, users need to create an account and specify their preferred music genre. The interaction with the service can be done through the web interface, via the embedded Flash player, or through the *Last.fm* player which the users need to download and install locally on the desktop. By listening to tracks and rating them (there are three options: *Love this track*, *Skip this track* or *Ban this track*), user profiles are created. Based on these profiles, the service produces a number of personalized features, such as finding out about artists that the user likes, other people with similar tastes, appropriate charts, or events in the neighborhood. Based on the music the user likes, *Last.fm* connects her to other users that have similar tastes (her neighbors). However, for using this feature, users need to listen to at least 5 artists and wait for about one week before their neighbors appear. Every week, the list of neighbors is updated, based on what the user has been listening to during that week.

We will not repeat the details related to the collected tags (21,177) and tracks (317,058), but instead present the details of the crawled set of users. We extracted details about 289,654 registered users on *Last.fm*. For these users the extracted information includes user ID, gender, age, location, registration date, listened number of tracks, list of friends and neighbors, overall top listened tracks and list of used tags. From this number of users, we filtered out all users who have not yet listened to at least 50 tracks and who have not used at least 10 different tags, and were thus left with 12,193 unique users for our experiments.

4.3.2 Tag-based Profiles vs. Track-based Profiles

Although most music recommendation methods use track-based algorithms to present the user with new interesting tracks, given the increasing tendency toward tagging all types of multimedia files on the Internet we wanted to investigate how well tag-based methods can perform. We also wanted to avoid extensive manual ranking, so opted to construct user profiles from locally stored MP3 files. This usually works very well, as most users have quite a few music files in MP3 format on their desktop, usually much more than needed for a music recommender system to provide satisfactory results.

We will distinguish between profiles created for *Last.fm* users and for Non-*Last.fm* users. They differ in terms of the source of information which is used for building up the profiles: for *Last.fm* users the starting point is represented by their web pages on

portal, whereas in the case of the Non-*Last.fm* users we start from the information available on their desktops. For this type of users we extract metadata about each track existing on the desktop, and match artist and track name (extracted from filename or ID3 tag) against the *Last.fm* music database. This provides all data necessary to create comprehensive user profiles that accurately reflect users' music preferences. We will use the following notations:

$ITF(TG)$ = *Inverse Tag Frequency for Tag TG*

$p(TR, U)$ = *Preference of User U for Track TR*

$p(TG, U)$ = *Preference of User U for Tag TG*

$TR\_U\_Listened$ = *Number of times User U has listened to Track TR*

$TR\_Overall\_Listened$ = *Number of times Track TR was overall listened on* Last.fm

$TG\_UsedFor\_TR$ = *Number of times Track TR was tagged with Tag TG by all users*

$TG\_Used\_Overall$ = *Number of times Tag TG was used overall*

TG_UsedBy_U = *Number of times User U has used Tag TG*

$Tracks\_Containing\_TG$ = *Number of tracks on the User's Desktop that were tagged with Tag*

We distinguish between *Last.fm* and Non-*Last.fm* users using either *Last.fm* or *Non-Last.fm* indices in the corresponding formulas.

Track-based Profiles

Track-based profiles are defined as collections of music tracks with associated preference scores, describing users' musical tastes, as follows:

$Profile\_Tracks(U) = \{< TR_i, P_i > \mid TR_i = \text{user's track}, P_i = p(TR_i, U)\}$

Track-based profiles for *Last.fm* users.

In the case of *Last.fm* users the profiles are inferred from the users' web pages available on the *Last.fm* site. Their collection of tracks includes all tracks the users have been listening to inside the system. Their associated scores are a function of the number of times users have listened to these music tracks. The algorithm for creating the track-based profiles for this type of users is described below:

Alg. 4.3.2.1: Track-based profile for *Last.fm* user

1: For each track TR in user's tracks list UTR
2: **Compute track's score** P
3: **Add** pair $< TR, P >$ to user profile
4: Return user's track-based profile

with $P = p(TR, U_{Last.fm}) = log(TR\_U_{Last.fm}\_Listened)$

Track-based profiles for Non-*Last.fm* users

For Non-*Last.fm* users the only available source of personal information is represented by their desktops. We first extract explicit metadata such as artist and track name either from the filename or from the ID3 tags (if any) of the music files existing on the desktop. This information is then matched against the *Last.fm* database and only tracks with a Lucene[11] TFxIDF[12] score above 0.9 are kept for further processing. This pre-processing step is described below:

Alg. 4.3.2.2: Get list of tracks

1: For each track (MP3) on user desktop
2: **Extract** artist name and track name from filename as *S1*
3: **Extract** artist name and track name from ID3 tag as *S2*
4: **Combine** *S1* and *S2* into *S*
5: **Search** with *S* on *Last.fm* index
6: **Retrieve** tracks *LT* matching query with Lucene TFxIDF score >0.9
7: **Add** tracks *LT* to the user's list of tracks *UTR*
8: Return *UTR*

Once the list of tracks for a Non-*Last.fm* user is created, the track-profile is realized in a similar manner as for a *Last.fm* user: algorithm 4.3.2.2 is applied on the list of tracks with the only difference that the preference scores for the tracks are now a function of the overall number of times tracks have been listened on *Last.fm*.

Alg. 4.3.2.3: Track-based profile for Non-*Last.fm* user

1: Create list of tracks *UTR* applying **Alg. 4.3.2.1**
2: Apply Alg. 4.3.2.2 on list of tracks *UTR*

with $P = p(TR, U_{Non-Last.fm}) = log(TR\_Overall\_Listened)$

Tag-based Profiles

Tag-based user profiles are defined as collections of tags together with corresponding scores representing the user's interest in each of these tags. The formal definition is given below:

$Profile\_Tags(U) = \{< TG_i, P_i > \mid TG_i = \text{user's tag}, P_i = p(TG_i, U)\}$

[11] Lucene. http://lucene.apache.org

[12] The TFxIDF score (term frequency-inverse document frequency) is a weight often used in information retrieval and text mining for evaluating how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus.

Again, we distinguish between profiles created for *Last.fm* and Non-*Last.fm* users. For this type of profiles, the list of tags can be extracted either from the users' list of tracks (tags which have been used to tag the tracks) or directly from the tags the users have used themselves.

Tag-based profiles for *Last.fm* users

For *Last.fm* users, the first type of tag-based profiles can be created starting from the list of tracks the users have been listening to on *Last.fm*. For each of these tracks, we extract the list of all tags which have been used inside the system for annotating them. In this case, the preference associated to a tag is proportional to the number of times these tracks tagged as TG were listened by the user and to the number times this tag has been used by all users on *Last.fm* to tag those tracks. The description of the algorithm is given below:

Alg. 4.3.2.4: Track-Tag-based Profile for *Last.fm* User

1: For each track TR in user's track list UTR
2: **Extract** list of used tags TTG for TR
3: **Add** TTG to user's tag list UTG
4: For each tag TG in UTG
5: **Compute** tag's score P
6: **Add** pair $< TG, P >$ to user's profile
7: Return user's tag-based profile

with $P = p(TG, U_{Last.fm}) = [ITF(TG)] \cdot log \sum_i (log(TR_i\_U_{Last.fm}\_Listened) \cdot log(TG\_UsedFor\_TR_i))$

$$ITF(TG) = log \frac{\sum_i TG_i\_Used\_Overall}{TG\_Used\_Overall}$$

Similar to the IDF value in Information Retrieval, in order to reduce the influence of tags which are very popular among users, but might not accurately reflect user's personal musical taste, we introduce an optional parameter in the preference formula, ITF. The formula penalizes tags which appear very often and boosts the preference for tags appearing more rarely.

The second possibility for creating the tag-based profiles for *Last.fm* users is to directly take the tags which the users have already used (information which can be found on their web pages) together with their frequency. The algorithm in this case looks as follows:

Alg. 4.3.2.5: Tag-based Profile for *Last.fm* User

1: For each tag TG in user's tag list UTG
2: **Compute** tag's score P
3: **Add** pair $< TG, P >$ to user's profile
4: Return user's tag-based profile

with $P = p(TG, U_{Last.fm}) = log(TG\_UsedBy\_U_{Last.fm})$

For this case, we do not need to introduce the ITF parameter in the preference formula, since now the profile is already very personalized – the user has directly used these tags.

Tag-based profiles for Non-*Last.fm* users

For Non-*Last.fm* users, the collection of tags building up their profiles is inferred based on the list of tracks the users have on their desktops. This list of tracks is compiled as presented in Alg. 4.3.2.2 and then transformed using Alg. 4.3.2.4. Preference scores for the tags are now computed as follows: the score depends on the number of times these tracks which are part of the users' profile and are tagged as TG have been listened by all *Last.fm* users and to the number of times this tag has been used by all users on *Last.fm*. Again in the formula we have the optional parameter ITF used to decrease the bias toward very popular tags. Moreover, for Non-*Last.fm* users we keep in the profile only the top 100 preferred tags, after evaluating recommendation results with several such values for this algorithm ranging from 10 to 500.

Alg. 4.3.2.6: Track-Tag-based Profile for Non-*Last.fm* User

1: Create list of tracks UTR applying **Alg. 4.3.2.2**
2: Apply Alg. 4.3.2.4 on list of tracks UTR
3: Retain in the profile **top 100** preferred tags
4: Return user's tag-based profile

with $P = p(TG, U_{Non-Last.fm}) = [ITF(TG)] \cdot log \sum_i (log(TR_i\_Overall\_Listened) \cdot log(TG\_UsedFor\_TR_i))$

$$ITF(TG) = log \frac{\sum_i TG_i\_Used\_Overall}{TG\_Used\_Overall}$$

The second variant of the tag-based profile corresponding to a Non-*Last.fm* user looks similar to the previous one. In this case the preference depends on the number of tracks on the user's desktop that are tagged with tag TG.

Alg. 4.3.2.7: Tag-based Profile for Non-*Last.fm* User

1: Create list of tracks UTR applying **Alg. 4.3.2.2**
2: Apply Alg. 4.3.2.4 on list of tracks UTR
3: Retain in the profile **top 29** preferred tags
4: Return user's tag-based profile

with $P = p(TG, U_{Non-Last.fm}) = log(Tracks\_Containing\_TG)$

Since Non-*Last.fm* users did not use the tags building up their profile by themselves — they are just inferred based on the music tracks they have on their desktop — we chose to simulate the *Last.fm* profiles by maintaining only the top 29 preferred tags. From the data we have crawled we could see that the average number of used tags among the users is 29, therefore we keep in the tag-profiles we create with algorithm 4.3.2.7 only top-29 most preferred tags.

4.3.3 Algorithms

Having seen how tags can be used to build music user profiles, we next discuss how these profiles can help rank music tracks subject to be recommended to users. We present algorithms relying on both track- and tag-based profiles for comparative reasons. In total, we describe 7 algorithms which, based on the type of profile and the technique we used for producing music recommendations, can be grouped into three categories: Collaborative Filtering based on Tracks, Collaborative Filtering based on Tags and Search based on Tags.

Track-based Recommendations

CF based on TRacks (CFTR). Traditional music recommender systems use User-Item Collaborative Filtering methods with music tracks as items. This method is successfully used in *Last.fm* and other systems, though we still have the cold start problem, *i.e.* users have to listen (and possibly rank) a minimum number of tracks and tracks have to be ranked by some listeners, before recommendations are possible. We will use such an algorithm as baseline to compare our other algorithms against.

Alg. 4.3.3.1: Collaborative Filtering based on Tracks (CFTR)
Track Profile ↪ Track Recommendation ↪ Tracks

1: Create users **track-based profile** (Alg. 4.3.2.1/ Alg. 4.3.2.2)
2: **Get track recommendations** based on the Taste-Recommender Java library:
3: Compute **top 10 most similar users** SU with current user U
based on cosine similarity between track profiles
4: **For each** similar user SU_i
5: Get tracks TR_j with preference $p(TR_j, SU_i)$
6: **Combine** lists TR_j of tracks into $\Rightarrow RTR$
7: **Recommend** music tracks RTR

Tag-Based Recommendations

For the three algorithms proposed in this paper as tag-based recommendation algorithms, the matrix on which we apply collaborative filtering is a User-Tag matrix. In this matrix, line i corresponds to the tag-profile of user i and contains corresponding preference scores for tags which have been used by the user (and 0 for the other tags). In these algorithms what we obtain as result of applying CF on the User-Tag matrix is of course a list of recommended tags, based on what tags other similar users have used. What we want to achieve are music recommendations and not tag recommendations. Therefore with this list of tags we search which tracks have been tagged with most of these tags, taking into account their associated preference scores. We return the top 10 matching tracks, scored by cosine similarity, as recommended songs.

CF based on Track-Tags with ITF (CFTTI). The first algorithm we propose in this category uses tag-based profiles which have been extracted from the list of tracks users have been listening to (Alg. 4.3.2.4/ 4.3.2.6). For not biasing profiles toward highly used tags, when computing preference scores associated to the tags we also include the ITF parameter. The recommended list of tags obtained after applying CF on the User-Tag matrix is then used for getting the music recommendations:

Alg. 4.3.3.2: CF based on Track-Tags with ITF (CFTTI)
Tracks ↪ Tag Profile ↪ Tag Recommendation ↪ Search w/ Tags ↪ Tracks

1: Create **tag-based profiles** (Alg. 4.3.2.4/ Alg. 4.3.2.6 both with ITF)
2: **Get tag recommendations** based on the Taste-Recommender Java library:
3: Compute **top 10 most similar users** SU with current user U based on cosine similarity between tag profiles
4: **For each** similar user SU_i
5: Get top 50 tags TG_j by preference $p(TG_j, SU_i)$
6: **Combine** lists TG_j of tags into $\Rightarrow RTG$
7: **Create** Query Q
8: **For** each tag TG_i in RTG
9: **Add** pair $< TG_i, p(TG_i, U) >$ to Q
10: **Search** with Q tracks being tagged with tags in $Q \Rightarrow RTR$
11: **Compute** cosine similarity between tracks in the Lucene index and Q
12: **Rank** resulted tracks RTR based on cosine similarity
13: **Recommend** music tracks RTR

CF based on Track-Tags No-ITF (CFTTN). This second algorithm differs from CFTTI by computing the tag-based profiles without the ITF parameter in the formula corresponding to tags' preference. Otherwise the steps in the algorithm are the same as in Alg. 4.3.3.2.

CF based on Tags (CFTG). For the third algorithm the user profiles on which the tag recommendation step is based, are more personal – users have already used those tags. In this case, line **1:** in Alg 4.3.3.2 is modified and the algorithm looks as follows:

Alg. 4.3.3.3: CF based on Tags (CFTG)
Tags ↪ Tag Profile ↪ Tag Recommendation ↪ Search with Tags ↪ Tracks

1: Create **tag-based profiles** (Alg. 4.3.2.5 / Alg. 4.3.2.7)
2: **Get tag recommendations** based on the Taste-Recommender Java library:
3: Compute **top 10 most similar users** SU with current user U based on cosine similarity between tag profiles
4: **For each** similar user SU_i
5: Get top 50 tags TG_j by preference $p(TG_j, SU_i)$
6: **Combine** lists TG_j of tags into $\Rightarrow RTG$
7: **Create** Query Q
8: **For** each tag TG_i in RTG
9: **Add** pair $< TG_i, p(TG_i, U) >$ to Q
10: **Search** with Q tracks being tagged with tags in $Q \Rightarrow RTR$
11: **Compute** cosine similarity between tracks in the Lucene index and Q
12: **Rank** resulted tracks RTR based on cosine similarity

13: Recommend music tracks RTR

Tag-Based Search

In our last set of algorithms, we use the tags extracted through the previously presented methods for direct matching with other tracks. This is done by creating a disjunctive query of clauses where each clause consists of a tag and its preference. The results are tracks ordered by cosine similarity between the vector of tags they have been tagged with and the vector of tags given in the query. Direct search using tags has the big advantage of being much faster than any collaborative filtering algorithm, the results being produced instantly. It also offers the user the possibility to enter keyword queries based on tags and get new tracks from different domains if wanted.

Search based on Track-Tags with ITF (STTI). Similar to CFTTI this algorithm is based on profiles created using the algorithms 4.3.2.4 and 4.3.2.6 and includes the ITF factor in the preference formula:

Alg. 4.3.3.4: Search based on Track-Tags with ITF (STTI)
Tags ↪ Tag Profile ↪ Search with Tags ↪ Tracks

1: Create **tag-based profiles** (Alg. 4.3.2.4/ Alg. 4.3.2.6 both with ITF)
2: Create Query Q
3: **For** each tag TG_i in the profile of current user U
4: **Add** pair $< TG_i, p(TG_i, U) >$ to Q
5: Search with Q tracks being tagged with tags in $Q \Rightarrow RTR$
6: **Compute** cosine similarity between tracks in the Lucene index and Q
7: Rank resulted tracks RTR based on cosine similarity
8: Recommend music tracks RTR

Search based on Track-Tags No-ITF (STTN). The second search-based algorithm is based on 4.3.3.4, just that we remove the ITF parameter in the preference formula.

Search based on Tags (STG). Like the CFTG algorithm, STG uses profiles created by Alg. 4.3.2.5 and 4.3.2.7. Tags contained in the profiles are then directly used for searching for tracks which have been tagged with these tags:

Alg. 4.3.3.3: Search based on Tags (STG)
Tags ↪ Tag Profile ↪ Search with Tags ↪ Tracks

1: Create **tag-based profiles** (Alg. 4.3.2.5 / Alg. 4.3.2.7)
2: Create Query Q

3: **For** each tag TG_i in the profile of current user U
4: **Add** pair $< TG_i, p(TG_i, U) >$ to Q
5: **Search** with Q tracks being tagged with tags in $Q \Rightarrow RTR$
6: **Compute** cosine similarity between tracks in the Lucene index and Q
7: **Rank** resulted tracks RTR based on cosine similarity
8: **Recommend** music tracks RTR

4.3.4 Evaluation

Experimental Setup

We evaluated our algorithms with 18 subjects (B.Sc., Ph.D., and Post- Doc students in different areas of computer science and education). They installed our desktop application to extract their user profiles as described in Alg. 4.3.2.3, 4.3.2.6, and 4.3.2.7. Then we ran all 7 variants of our algorithms (Track Collaborative Filtering $CFTR$ - baseline, Track-Tag CF $CFTTI$, $CFTTN$ - with or without Inverse Tag Frequency ITF, Tag CF $CFTG$, Track-Tag Search $STTI$, $STTN$ - with or without ITF, Tag Search STG) over their profiles. The average number of tracks in a user profile was 658, ranging from 17 up to 2,848, not being statistically significant in influencing algorithm outcome. For each of the algorithms we collected the top-10 recommended items (*i.e.*, tracks), such that each user had to rate a maximum number of 70 recommended tracks (it is possible that the same track gets recommended by more of the proposed algorithms, in which case the track was listed only once). Results were presented to the subjects in shuffled order, so that they were not aware of the algorithm which produced the result nor the score of the recommended item. For each of the recommended tracks, the users had to provide two different scores: one measuring how well the recommended track matches their music preferences ([0] - I don't like this track, [1] - I don't mind listening to this track, [2] - I like the track) and one reflecting the novelty of the track ([0] - I already know this track, [1] - I know something about this track, *e.g.* I know the artist OR I heard the track on the radio, but I do not remember the name, *etc.*, and [2] - this track is really new for me).

The quality of the recommended results was measured using the normalized version of Discounted Cumulated Gain (DCG) [JK00], a rich measure which gives more weight to highly ranked documents, while also incorporating different relevance levels by giving them different gain values:

$$DCG(i) = \begin{cases} G(1) & , if\ i = 1 \\ DCG(i-1) + G(i)/log(i) & , otherwise. \end{cases}$$

For novelty, only the average of the marks given was used, resulting in a value ranging from 0 (known) to 2 (really new).

Results

Table 4.1 shows the NDCG value, its statistical significance over the $CFTR$ baseline computed using T-tests, and the average popularity (number of times a track was listened to on *Last.fm*) of recommended tracks for each algorithm. All Collaborative Filtering algorithms based on tags ($CFTG$, $CFTTI$, $CFTTN$) performed worse than the baseline, as standard User-Item CF techniques already show high precision. All our search algorithms, though, show quite substantial improvements over track based CF (STG 12%, $STTI$ 37%, $STTN$ 44% as shown in Figure 4.1; $STTI$ and $STTN$ both highly statistically significant). This outcome is certainly positively influenced by the consistent usage of tags on *Last.fm*: Most frequently used tags denote the track's genre, so our search gets biased toward specific user preferred music genres. It was also interesting to note that the better people knew the tracks (*i.e.*, a lower novelty value), the higher they rated the recommendations. We observed an almost perfect inverse correlation between these two scores, with a Pearson's correlation coefficient between average NDCG and Novelty values per algorithm of $c = -0.987$, and still a high inverse correlation of all preference and novelty marks with $c = -0.513$.

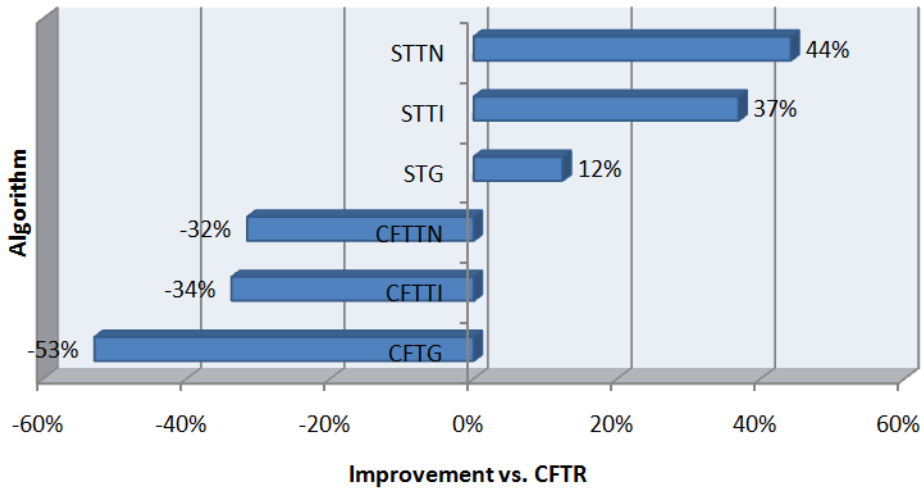

Figure 4.1 Relative NDCG gain (in %) over the $CFTR$ baseline for each algorithm.

Nr.	Algorithm	NDCG	Signif. vs. CFTR	Popularity	Novelty
1	**CFTR**	0.54	-	15,177	1.39
2	CFTG	0.25	High, $\mathbf{p \ll 0.01}$	4,065	1.83
3	CFTTI	0.36	High, $\mathbf{p \ll 0.01}$	6,632	1.72
4	CFTTN	0.37	High, $\mathbf{p \ll 0.01}$	13,671	1.74
5	**STG**	**0.60**	No, $\mathbf{p = 0.22}$	7,587	1.07
6	**STTI**	**0.73**	**High**, $\mathbf{p \ll 0.01}$	10,380	0.82
7	**STTN**	**0.77**	**High**, $\mathbf{p \ll 0.01}$	16,309	0.78

Table 4.1 Normalized Discounted Cumulative Gain over the first 10 recommended tracks, along with the average track popularity and average novelty

Another interesting result is that *STG* recommends much less popular tracks than our *CFTR* baseline, but still of higher quality, so that it is suited for people demanding a higher diversity of music, not listening to the same tracks over and over again. We can thus suggest different algorithms, depending on the user's preference concerning popularity and novelty of tracks. Because of the high use of the "rock" tag (used twice as much as any other tag), many *hard rock* or *heavy metal* songs were recommended, mostly by tag-based CF algorithms. Further research has to be done in order to disambiguate tag meanings, and to reduce unwanted tag weights.

Nr.	Algorithm	NDCG	Signif. vs. CFTR	Popularity	Novelty
1	**CFTR**	0.60	-	19,717	1.33
2	CFTG	0.29	Yes, $\mathbf{p = 0.02}$	7,787	1.84
3	CFTTI	0.33	Yes, $\mathbf{p = 0.02}$	9,970	1.79
4	CFTTN	0.32	High, $\mathbf{p \ll 0.01}$	25,576	1.77
5	STG	0.55	No, $\mathbf{p = 0.29}$	7,799	1.11
6	**STTI**	**0.76**	Minimal, $\mathbf{p = 0.10}$	10,709	0.81
7	**STTN**	**0.80**	Minimal, $\mathbf{p = 0.07}$	15,664	0.61

Table 4.2 Normalized Discounted Cumulative Gain, average track popularity, and average novelty over the first 10 recommended tracks, only for users with less than 50 tracks on their Desktop

Nr.	Algorithm	NDCG	Signif. vs. CFTR	Popularity
1	**CFTR**	0.31	-	22,766
2	CFTG	0.19	No, $\mathbf{p = 0.21}$	4,852
3	CFTTI	0.24	No, $\mathbf{p = 0.42}$	5,758
4	CFTTN	0.29	Minimal, $\mathbf{p = 0.13}$	17,419
5	STG	0.27	No, $\mathbf{p = 0.25}$	21,111
6	STTI	0.26	No, $\mathbf{p = 0.36}$	29,196
7	**STTN**	**0.35**	No, $\mathbf{p = 0.25}$	44,490

Table 4.3 Normalized Discounted Cumulative Gain and average track popularity over the first 10 recommended tracks, only for items with high novelty

When looking only at people with less than 50 personal music tracks on their desktop (Table 4.2) - this was the case for 7 of our test subjects, the number of tracks ranging from 17 to 48 and averaging at 31 - we still find a gain of 26% and 33% over the baseline for *STTI* and *STTN*, respectively. This indicates that our user tag profiles also work with less rich music repositories. Results presented in Table 4.3 only for recommended tracks with high novelty (*i.e.*, novelty mark = 2), show a decrease in NDCG for all algorithms, mostly not statistically significant since users had different music knowledge. Still *STTN* performs 13% better than *CFTR*, mainly because it recommends tracks with higher popularity.

4.4 Using Tags for Personalized Web Ranking

In the previous section we have seen how tags can be used to improve ranking of multimedia content, in particular music, subject to be recommended to users. We

continue presenting another use of tags and we propose an approach for achieving improved personalized Web ranking. In this context, with tags we will refer to the manually assigned topic categories from the Open Directory Project (ODP)[13]. Unlike a usual tagging system, where users can freely assign tags to almost any content, ODP relies on a number of volunteer human editors (82,293) who manually assign Web pages to one or more of the 590,000 existing categories. These categories are organized into an hierarchical structure with 16 top level classes and comprise over 4,5 million Web sites. Although the discussions and the algorithms presented in this section refer to ODP data, any other similar taxonomy can be used instead. Moreover, any social bookmarking system (*e.g. Del.icio.us*, *Digg*[14], *StumbleUpon*[15]) can be used as substitute, as the tags employed inside such platforms can be used to automatically classify Web pages [HJSS06a] into a similar hierarchical structure to ODP. Thus far, ODP is one of the largest efforts to manually annotate Web pages, exporting all this metadata information to RDF format. One good use of these tags is to personalize search, *i.e.*, return search results which are both relevant to the user profile, as well as of good quality. Still, given the fact that Google now indexes more than 8 billion pages[16], the ODP effort only covers around 0.05% of the Web pages indexed by Google. The question that we will try to answer in this section is thus: "Does search using these ODP tags stand any chance against Google in providing better and personalized results?".

We will propose two ways of personalized search: First, using ODP entries directly, we show how to generalize personalized search in catalogs such as ODP and Google Directory beyond the currently available search restricted to specific categories. Second, extending the manual ODP classifications from its current 4 million entries to a 8 billion Web in an automated way is feasible, based on an analysis of how topic classifications for a small but important subset of a large page collection can be extended to this large collection via topic-sensitive biasing of PageRank values [PBMW98]. This generalizes earlier approaches which already investigated topic-sensitive PageRank, but relied on very simple classifications using only 16 topics.

4.4.1 Using ODP Tags for Personalized Search

Even though several approaches to personalize Web search exist already, as we saw in Section 4.2.2, they are still far from being perfect: in [PBMW98], we need to run the entire algorithm for each preference set (or biasing set), which is practically impossible in a large-scale system; at the other end, [Hav02] computes biased PageRank vectors limited only to the broad 16 ODP top-level categories because of the same problem;

[13]ODP. http://www.dmoz.org
[14]Digg. http://www.digg.com
[15]StumbleUpon. http://www.stumbleupon.com
[16]8 billion indexed pages was the information provided on the Google site at the time when we conducted this study (2005). Current official information on the Google blog (http://googleblog.blogspot.com/2008/07/we-knew-web-was-big.html) indicates more than 1 trillion indexed pages.

[JW03] improves this somewhat, allowing the algorithm to bias on any subset of a given set of pages (H). Although work has been done in the direction of improving the quality of this latter set [CON04], one limitation is still that the preference set is restricted to a subset of this given set H (if $H = \{CNN, FOXNews\}$ we cannot bias on MSNBC for example). Moreover, the bigger H is, the more time is needed to run the algorithm. Thus finding a simpler and faster algorithm with at least similar personalization granularity is a worthy goal to pursue.

Both ODP and Google Directory[17] offer some rudimentary ways for "personalized search": ODP by restricting search to the entries of just one of the 16 main categories, Google by offering to restrict search to a specific category or subcategory. We will try to improve this personalized search feature by taking the user profile into account in a more sophisticated way, and investigate how such an enhanced personalized search on the ODP or Google entries compare to ordinary Google results.

Algorithm

As already mentioned, the algorithm we propose for personalizing Web search requires as input user profiles. For defining the profiles each user needs to select several topic-tags from ODP, which best fit her interests, such as:

/Arts/Architecture/Experimental
/Arts/Architecture/Famous_Names
/Arts/Photography/Techniques_and_Styles

Having the user profiles at hand, in the form of ODP subcategories representing the user's topics of interest, at run-time, the output given by a search service (from Google, ODP Search, *etc.*) is re-sorted using a calculated *distance* from the user profile to each output URL. This step-by-step process is depicted in Algorithm 4.4.1.1.

Algorithm 4.4.1.1. Personalized Search.

Input: $Prof_u$: Profile for user u, given as a vector of topics
Q : Query to be answered by the algorithm.
Output: Res_u: Vector of URLs, sorted after user u's preferences

---

1: Send Q to a search engine S (*e.g.*, Google)
2: Res_u = Vector of URLs, as returned by S
3: **For** $i = 1$ **to Size**(Res_u)
$Dist[i] = Distance(Res_u[i], Prof_u)$
4: **Sort** Res_u using $Dist$ as comparator

When searching on Open Directory, each resulting URL comes with an associated ODP topic tag. Similarly, many of the URLs output by Google are connected to one

[17] Google directory is build on top of ODP. http://directory.google.com

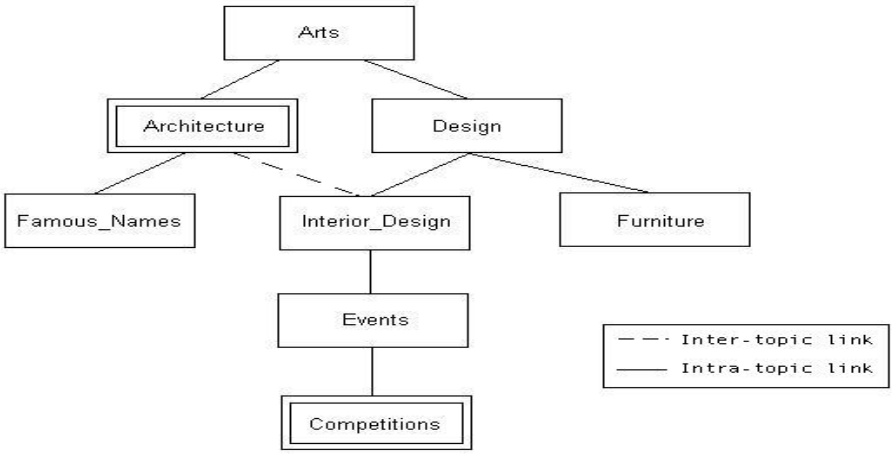

Figure 4.2 Example tree structure of topics from ODP

or more topics within the Google Directory (almost 50% of top-100 returned results, as our analyses indicated). Therefore, in both cases, for each output URL we are dealing with two sets of nodes from the topic tree: (1) Those representing the user profile (set A), and (2) those associated with the URL (set B). The distance between these sets can then be defined as the minimum distance between all pairs of nodes given by the Cartesian product $A \times B$. We additionally need a function to estimate the distance between a URL and the topics corresponding to a user profile. There are quite a few possibilities to define the distance between two nodes and below we present some.

Distance Metrics

Naïve Distances. The simplest solution is the minimum tree-distance, which, given two nodes a and b, returns the sum of the minimum number of tree edges between a and the subsumer (the deepest node common to both a and b) plus the minimum number of tree edges between b and the subsumer (*i.e.*, the shortest path between a and b). On the example from Figure 4.2, the distance between */Arts/Architecture* and */Arts/Design/Interior_Design/Events/Competitions* is 5, and the subsumer is */Arts*.

If we also consider the inter-topic links from the Open Directory, the simplest distance becomes the graph shortest path between a and b. For example, if there is a link between *Interior_Design* and *Architecture* in Figure 4.2, then the distance between *Competitions* and *Architecture* is 3. This solution implies to load either the entire topic graph or all the inter-topic links into memory. Furthermore, its utility is subjective from user to user: the existence of a link between *Architecture* and *Interior_Design* does not always imply that a famous architect (one level below in the tree) is very close to the area of interior design. Given these considerations, we will consider only the intra-topic links directly connected to a and b and output the

shortest path between them.

Complex Distances. The main drawback of the above metrics comes from the fact that they ignore the depth of the subsumer. The bigger this depth is, the more related are the nodes (*i.e.*, the concepts represented by them). This problem is solved by [LBM03], who investigates ten intuitive strategies for measuring semantic similarity between words using hierarchical semantic knowledge bases such as WordNet [Mil95]. Each of them was evaluated experimentally on a group of testers, the best one having a 0.9015 correlation with the human judgment and the following formula:

$$S(a,b) = e^{-\alpha \cdot l} \cdot \frac{e^{\beta \cdot h} - e^{-\beta \cdot h}}{e^{\beta \cdot h} + e^{-\beta \cdot h}} \tag{4.6}$$

The parameters are as follows: α and β were defined as 0.2 and 0.6 respectively, h is the tree-depth of the subsumer, and l is the semantic path length between the two words. Considering we have several words attached to each concept and sub-concept, then l is 0 if all words are in the same concept, 1 if they are in different concepts, but the two concepts have at least one common word, or the tree shortest path if the words are in different concepts which do not contain common words.

Although this measure is very good for words, it is not perfect when we apply it to the Open Directory topical tag tree because it does not make a difference between the distance from a (the profile node) to the subsumer, and the distance from b (the output URL) to the subsumer. Consider node a to be */Top/Games* and b to be */Top/Computers/Hardware/Components/Processors/x86*. A teenager interested in computer games (level 2 in the ODP tree) could be very satisfied receiving a page about new processors (level 6 in the tree) which might increase his gaming quality. On the other hand, the opposite scenario (profile on level 6 and output URL on level 2) does not hold any more, at least not to the same extent: a processor manufacturer will generally be less interested in the games existing on the market. This leads to our following extension of the above formula:

$$S'(a,b) = ((1-\gamma) \cdot e^{-\alpha \cdot l_1}) + (\gamma \cdot e^{-\alpha \cdot l_2}) \cdot \frac{e^{\beta \cdot h} - e^{-\beta \cdot h}}{e^{\beta \cdot h} + e^{-\beta \cdot h}} \tag{4.7}$$

with l_1 being the shortest path from the profile to the subsumer, l_2 the shortest path from the URL to the subsumer, and γ a parameter in $[0, 1]$.

Combining the Distance Function with Google PageRank. Once a similarity score is computed between the topics of the URLs returned by a search engine and the topics representing the user profiles, we need to also combine these scores, with the scores produced by the search engine for these resulted URLs. If we use Google to do the search and then sort the URLs according to the Google Directory taxonomy, some high quality pages might be missed (*i.e.*, those which are top ranked, but which are not in the directory). In order to avoid this situation, the above formula needs to be combined with the Google PageRank scores and we propose the following approach:

$$S''(a,b) = \delta \cdot \frac{1}{1 + S'(a,b)} + (1 - \delta) \cdot PageRank(b) \tag{4.8}$$

δ is another parameter in $[0, 1]$ which allows us to keep the final score $S''(a, b)$ also inside $[0, 1]$ (for normalized PageRank scores). Finally, if a page is not in the directory, we take $S'(a, b)$ to be ∞.

Experimental Results

To evaluate the benefits of our personalization algorithm, we interviewed 17 of our colleagues (researchers in different computer science areas, psychologists, pedagogues and designers), asking each of them to define a user profile according to the Open Directory topics, as well as to choose three queries of the following types:

- One *clear* query, which they knew to have one or maximum two meanings
- One *relatively ambiguous* query, which they knew to have two or three meanings
- One *ambiguous* query, which they knew to have at least three meanings, preferably more

We then compared test results using the following four types of Web search:

1. "Pure" Open Directory Search (**ODPS**)
2. *Personalized Open Directory Search* (**P-ODPS**), using our algorithm from Section 4.4.1 to reorder the top 1000 results returned by the ODP Search
3. Google Search (**GS**), as returned by the Google API [Goo]
4. *Personalized Google Directory Search* (**P-GDS**), using our algorithm from Section 4.4.1 to reorder the top 100 URLs returned by the Google API[18], and having as input the Google Directory topics returned by the API for each resulting URL.

For each algorithm, each tester received the top 5 URLs with respect to each type of query, 15 URLs in total. All test data was shuffled, such that testers were neither aware of the algorithm, nor of the ranking of each assessed URL. We then asked the subjects to rate each URL from 1 to 5, 1 defining a very poor result with respect to their profile and expectations (*e.g.*, topic of the result, content, *etc.*) and 5 a very good one[19]. Finally, for each sub-set of 5 URLs we took the average grade

[18] We were forced to use only the top 100 URLs, because of the limitations imposed by the Google API, as well as the limited number of Google API licenses we had.

[19] This is practically a weighted P@5.

as a measure of importance attributed to that $< algorithm, querytype >$ pair. The average values for all users and for each of these pairs can be found in Table 4.4, together with the averages over all types of queries for each algorithm.

Algorithm	Query Type			Avg./Alg.
	Ambiguous	Semi-ambiguous	Clear	
ODPS	2.09	2.29	2.87	2.41
P-ODPS	3.11	3.41	3.13	3.22
GS	2.24	2.79	3.27	2.76
P-GDS	2.73	3.15	3.74	3.20

Table 4.4 Survey results for the analyzed web search approaches

We expect the "pure" ODP search (ODPS) to be significantly worse than the Google search (GS), and that is indeed the case: an average of 2.41 points for ODP versus the 2.76 average received by Google. Also predictable was the dependence of the grading on the query type. If we average the values on the three columns representing each query type, we get 2.54 points for ambiguous queries, 2.91 for semi-ambiguous ones and 3.25 for clear ones - thus, the clearer was the query, the better rated were the URLs returned.

Personalized Search using ODP (P-ODPS) is *clearly better than Google search (GS)*, regardless whether we use Open Directory or Google Directory as taxonomy. Therefore, a personalized search on a well-selected set of 4 million pages often provides better results than a non-personalized one over a 8 billion set. This a clear indicator that taxonomy-based result sorting is indeed very useful. For the ODP experiments, only our clear queries did not receive a big improvement, mainly because for some of these queries, ODP contains less than 5 URLs matching both the query and the topics expressed in the user profile.

Personalized Search using Google (P-GDS) was far better than the usual Google search. We would have expected it to be even better than the ODP-based personalized search (P-ODPS), but results were probably negatively influenced by the fact that the ODP experiments were run on 1000 results, whereas the Google Directory ones only on 100, due to the limited number of Google API licenses we had.

The grading results are summarized in Figure 4.3. Generally, we can conclude that personalization significantly increases output quality for ambiguous and semi-ambiguous queries. For clear queries, one should prefer Google to Open Directory search, but also Google Directory search to the pure Google search. Google search is still better than Open Directory search, but we provided a personalized search algorithm which *outperforms* the existing Google and Open Directory search capabilities.

Another interesting result is that 40.98% of the top 100 Google pages were also contained in the Google Directory. More specifically, for the ambiguous queries 48.35% of the top pages were in the directory, for the semi-ambiguous ones 41.35%, and for

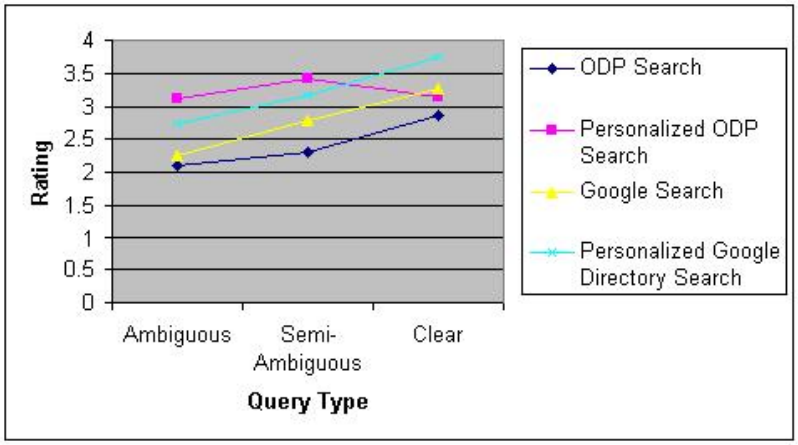

Figure 4.3 Algorithm grading for each query type

the clear ones 33.23%[20].

We also analyzed the grading for each URL and results are presented in Figure 4.4 (grades are sorted in order to produce a clearer graph). We can see Google surpassing ODP almost everywhere. The low grades for Google usually relate to ambiguous queries, where we receive pages corresponding to all the different meanings covered by the query. It is also important to remark that personalized search on Google Directory is always better than simple Google search, and similarly, personalized search on ODP is always better than simple ODP search.

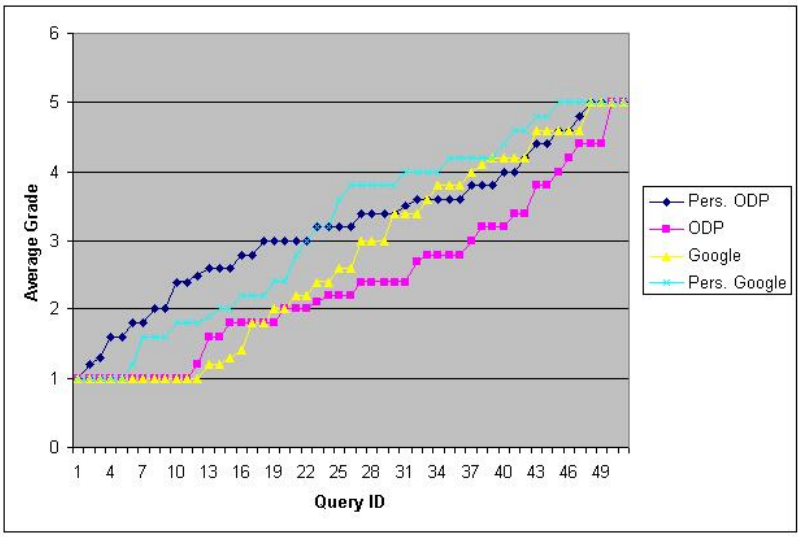

Figure 4.4 Grading behavior for all queries

[20]There were more pages for the ambiguous queries, because they were covering multiple topics.

Src. of variance	QS	Deg. of Free.	F-value [Win62]
Query Type	17.092	2	F(2,32,75%) = 2.114
Algorithm	22.813	3	F(3,48,99%) = 6.812
Inter-Relation	7.125	6	F(6,96,95%) = 2.512

Table 4.5 Survey results for the analyzed web search approaches

Finally, the statistical significance tests we performed[21] on our experiments [Bor93], provided the following findings:

- Statistical significance with an error rate below 1% for the "algorithm" criterion, *i.e.*, there is significant difference between each algorithm grading.
- An error rate below 25% for the "query type" criterion, *i.e.*, the difference between the average grades with respect to query types is less statistically significant.
- Statistical significance with an error rate below 5% for the inter-relation between query type and algorithm, *i.e.*, the results are overall statistically significant.

For a more in-depth view, the statistical analysis data is collected in Table 4.5.

4.4.2 Extending ODP Annotations to the Web

In the last section we have shown that using ODP entries and their categorization directly for personalized search turns out to be amazingly good. Can this huge annotation effort invested in the ODP project (with over 82,000 volunteers participating in building and maintaining the ODP database) be extended to the rest of the Web? This would be useful if we want to find less highly rated pages not contained in the directory. Just extending the ODP effort does not scale, because first, significantly increasing the number of volunteers seems improbable, and second, extending the selection of ODP entries to a larger percentage obviously becomes harder and less rewarding once we try to include more than just the "most important" pages for a specific topic.

We start with the following questions:

- Given that PageRank for a large collection of Web pages can be "biased" towards a smaller subset, can this be done with sets of ODP entries corresponding to given categories / subcategories as well?
- Specifically, ODP entries consist of many of the "most important" entries in a given category. Do we have enough entries for each topic such that biasing on these entries makes a difference?

[21] More specifically, we used an Analysis of Variance (ANOVA).

When does biasing make a difference?

One of the most important works investigating PageRank biasing is [Hav02]. It first uses the 16 top levels of the ODP to bias PageRank [PBMW98] on and then provides a method to combine these 16 resulting vectors into a more query-dependent ranking. But what if we would like to use one or several ODP (sub-)topics to compute a Personalized PageRank vector? More general, what if we would like to achieve such a personalization by biasing PageRank towards some generic subset of pages from the current Web crawl we have? Many authors have used such biases in their algorithms. Yet none has studied the boundaries of this personalization, the characteristics the biasing set has to exhibit in order to obtain relevant results (*i.e.*, rankings which are different enough from the non-biased PageRank). We will investigate this in the current section. Once these boundaries are defined, we will use them to evaluate (some of) the biasing sets available from ODP in Section 4.4.2.

First, let us establish a characteristic function for biasing sets, which we will use as parameter determining the effectiveness of biasing. Pages in the World Wide Web can be characterized in quite a few ways. The simplest of them is the out-degree (*i.e.*, total number of out-going links), based on the observation that if biasing is targeted to such a page, the newly achieved increase in PageRank score will be passed forward to all its out-neighbors (pages to which it points). A more sophisticated version of this measure is the hub value of pages. *Hubs* were initially defined in [Kle99] and are pages pointing to many other *high quality* pages. Reciprocally, high quality pages pointed to by many hubs are called *authorities*. There are several algorithms for calculating this measure, the most common ones being HITS [Kle99] and its more stable improvements SALSA [LM00] and Randomized HITS [NZJ01]. Yet biasing on better hub pages will have less influence on the rankings because the "vote" a page gives is propagated to its out-neighbors divided by its out-degree. Moreover, there is also an intuitive reason against this measure: PageRank biasing is usually performed to achieve some degree of personalization and people tend to prefer highly valued authorities to highly valued hubs. Therefore, a more natural measure is an authority-based one, such as the non-biased PageRank score of a page.

Even though most of the biasing sets consist of high PageRank pages, in order to make this analysis complete we have run our experiments on different choices for these sets, each of which must be tested with different sizes. For comparison to PageRank, we used two degrees of similarity between the non-biased PageRank [PBMW98] and each resulting biased vector of ranks. They are defined in [Hav02] as follows:

- **OSim** indicates the degree of overlap between the top n elements of two ranked lists τ_1 and τ_2. It is defined as

$$\frac{|Top_n(\tau_1) \cap Top_n(\tau_2)|}{n} \tag{4.9}$$

- **KSim** is a variant of Kendall's τ distance measure. Unlike OSim, it measures the *degree of agreement* between the two ranked lists. If U is the union of items in τ_1 and τ_2 and δ_1 is $U \setminus \tau_1$, then let τ_1' be the extension of τ_1 containing δ_1 appearing after all items in τ_1. Similarly, τ_2' is defined as an extension of τ_2. Using these notations, KSim is defined as follows:

$$KSim(\tau_1, \tau_2) = \frac{|(u,v) : \begin{matrix}\tau_1' \text{ and } \tau_2' \text{ agree on or-} \\ \text{der (u,v), and } u \neq v\end{matrix}|}{|U| \cdot |U-1|} \tag{4.10}$$

Even though [Hav02] used $n = 20$, we chose n to be 100, after experimenting with both values and obtaining more stable results with the latter value. A general study of different similarity measures for ranked lists can be found in [DKNS01].

Let us start by analyzing the biasing on high quality pages (*i.e.*, with a high PageRank). We consider the most common set to contain pages in the range $[0-10]\%$ of the sorted list of PageRank scores. We varied the sum of scores within this set between 0.00005% and 10% of the total sum over all pages (for simplicity, we will call this value TOT hereafter). For very small sets, the biasing produced an output only somewhat different: about 38% Kendall similarity (see Figure 4.5). The same happened for large sets, especially those above 1% of TOT. Finally, the graph makes also clear where we would get the most different rankings from the non-biased ones (in a set size from 0.003% to 0.1%)[22].

Someone could wish to bias only on the best pages (the top $[0-2]\%$, as in Figure 4.6). In this case, the above results would only move a little bit to the right on the x-axis of the graph, *i.e.*, the highest differences would be achieved for a set size from 0.02% to 0.75%. This was expectable, as all the pages in the biasing set were already top ranked, and it would therefore take a little bit more effort to produce a different output with such a set.

Another possible input set consists of randomly selected pages (Figure 4.7). Such a set most probably contains many low PageRank pages. This is why, although the biased ranks are very different for low TOT values, they start to become extremely similar (up to almost the same) after TOT exceeds 0.01% (because it would take *a lot* of low PageRank pages to accumulate a TOT value of 1% of the overall sum of scores, for example).

The extreme case is to bias *only* on low PageRank pages (Figure 4.8). In this case, the biasing set will contain too many pages even sooner, around $TOT = 0.001\%$.

The last experiment is mostly theoretical. One would expect to obtain the smallest similarities to the non-biased rankings when using a biasing set from $[2-5]\%$ (because these pages are already close to the top, and biasing on them would have

[22] Generally, if the similarity (y-axis value) is below the threshold line, then we consider the biased ranks to be relevant, *i.e.*, different enough from the non-biased ones.

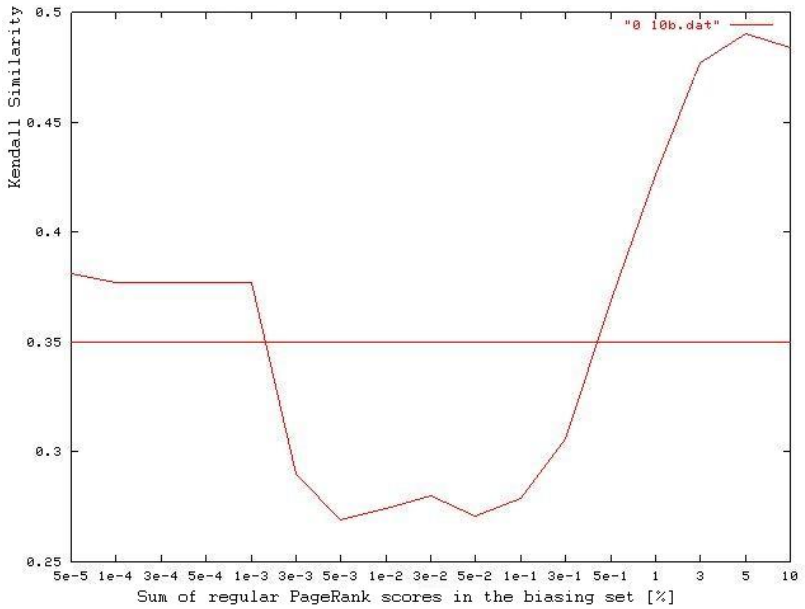

Figure 4.5 Biasing behavior for top 0 - 10% PageRank pages

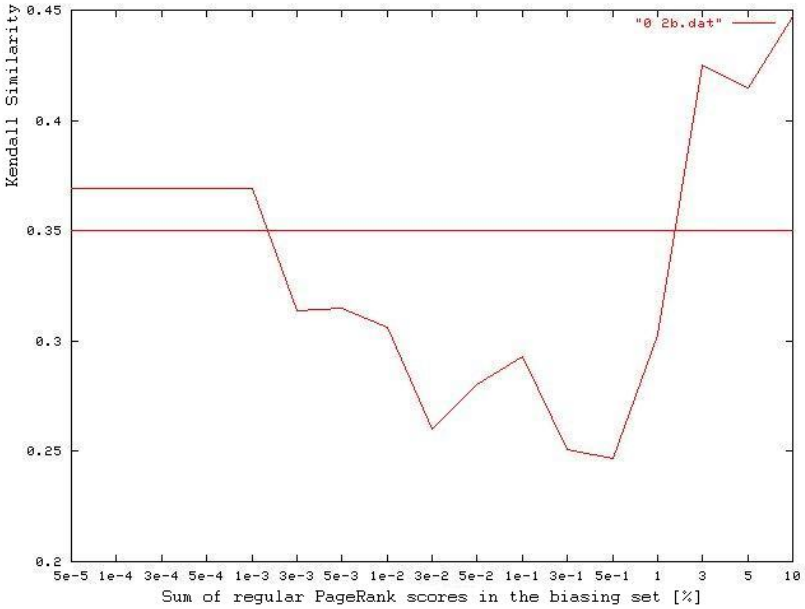

Figure 4.6 Biasing behavior for top 0 - 2% PageRank pages

best chances to overturn the list). Experimental results support this intuition (Figure 4.9), generating very different rankings for very small biasing sets and up to sets of $TOT = 0.1\%$, that is for a large scale of sizes for the biasing set.

All these presented graphs were initially generated based on a crawl of 3 million pages. Once all of them had been finalized, we selectively ran similar experiments on

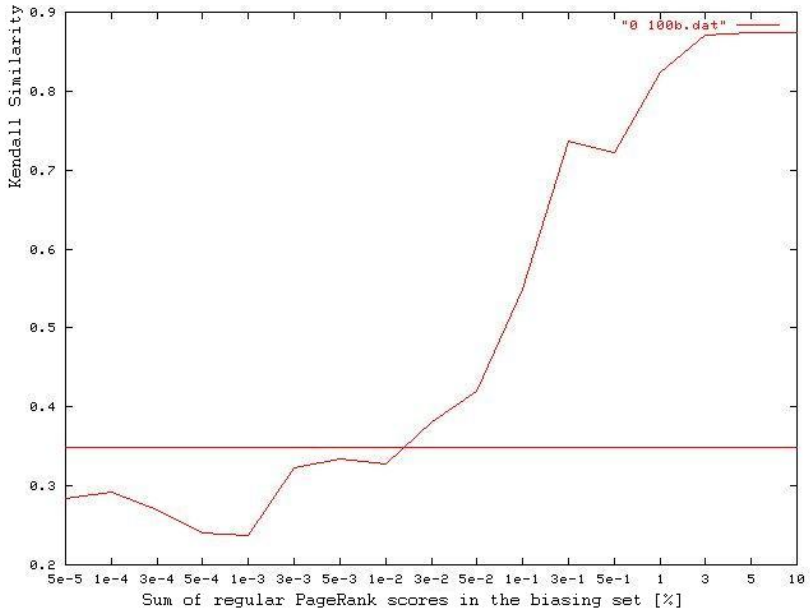

Figure 4.7 Biasing behavior for random pages

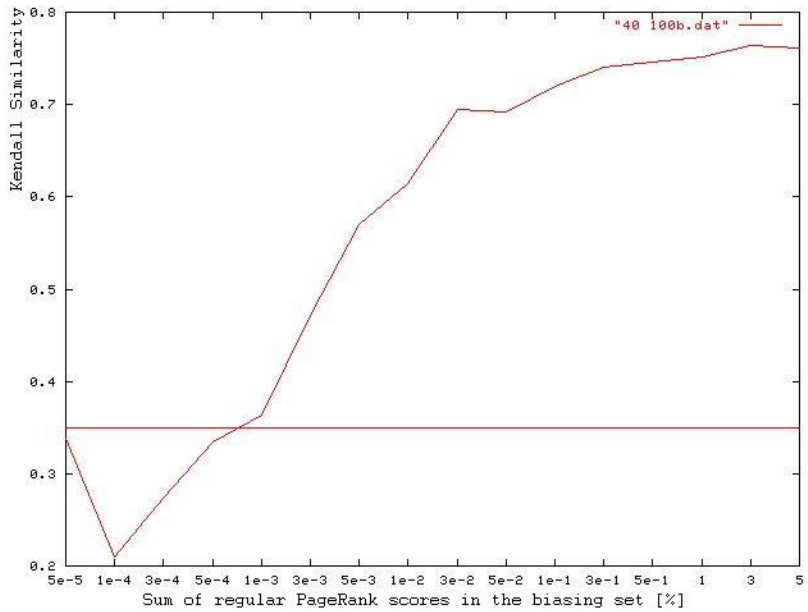

Figure 4.8 Biasing behavior for random low PageRank pages

the Stanford WebBase crawl [Sta], obtaining similar results. For example, a biasing set of size $TOT = 1\%$ containing randomly selected pages produced rankings with a 0.622% Kendall similarity to the non-biased ones, whereas a set of $TOT = 0.0005\%$ produced a similarity of only 0.137%. This was necessary in order to prove that the above discussed graphs are not influenced by the crawl size. Even so, the limits they

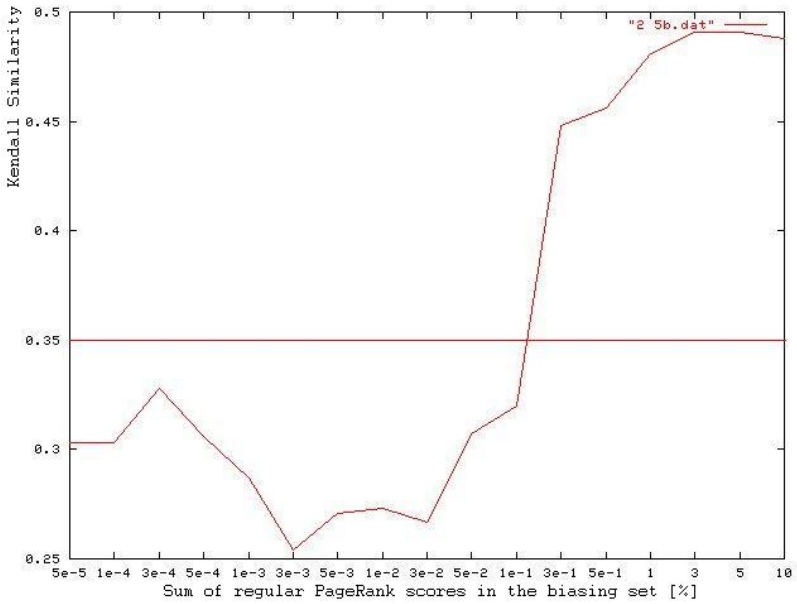

Figure 4.9 Biasing behavior for top 2 - 5% PageRank pages

establish are not totally accurate, because of the random or targeted random selection (*e.g.*, towards top $[0-2]\%$ pages) of our experimental biasing sets.

Is biasing possible in the ODP context?

The URLs collected in the Open Directory are manually added Web pages supposed to (1) cover the specific topic of the ODP tree leaf they belong to and (2) be of high quality. Both requirements are not fully satisfied. Sometimes (rarely though) the pages are not really representing the topic in which they were added. More important for PageRank biasing, they usually cover a large interval of page ranks, which made us decide for the random biasing model. However, we are aware that in this case, the human editors chose much more high quality pages than low quality ones, and thus the decisions of the analysis are susceptible to errors.

Generally, according to the random model of biasing, every set with TOT below 0.015% is good for biasing. According to this, all possible biasing sets analyzed in tables 4.6, 4.7 and 4.8 would generate a different enough PageRank vector[23].

We can therefore conclude that biasing is (most probably) possible on *all* subsets of the Stanford Open Directory crawl.

[23]Only biasing on the entire topic set of "Computers" seems to exceed this limit a little bit, but running the biased PageRank with it produced a good enough similarity - most probably because of the special structure of the ODP topic sets, as we discussed above in this section.

/Computers	*TOT* **Value**
/CAD/Mapping_and_GIS	0.000072%
/Education/Internet	0.000001%
/Internet/Consulting	0.000041%
/Internet/Bulletin_Board_Services	0.000018%
/Internet/E-mail	0.000001%
/Internet/Resources	0.000207%
/Internet/Broadcasting/Video_Shows	0.000065%
/Internet/Commercial_Services/Web_Hosting/Free/Games_Related	0.000001%
/Programming/Internet	0.000052%
/Security/Anti_Virus	0.000110%
/Companies/Product_Support	0.003163%
/Education/Hardware/HowTos_and_Tutorials	0.000198%
/Internet/Statistics_and_Demographics	0.000101%
/Internet/Cyberspace	0.000167%
/Internet/Organizations	0.000377%
/Internet/Telephony	0.000008%
/Internet/E-mail/Electronic_Postcards/Humor	0.000007%
/Programming/Games	0.000124%
/Publications/Mailing_Lists	0.000603%
/Security/Internet	0.001193%

Table 4.6 Low-level ODP biasing analysis for the Stanford ODP crawl

Topic	*TOT* **Value**	**Topic**	*TOT* **Value**
/Arts	0.01062%	/Business	0.01046%
/Computers	0.02343%	/Games	0.00297%
/Health	0.00596%	/Home	0.00528%
/Kids & Teens	0.00532%	/News	0.00707%
/Recreation	0.00541%	/Reference	0.01139%
/Regional	0.00839%	/Science	0.01314%
/Shopping	0.00296%	/Society	0.01201%
/Sports	0.00235%	/World	0.01091%

Table 4.7 Low-level ODP biasing analysis for the Stanford ODP crawl

/Computers	*TOT* **Value**	**/Computers**	*TOT* **Value**
/Algorithms	0.000072%	/Artificial_Intelligence	0.000146%
/Artificial_Life	0.000127%	/Bulletin_Board_Syst.	0.000063%
/CAD	0.000078%	/Companies	0.004042%
/Data_Comm.	0.000001%	/Data_Formats	0.000059%
/Desktop_Publishing	0.000038%	/E-Books	0.003534%
/Ethics	0.000253%	/Graphics	0.000033%
/Hacking	0.000002%	/Hardware	0.001286%
/Home_Automation	0.000001%	/HCI	0.000223%
/Internet	0.002062%	/Multimedia	0.000713%
/Organizations	0.000008%	/Parallel_Computing	0.000055%
/Programming	0.000188%	/Publications	0.000626%
/Robotics	0.000226%	/Security	0.001308%
/Software	0.007318%	/Speech_Technology	0.000008%
/Supercomputing	0.000835%	/Usenet	0.000089%
/Virtual_Reality	0.000066%	/History	0.000511%
/Education	0.000460%		

Table 4.8 Low-level ODP biasing analysis for the Stanford ODP crawl

4.5 Discussion

In this chapter we analyzed the potential of tags for supporting personalization applications (*Problem 2*). Two different aspects have been considered: (1) using tags in order to provide personalized music recommendations; and (2) using tags for achieving personalized Web ranking.

In the first part of this chapter we analyzed tag usage and statistics for one of the most popular music community sites, *Last.fm*, and compared user profiles based on these tags with conventional ones based on tracks. Using these tag-based profiles, we defined several new recommender and search algorithms, and investigated their behavior, comparing it to classical collaborative filtering based on track-based profiles as a baseline. A first set of algorithms, using collaborative filtering on tag profiles that were extracted from tracks, proved to be less successful than the baseline. A second set of tag-based search algorithms however improved results' quality significantly. In addition to a 44% increase in quality for the best algorithm, search-based methods are also much faster than collaborative filtering and do not suffer from the cold-start problem.

In the second part of this chapter, we showed how to personalize Web ranking, by relying on annotations produced by human experts and gathered from the ODP catalog. Although the discussions and the algorithms presented here refer to ODP data, any other similar taxonomy or social bookmarking system can be used instead, since the tags employed inside such platforms can be used to automatically classify Web pages into similar hierarchical structures as ODP. Given that directories like ODP contain only a very small amount of tagged pages, compared to the Google's number of indexed pages, we investigated the impact these annotations have and specifically their feasibility to implement personalized search based on these tags. We investigated two possibilities to do that, and made the following contributions:

First, using ODP entries directly, we showed how to generalize personalized search in catalogs such as ODP and Google Directory beyond the currently available search restricted to specific categories. The precision of this personalized search significantly surpassed the precision offered by unpersonalized search in a set of experiments.

Second, extending the manual ODP classifications from its current 4 million entries to a 8 billion Web is feasible, based on an analysis of how topic classifications for subsets of large page collection can be extended to this large collection via topic-sensitive biasing of PageRank values [PBMW98].

Chapter 5

Tags Supporting Knowledge Discovery

5.1 Introduction

In the previous chapters we have seen some direct applications of tags for search and personalization. Still a lot more information can be inferred from collaboratively created user tags. These semantically rich user generated annotations are especially valuable for content collections covering multimedia resources such as music, pictures or video items, where these metadata enable retrieval relying not only on content-based (low-level) features, but also on the textual descriptions represented by tags. Apart from being extremely important for multimedia retrieval (as well as for Web retrieval, like we have seen in Section 4.4), tags can also reveal some of the hidden aspects of the content they annotate, and which would be much more expensive to extract through content analysis methods. In turn, these hidden content features that are made accessible through either tag- or content-analysis, can be used to support information retrieval.

For example, in case of music resources, tags reveal a lot more information than only the music genre a track belongs to. They can for instance indicate which is the emotional state induced by listening to a particular song (*e.g.* happy, sad, lazy, aggressive, *etc.*) or which is the most suitable situation for listening some music (*e.g.* pool party, wedding, rainy day, dinner ambiance, driving, *etc.*). Tags can be even used as an indication of the potential popularity of songs, and thus help identifying high quality music content. Having the music resources indexed also according to these features, will advance considerably current possibilities of multimedia search and retrieval.

Similarly, tags attached to picture resources can reveal both simple information, such as names of persons appearing on the photos, location names, or personal impressions regarding the event where the picture has been taken, *etc.*, as well as more complex information, such as names of touristic objectives or events. These types of features can help users both for achieving better organization of content, as well as for content retrieval.

Unfortunately many of the user provided tags represent explicit information, which can be anyway relatively easily inferred from other sources, *e.g.* music genres[1] are also identifiable from ID3 tags, date and time stamp information are present also in the pictures' associated Exif metadata[2]. Still relatively few resources are tagged with keywords bearing information difficult to extract otherwise. Therefore, developing methods to automatically extend such information to resources not yet containing such tags would be highly beneficial. This additional information that can be inferred from already existing user tags can thus be used for various purposes:

- As part of an application where the extracted knowledge is presented directly to the users in the form of tag recommendations. Users can then select those that they consider relevant and add them to the content they are interacting with. Since the methods used to infer additional content information are based on tags which by nature are very heterogeneous and thus might introduce some noise in the output of the algorithms, this approach has the advantage of reinforcing the identified relevant hidden features. Users will select predominantly the relevant suggested tags and will ignore the irrelevant ones (if any), such that after some tagging cycles, the new tag features will get reinforced due to other taggers recognizing these valuable tags already employed by others and using them themselves.

- Another possibility is to index the new inferred information together with the content it is attached to, thus enriching the metadata index of a digital library, or more general, of a search engine. For the indexing process we can use either directly the output of the knowledge extraction algorithms or perform an intermediate validation step on the algorithms' results through tag recommendations (see previous item).

- Last but not least, the knowledge inferred from tags can be used to organize content along dimensions corresponding to the identified aspects, thus enabling users to more easily access their data – *e.g.* create playlists according to songs' mood or theme, organize pictures based on landmark information, *etc.*

In this chapter we will focus on inferring information from tags associated to music and picture resources. The methods we will introduce in the next sections

[1]The analysis of *Last.fm* tags' distribution, presented in Section 3.3.2, indicated that 60% of the top-100 most popular tags represent genre information.

[2]Exif 2.2 Specifications. http://www.digicamsoft.com/exif22/exif22/html/exif22_1.htm

aim at improving access to multimedia content and thus aim at solving *Problem 3* announced in Chapter 1. In Section 5.2 we discuss the relevant existing approaches in this area and compare them to our methods. Next, in Sections 5.3 and 5.4 we show how to use music tags for identifying the moods and themes of the songs, and respectively, how to identify potential hit songs. Section 5.5 focuses on discovering landmark information from tags attached to picture resources and in Section 5.6 we discuss the results and contributions presented along this chapter.

5.2 Specific Background

Several existing papers focus on automatically inferring additional information from available content or (user generated) metadata. Based on the type of content and metadata that they analyze, the methods we will review in this section can be classified into three classes: (1) knowledge discovery methods for *Music Resources*; (2) knowledge discovery methods for *Pictures*; and (3) knowledge discovery methods for *Web pages*. Below we present the details for all three categories, focusing mostly on the first two, the presentation being structured according to the specific aspects we also address within this chapter.

5.2.1 Knowledge Discovery Methods for Music Resources

Music Mood and Theme Detection

Quite a few of the previous works focused on music mood detection. [LLZ03] for example aims at enriching songs with mood information. The proposed approach relies on the Thayer's model and according to it, mood is entailed by two factors: stress (happy / anxious) and energy (calm / energetic), that divide the music mood space into the 4 clusters – contentment, depression, exuberance and anxious / frantic. For detecting the mood of music, timbre, intensity and rhythm features are extracted and a Gaussian Mixture Model is used to model each feature set. Unlike this approach, we do not perform any low level feature analysis of the music tracks, but rely entirely on the textual information created by the collaborative effort of the taggers. Moreover, we are not bound to only a particular music genre (like classical music in case of [LLZ03]) for which we aim to detect the songs' mood. For a given music clip, [LLZ03] first classifies it into Group 1 (contentment and depression) or Group 2 (exuberance and anxious) based on its intensity information. Then classification is performed in each group based on timbre and rhythm features. For the songs classified into Group 1, timbre features are further used to cluster them into "contentment" or "depression" songs, whereas for Group 2 rhythm features are more important for labeling the songs with either "exuberance" or "anxious".

In [FZP03], the authors propose a schema such that music databases are indexed on four labels of music mood: "happiness", "sadness", "anger" and "fear". The

relative tempo of the music tracks, the mean and standard deviation of average silence ratio are used to classify moods, using a neural network as classifier. In our approach we use more than four mood classes, since we do not consider that such a small number of moods is satisfactory and useful for the users when searching for music. The work presented by Laurier *et al.* in [LGH08] uses, like [FZP03], only four mood clusters, though a bit different ("happy", "sad", "angry", "relaxed") and for automatically classifying the songs into these classes, it makes use of both audio features and lyrics. Audio features solely yield classification accuracies between 80–90%, depending on the mood class. Using distance-based methods and Latent Semantic Analysis, the authors are able to classify lyrics better than random, but the performance is inferior to that of audio-based techniques. Methods based on differences between language models seem to give performance closer to audio-based classifiers and combining language model differences with audio descriptors boosts the performance above that of audio-based classifiers.

[SMvdP07] also aims at automatically detecting mood for music tracks and uses a set of 12 mood classes which are not mutually exclusive. However, the main focus of the paper is creating a ground truth database for music mood classification. The classification accuracy for the 12 mood classes ranges between 75 and 90% and the binary classifiers employed make use of only audio features. The approach presented in [LH07] is also entirely relying on audio features representing spectral, temporal, tonal information, as well as loudness and danceability. Here, music songs are classified into 5 mood clusters, as defined for the MIREX'07 challenge[3], the average classification accurracy being around 60%.

Somewhat complementary to our approach, [HD07] aims at studying the relationships between moods and artists, genres and usage metadata. As a test set for the experiments, the authors use *AllMusic.com*, *Epinions*[4] and a subset of *Last.fm* data.

The only published paper that we could find, touching the aspect of theme identification is the work of Mahedero *et al.* [MMC+05]. Thematic categorization is just one of the possibilities the authors discuss regarding applicability of natural language techniques on songs' lyrics. In this work, only five thematical categories are considered: "love", "violent", "protest", "Christian" and "drugs" and evaluation is performed on only 125 manually classified songs. Compared to [MMC+05], we consider more theme categories and evaluate our approach on a much bigger and expert-created ground truth dataset.

Music Hit Prediction

Some previous work focused on automatic prediction of hit songs: in [DL05], the authors explore the automatic separation of hits from non-hits by extracting both

[3]MIREX (Music Information Retrieval Evaluation eXchange) 2007. http://www.music-ir.org/mirex/2007/index.php/Main_Page

[4]Epinions. http://www.epinions.com

acoustic and lyrics information from songs using standard classifiers on these features. For this, global sounds are learned in an unsupervised fashion from acoustic data and global topics are learned from a lyrics database. Experiments show that the lyrics-based features are slightly more useful than the acoustic features in correctly identifying hit songs. As ground truth data the authors made use of the Oz Net Music Chart Trivia Page[5]. This set is somewhat limited as it only contains top-1 hits in US, UK and Australia and the corpus used in the experiments was quite small – 1700 songs. In our approach we use a larger corpus and a much richer ground truth data set – the *Billboard.com* charts. Besides, our algorithms do not rely on lyrics or acoustic information but exploit tags and social network data.

[CSB06] focuses on a complementary dimension: given the first weeks' sales data, the authors try to predict how long albums will stay in the charts. They also analyze whether a new album's position in the charts can be predicted for a certain week in the future. For the experiments, the authors used bi-weekly sales data from the *Billboard.com* magazine, specifically the Top Jazz charts. Interesting findings refer to the role of marketing before starting sales of an album, since the data shows that the higher the starting position of an album is, the longer it is likely to stay in the charts.

One of the most prominent commercial products for music hit prediction HSS[6] employs Spectral Deconvolution for analyzing the underlying patterns in music songs, *i.e.* it isolates patterns such as harmony, tempo, pitch, beat, and rhythm. Patterns in new music are then compared to patterns identified in recent chart hits. Users of this service can upload a song, the system then analyzes it and compares it against existing chart hits from its database. The resulting similarity score, its 'affinity', is a real number between 0 and 10, a score of 7.30 or above denoting clear mathematical hit potential. The drawback of this system is that by using low-level features only, it cannot correctly predict the success of completely new types of music.

[PR08] claims that the popularity of a track cannot be learned by exploiting state-of-the-art machine learning (see also [Wat07]). The authors conducted experiments contrasting the learnability of various human annotations from different types of feature sets (low-level audio features and 16 human annotated attributes like genre, instruments, mood or popularity). The results show that while some subjective attributes can be learned reasonably well from given music features, popularity is not predictable beyond-random – indicating that classification features commonly used may not be informative enough for this task. We investigate whether user generated interaction and (meta)data can serve as the missing link.

Similar to our methods are the algorithms proposed in [ACD+08], though the domain is quite different. Here, the authors make use of social media data for identifying high-quality content inside the Yahoo! Answers portal. For the community question-answering domain, they introduce a general classification framework for combining

[5] Oz Net Music Chart Trivia. http://www.onmc.iinet.net.au/trivia/hitlist.htm
[6] Hit Song Science. http://www.hitsongscience.com

the evidence from different sources of information. Their proposed algorithms prove the ability to separate high-quality items from the rest with an accuracy close to that of humans. Our algorithms have a similar goal, though applied to a different domain – music. Although, it does not use tags, to a certain extent, the work of [GGK+05] is similar to ours: the authors analyze the potential of blog posts to influence future sales ranks of books on *Amazon.com*. Even if in general it appears hard to predict future improvement or decline in rank positions for books, the study shows that there is often a strong time-correlation between blog mentions of the books and increases/decreases in their corresponding sales ranks. Moreover, the authors showed that simple predictors based on blog mentions around a product can be effective in predicting spikes in sales ranks.

5.2.2 Knowledge Discovery Methods for Pictures

Landmark Identification

The increasing popularity of the *Flickr* photo sharing service recently brought a special focus to research. In the literature, several directions can be identified, amongst the most frequent, extraction of summaries and representative views for geographic locations. Within this category, previous algorithms have employed both purely content-based techniques, as well as methods combining content and contextual information of the pictures. In [JNTD06], the authors propose a three-steps approach for generating photo summaries: in the first step geo-tagged photos are partitioned into a hierarchy of clusters; then each cluster in the hierarchy is scored and finally a flat ordering of all photos in the dataset is generated, by recursively ranking the sub-clusters at each level, starting from the leaf clusters and ending at the root. The clustering is a fixed one-time computation step, but the ranking can be re-evaluated, allowing users to specify personal preferences towards social, temporal, spatial or other available features. In later work [KNA+07], the original clustering algorithm was replaced by the K-Means algorithm and analysis of image visual features has been added. The additional step with extraction of image color, texture and interest points allowed to select photos of the same landmark from different positions and improved perceived quality of photo summaries.

A similar approach, combining context- and content-based tools is presented in [KN08]: landmarks are detected by analyzing the distribution patterns of the tags in the dataset, whereas the representative pictures for a landmark are identified based on canonical views. Using various image processing methods, the landmark images are clustered into visually similar groups, and are linked to each other if they contain the same landmark. Based on the clustering and the link structure, they then select the most representative pictures for each of these views. [BF07] uses content-based techniques for ranking iconic images labeled with a particular theme, according to how well they represent a visual category. The proposed algorithm consists of first learning

a segmentation procedure for locating the main subject inside the pictures and then it is applied to the remaining photos. The segmented test images are ranked according to shape and appearance similarity of 5 hand-labeled images per category. Three ranking algorithms are compared: random ranking inside categories (considered the baseline), ranking using similarity over the whole image, and ranking using similarity of the segmented objects from the pictures. The three ranking methods are evaluated through a user study and results show that the ranking with segmentation algorithm performs best.

We consider a similar problem of generating a summary of landmarks, but given no prior geo-spatial information. In a real world setting, the majority of pictures still does not have manually specified geographic location, so we try to find out photos of famous landmarks based on predefined training sets of known landmarks and tag co-occurrence patterns.

A complementary dimension of investigations refers to identifying time and location information for *Flickr* pictures for the purpose of photo organization: Naaman *et al.* presented in [NSPGM04] the PhotoCompass system, which utilizes time and location information for organizing personal photo collections. Pictures are organized into location- and event-based hierarchies and location names are assigned for the identified clusters. For labeling the clusters with location information they create a set of possible state, city or park names, as well as neighboring cities for each pair latitude/longitude values by matching the pictures' geo-spatial coordinates against geographical datasets. After this set is created, several heuristics are applied to select 1 - 3 of the terms as each cluster caption. [NSPGM04] is similar to our work in the sense that they also use external sources of information for inferring location names. However, in our approach the external geographical dataset is used as a training sample for an algorithm which then tries to predict for the rest of unseen *Flickr* tags if they are related to some landmark or not.

Another use of the identified landmarks / location information refers to image annotation: in [TLR03] the authors examine the synergy of location information with image based media and propose solutions for how to acquire location metadata. They identify 6 different ways of gathering location tags for image media: (1) by manual entry, (2) from the camera, (3) from a separate location-aware device, (4) from a digital calendar, (5) from the surrounding text and (6) by association with other digital documents with known location tags. Complementary to [TLR03], Davis *et al.* [DKGS04] aims to enhance photos with metadata, though not only location-related metadata, but also metadata referring to persons, objects and activities.

Detection of Other Concepts

Based on tags' semantics, some of the previous papers investigated ontology creation from *Flickr* tags [ASC07, Sch06]. For example, [Sch06] induces faceted, non-exclusive ontologies from *Flickr* data by using a subsumption-based model. [RGN07] tries to

infer event and place semantics from tag location and time usage distributions. Inspired by a burst detection algorithm, authors apply a family of Spatial Scan methods for tag semantics attribution and achieve relatively good recall and precision rates. However, compared to our approach [RGN07] also relies on geo-tagged photos.

In [SvZ08], the authors focus on a subset of *Flickr* pictures and analyze the different tag categories used by users to annotate their pictures. The analysis is performed automatically based on WordNet categories. The work is relevant to the present chapter, as it also tackles the aspect of tag recommendation. For a given photo with user-defined tags, the algorithm first derives a list of m candidate tags, based on co-occurrence information. Then, the list is processed and different aggregation and ranking strategies are applied to it, such that a ranked list of n additional tags can be suggested to the user.

5.2.3 Knowledge Discovery Methods for Web Pages

Similar to [SvZ08], [SOHB07] introduces a system, TagAssist, designed to suggest tags for blog posts. The system takes a new, untagged post, finds other blog posts similar to it, which have already been tagged, aggregates those tags and recommends a subset of them to the end user. In contrast to the tag suggestions found in systems like *Del.icio.us*, here not only popularity / frequency of a tag is considered. Beside normalization and compression (or stemming) of tags, the system exploits different heuristics and information retrieval measures to select the best candidate tags. After similar posts are retrieved, their associated tags are evaluated according to a.) the tag's occurrence frequency in the top-35 results, b.) whether a tag appears in the new post's text, c.) the number of times a tag was used in the training corpus, d.) the popularity of the blog containing a candidate post - inbound links (similar to page rank) and e.) the pair-wise comparison of similarity of each tag's related tag set to find topical clusters.

A similar technique somewhat related to item-based collaborative filtering was presented by Mishne [Mis06]. In addition to automatically generating tags by finding similar tagged content, [XFMS06] proposes the introduction of a reputation score for users to combat tag spam by spotting high quality tags (high coverage of multiple facets, high popularity, uniformity of a certain type). In suggesting tags from collective user authorities, a goodness measure (adjusted by a reward-penalty algorithm) takes the criteria for a good tagging system into account to spot the high quality tags. An approach for personalized suggestions is given by [BWC07]. Here, tags previously assigned by a user are recommended for new Web pages based on the similarity a Web site has with the pages already tagged by that user.

[LGZ08] is similar to our work in the sense that it also tries to infer additional information from user generated tags. The authors employ association rules-based techniques for discovering patterns of frequent co-occurrences of user tags. These patterns are then used to characterize and capture topics of user interests and to cluster

resources accordingly. In [LGZ08] association rules are learned from *Del.icio.us* tag co-occurrences in order to identify topics of interest and to cluster URLs / documents and users with respect to these topics. The authors emphasize the usefulness of tags as content descriptors, as they tend to highly overlap with the top keywords of a resource but seem to be more precise and "closer to the people's understanding".

In [HRGM08] the authors investigate the predictability of social tags for *Del.icio.us* bookmarks. From the features used in classification, page text was superior to anchor text, surrounding hosts, and other tags for the URL. Most tags seem to be easily predictable from the given information within a Web page. As a second way to predict tags for Web pages, association rules are learned and the inferred tags are further used for tag expansion. Mining of association rules from folksonomies has also been studied in [SHJS06] and [SHDK07], in the latter case to exploit them for user adaptation. By exploiting also tag co-occurrences, Mika [Mik07] and Schmitz [Sch06] suggested capturing the emergent semantics of folksonomies by inducing ontologies. Mika [Mik07], for example, uses co-occurrence and properties of tags to induce clusters and hierarchical orderings – superconcepts and subconcepts - of tags. By building an actor-concept graph from users and the tags they assigned, different sub-communities can be modeled, so that concepts appear in the context of the community they belong to. Two case studies are presented in the paper, in order to validate the proposed approach to ontology building – on *Del.icio.us* and *Flink*[7] data.

5.3 Inferring Music Mood and Theme Annotations

Currently no available search engine supports music search by sample music files, thus people are still constrained to search for music using textual queries. In this context, supporting users in providing meaningful tags for music tracks becomes crucial – tags and other metadata (*e.g.* extracted from ID3 tags), can be indexed together and later be used to support music search. An extensive study included in [BFNP08] showed that in the case of music resources, the majority of the generally accurate and reliable user provided tags falls into the genre category (60% of the tags) – somewhat redundant information, as this can also most often be extracted from ID3 associated tags. Considerably less frequent are tags referring to moods / opinions / qualities (20%) or themes / context / usage (5%) of the music songs, though when searching for music, the majority of queries falls into these categories – 30% of the queries are theme-related (*e.g.* "party music", "wedding songs", "mellow music"), 15% target mood information and the rest being almost uniformly distributed among six other categories. A natural question that arises is therefore: How can we support users to provide these kinds of tags? Enriching music content with tags from these categories

[7] Flink is a system offering a Web-based presentation of the social networks and research interests of Semantic Web researchers. The community of researchers represented in Flink includes all authors, program committee members and organizers of all past International Semantic Web events from 2001, altogether 607 persons. http://flink.semanticweb.org

will definitely improve precision and recall, as they are typical of how we think or talk about music – emotions and context are highly interlinked with music perception.

One possibility to make users use keywords from the categories we need is to unobtrusively recommend such tags and thus support the users in the tagging process. Besides minimizing cognitive load by changing the task from generation to recognition [SOHB07] such recommendation of under-represented but valuable tags will very likely trigger reinforcement, *i.e.* enforce preferential attachment. As presented in [SLR+06, HRS07], seeing previous tag assignments from other users strongly influences which tags will be assigned next and thus to which tag set a resource's vocabulary will converge.

In this section we will focus on supporting users tag music tracks with tags referring to opinions and usage context information. We will refer to these categories as "moods" and "themes". With the "mood of a song" we understand the state or the quality of a particular feeling induced by listening to that song (*e.g. aggressive*, *happy*, *lazy*, *sad*, *sentimental*, *etc.*). The "theme of a song" refers to the context or situation which fits best when listening to the song, *e.g. at the beach*, *dinner ambiance*, *night driving*, *party time*, *etc.* Consider for example the song of ABBA, "Dancing Queen": by listening to the song or just considering the lyrics (*"Friday night and the lights are low / looking out for the place to go / where they play the right music / getting in the swing / you come to look for a king ..."*) one immediately gets transposed into a weekend party atmosphere and an enjoyable state of mind. It would therefore be natural to describe and also search for this song with mood related words such as "fun", "happy", *etc.* and with the theme tags: "Party Time", "Thank God It's Friday!" or "Girls Night Out". Nevertheless, when inspecting the tags *Last.fm* users provided for this track, we cannot really identify these concepts. Instead, tags such as "pop", "disco", "70s" or "dance" are quite often employed. Therefore, in this section we will propose algorithms which can provide users with mood- and theme-related tags to choose from during the tagging process. For comparison reasons, we also experiment and compare with genre / style-tag recommendations, as this task is much easier to perform and has already been investigated [LOL03].

5.3.1 Datasets

For obtaining the datasets for our experiments we used several data sources: *Last.fm*, *AllMusic.com*, *www.lyricsdownload.com* and *www.lyricsmode.com*. In the following we present some relevant statistics for all of them.

AllMusic.com. Established in 1995, the *AllMusic.com* website was created as a place and community for music fans. Almost all music genres and styles are covered, ranging from the most commercial popular to the most obscure ones. Not only genres can be found on *AllMusic.com*, but also reviews of albums and artists within the context of their own genres, as well as classifications of songs and albums according to themes, moods or instruments. All these reviews and classifications are

manually created by music experts from the *AllMusic.com* team, therefore the data found here serves as a good ground truth corpus.

For our experiment we collected the *AllMusic.com* pages corresponding to music themes, moods, genres and styles. We could find 178 different moods, 73 themes, 20 genres and 633 styles (more fine-grained music genre classes). Figure 5.1 shows the distribution of songs per mood, theme, and style in *AllMusic.com*. Although it's not a power law distribution, the figure still shows a relatively large amount of songs annotated with a restricted set of moods, themes, or styles. From the pages corresponding to moods / themes / genres / styles, we also gathered information related to which music tracks fall into these categories and we restrict the data set to contain only tracks also present in our *Last.fm* crawl. With this procedure, we ended up with 13,948 songs. Looking at the songs identified in each of the categories, we have 7,750 track-moods, 1,164 track-themes, 1,521 genre-tracks assignments, and 16,023 track-style pairs. Figure 5.2 shows the average and the standard deviation of the number of moods, themes, and styles per song. On average songs are annotated with 1.73 moods, 1.21 themes and 1.65 styles with maximum number of annotations of 12, 6 and 9 respectively.

Last.fm. For the purpose of our investigations, we crawled an extensive subset of the *Last.fm* website, namely pages corresponding to tags, music tracks and user profiles. We started from the crawl described in Section 3.3.1 and recollected the information related to tags associated to music tracks. From all tracks that we obtained from *AllMusic.com*, we could also find 13,948 of them in the *Last.fm* data set. For this intersection we had 81,964 different tags and for each of these tags we have extracted information regarding the number of times each tag has been used. In Figure 5.3 we show a log scale plot of usage frequencies for tags attached to the songs in the intersection of *AllMusic.com* and *Last.fm* datasets, for which we had style, mood or theme information. The plot shows, as expected, a power law distribution of tag frequencies for all types of annotations.

Lyrics. To investigate whether another source of information, namely lyrics as one part of music content, can provide added value in the task of mood, theme and genre / style recommendation, we also obtained the corresponding lyrics for the 13,948 tracks from the intersection of *AllMusic.com* and *Last.fm* songs, if available. For this purpose we used a previous crawl (described in Section 3.4.3 and also included in [BFNP08]) of the *www.lyricsdownload.com* site. Additionally, we crawled the *www.lyricsmode.com* website, such that we could gather lyrics information for a total of 6,915 tracks (6,592 song lyrics found on *www.lyricsdownload.com* and 323 on *www.lyricsmode.com*). In total we had 27,817 words appearing in the lyrics of songs having attached either a style, mood, or theme. Figure 5.4 shows a log scale plot of usage frequencies for the words across the three sets corresponding to the songs having mood / theme / style labels. Similar to the case of tag frequencies, we find a power law distribution of word frequencies.

Considering only those music tracks for which also lyrics information was available,

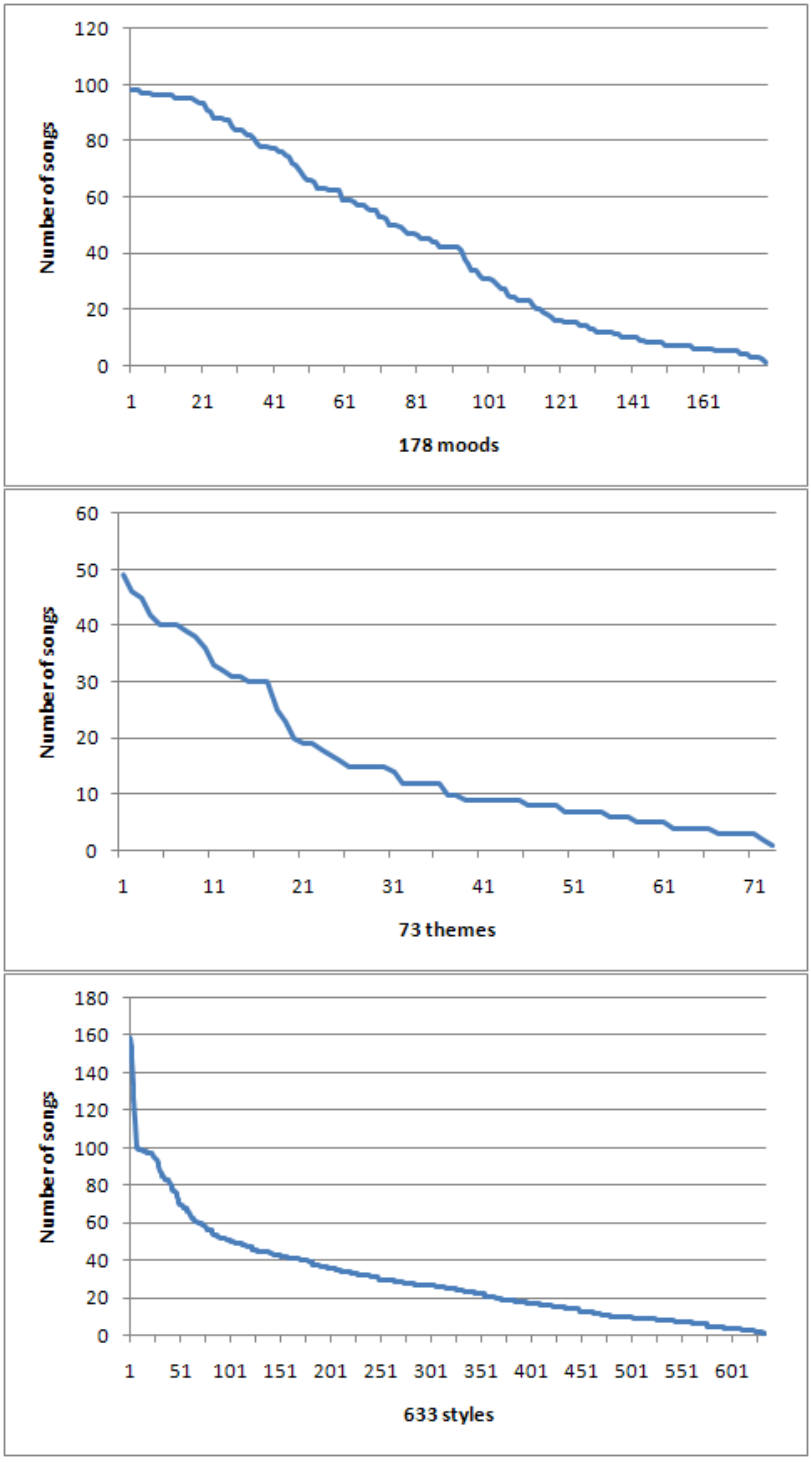

Figure 5.1 Number of songs per mood, theme, and style

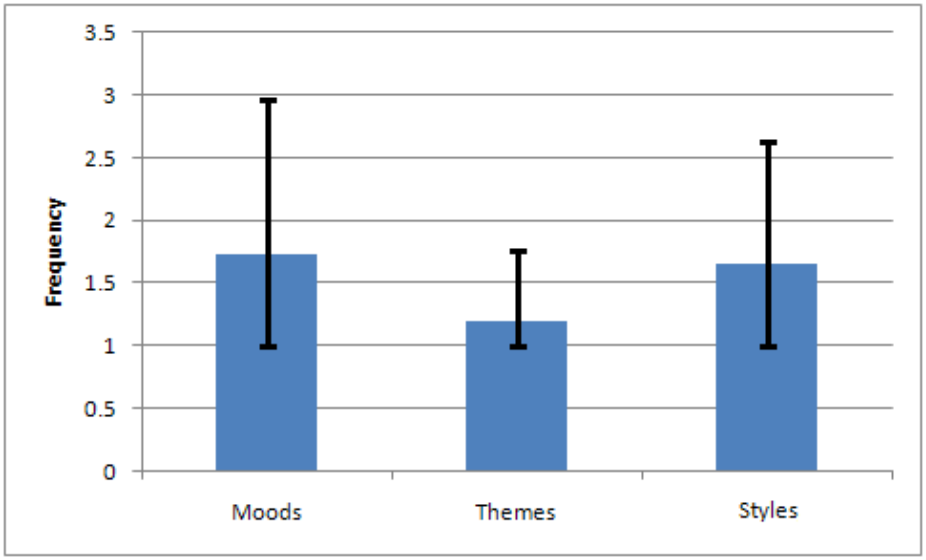

Figure 5.2 Average and standard deviation for the number of moods, themes and genres / styles per song

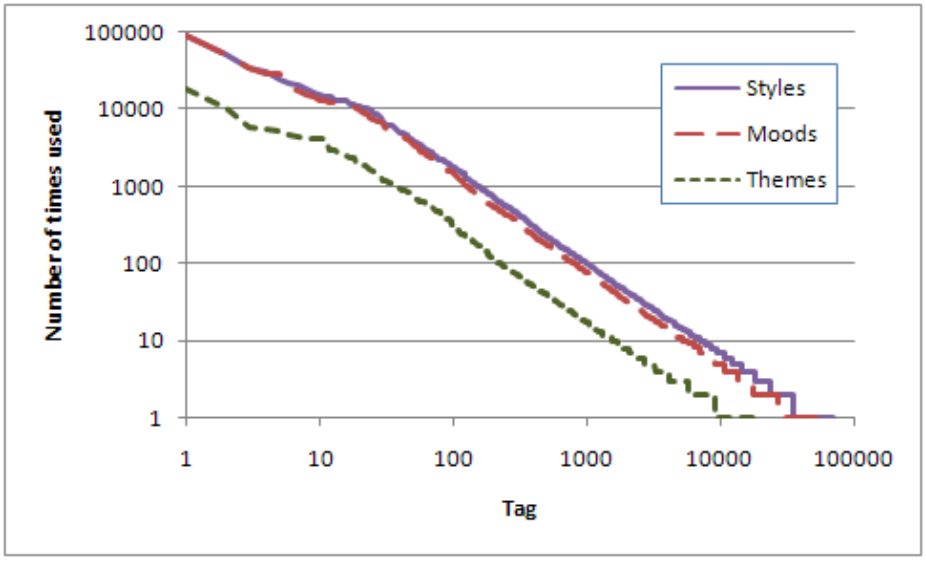

Figure 5.3 Tag frequencies for songs in the intersection of *AllMusic.com* and *Last.fm* data sets

for our experiments we thus had at our disposal: 6,116 song-mood pairs, 892 track-theme, 655 tracks with lyrics and genre information and 8,155 track-style pairs.

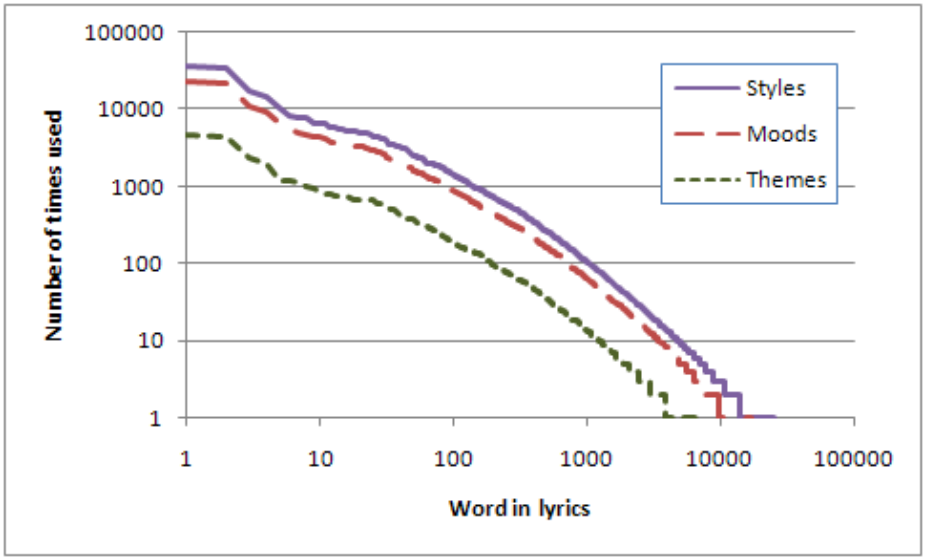

Figure 5.4 Word frequencies in lyrics of songs in the intersection of *AllMusic.com* and lyrics data sets

5.3.2 Algorithm

For recommending themes and moods, we base our solution on collaboratively created social knowledge – *i.e.* tags associated to music tracks – extracted from *Last.fm*, as well as on lyrics information. Based on already provided user tags, on the lyrics of music tracks, or on combinations of the two, we build classifiers which try to infer other annotations corresponding to moods and themes of the songs. Additionally, for comparison reasons we also experiment with predictions of music genres and the finer grained styles. Our approach thus relies on the following hypotheses:

1. Existing tags provided by users for a particular song carry information which can be used to infer the mood, theme or genre of that song – *e.g.* songs tagged with "hard-rock" are more likely to have an "aggressive" mood than "mellow" - tagged songs.

2. The lyrics of the tracks give a hint on the mood, theme or genre of the songs – *e.g.* tracks with love-related lyrics belong to the "blues" genre, have "romantic evening" as theme and correspondingly, a "romantic" mood.[8]

[8] Juslin and Laukka reported in [JL04] that 29% of the people consider lyrics as a factor of how music expresses emotions, showing thus the relevance of studying lyrics in this context.

Class \ Feature Type	Tags	Lyrics	Tags+Lyrics
Themes	787	1,037	1,824
Moods	6,170	3,976	10,146
Genres	7,710	5,435	13,145
Styles	7,710	5,435	13,145

Table 5.1 Number of feature tokens per feature type and class

The core of the mood, theme and genre / style recommendation methods is a probabilistic classifier trained on the *AllMusic.com* ground truth using tags and/or lyrics as features. Separate classifiers correspond to the different types of classes that we aim to recommend. For building the classifiers, we use the open source machine learning library Weka[9]. In the experiments presented in this paper, we use the Naïve Bayes Multinomial implementation available in Weka. We also experimented with other classifiers (*e.g.* Support Vector Machines, Decision Trees), which resulted in similar classification performances, but were much more computationally intensive. Depending on which kind of annotations we aim to recommend – moods / themes / genres / styles – the classifiers we build are trained to predict the membership of songs to classes of moods, themes and genres / styles respectively. We have one classifier trained for the whole available set of classes (*i.e.* either for moods or themes or genres or styles). This classifier produces for every song in the test set a probability distribution over all classes (*e.g.* over all moods). Thus, one or more classes (based on probabilities or on a given rank number) can be then assigned to each song.

Based on the hypotheses enumerated above, we also experiment with three types of input features for the different sets of classifiers. We can thus have as input features: (1) tags; (2) words from lyrics; or (3) tags *and* words from lyrics. Depending on the type of features used to train the classifier and on the type of class that the classifier will assign to songs, we propose 12 methods (4 types of output classes – moods, themes, genres, styles – and 3 types of features – tags, lyrics, tags+lyrics). Each of these algorithms uses a different number of input features, as the sets of AllMusic songs having mood, theme and genre / style labels do not overlap perfectly. We summarize the number of features available for each algorithm in Table 5.1. This is the full available feature set for each class. We experimented with feature selection based on automatic methods (*e.g.* Information Gain) but the results showed that the full set is better suitable for learning, even though it contains some noise.

Algorithm 1 presents the main steps of our approach. We show the algorithm for mood recommendations based on tag features, the other algorithms are corresponding variants.

[9]Weka. http://www.cs.waikato.ac.nz/~ml/weka

Alg. 1. Mood recommendation based on tag features

1: Apply clustering method to cluster moods (*optional*, see Section 5.3.3)
2: Select classes of moods M to be learned
2a: For each mood class
2b: If the class does not contain at least 30 songs
Discard class
3: Split song set S_{total} into
S_{train} = songs used for training the classifier
S_{test} = songs used for testing the recommendations
4: Select tag features for training the classifier
4a: For each song $s_i \in S_{train}$
4b: Create feature vector $F(s_i) = \{t_j | t_j \in T\}$, where
T = set of tags from all songs in all mood classes
$t_j = \begin{cases} log(freq(t_j) + 1), \text{if } s_i \text{ has tag } t_j; \\ 0, \text{otherwise.} \end{cases}$
5: Train Naïve Bayes classifier on S_{train} using $\{F(s_i); s_i \in S_{train}\}$
6: For each song $s_i \in S_{test}$
6a: Compute probability distribution $P(s_i)$, $P(s_i) = \{p(m_j|s_i); m_j \in M\}$
6b: Select top k moods M_{top-k} from M based on $p(m_j|s_i)$
6c: Recommend M_{top-k} to the user

Step 1 of the algorithm above aims to reduce the number of mood classes to be predicted for the songs. This step is optional (described in detail in Section 5.3.3), as we experiment with all classes of moods / themes from AllMusic, as well as with a subset resulted from applying a clustering method on the original set. In the case of genre and style predictions we do not apply any clustering method, since the distinction among music genres is mostly clear for the users. This is not the case for moods, where *AllMusic.com* provides 178 labels many of which are hardly distinguishable for a non-expert. If two classes are clustered, the resulted class will contain all songs which have been originally assigned to any of the composing classes. As we need a certain amount of input data in order to be able to consistently train the classifiers, we discard those classes having less than 30 songs assigned (step 2).

After selecting separate sets of songs for training and testing in step 3, we build the feature vectors corresponding to each song in the training set (step 4). In the case of features based on tags, the vectors have as many elements as the total number of distinct tags assigned to the songs belonging to the mood classes. The elements of a vector will have values depending on the frequency of the tags occurring along with the song, or 0 if they have not been used for that song. We experimented with different variations for computing the vector elements, but the formula based on the logarithm of the tag frequency provided best results. Once the feature vectors are constructed, they are fed into the classifier and used for training (step 5). A model is learned and afterwards is applied to any new, unseen data. We can choose how many moods are recommended to the user based on the probabilities resulted from the classification or by setting an absolute threshold.

Primary (Man. 1^{st})	**Secondary Emotion (Man. 2^{nd})**
Love	Affection, Lust, Longing
Joy	Cheerfulness, Zest, Contentment, Pride, Optimism, Relief
Anger	Irritation, Exasperation, Rage, Disgust
Sadness	Suffering, Sadness, Disappointment, Neglect, Sympathy
Fear	Horror, Nervousness
Neutral	Complex, Sophisticated, Spiritual

Table 5.2 Adapted emotion hierarchy

5.3.3 Data Preprocessing

Given the fact that the number of classes existing in *AllMusic.com* is quite large (*e.g.* 178 different moods) and thus it is very difficult for an untrained user to distinguish between the different classes and correctly assign labels to music tracks, we applied different clustering methods on the initial set. This procedure was applied in the case of moods and themes, employing several clustering methods. In Table 5.3 we present some samples of the resulted clusters for moods and themes when applying different clustering algorithms. In the following we present the details of these methods.

Manual Clustering

Moods. For grouping the 178 *AllMusic.com* moods we made use of the extensive work already done on studying human emotions. Though there is little agreement on the exact number of basic emotions let alone on a taxonomy including combinations of the basic concepts into complex, secondary emotions, we found the hierarchy reported in Shaver *et al.* [SSKO87] useful for our task. Moods are usually considered very similar to emotions but being longer in duration, less intensive and missing object directedness. This taxonomy comprises 6 primary emotions that can be differentiated across dimensions like valence/pleasantness or arousal/activity: *Love*, *Joy*, *Surprise*, *Anger*, *Sadness*, and *Fear*, and each of the first level classes has in turn 1 to 6 corresponding secondary emotions.

In manually categorizing the *AllMusic.com* moods we had to slightly adapt the taxonomy to fit our data: *Surprise* was removed since no example moods were found; the same happened for some secondary emotions. Since some moods do not actually denote a mood (*e.g.* "literate"), we introduced a new class (*Neutral*) with three second level classes. In total, we obtained 23 second level classes ("Man. 2^{nd}") falling into 6 first level classes ("Man. 1^{st}") as presented in Table 5.2.

Themes. Since *AllMusic.com* themes do not directly correspond to human emotions, mapping the 73 theme terms into the moods taxonomy used before was not possible. However, these themes are strongly related, *i.e.* usually associated, with different moods. For manual clustering we adopted the procedure used in [SSKO87]

for building the aforementioned taxonomy of basic and secondary emotions. In a similarity sorting task, all *AllMusic.com* theme terms written on cards were sorted by the authors into as many and as high piles as seemed appropriate. Individual co-occurrence matrices were built and added to find good groupings by analyzing the clusters. Unclear membership of singular labels was resolved after discussion. Applying this method resulted in a reduction of the theme list to 20 labels.

Co-occurrence - based Clustering

Moods. The second method we applied for clustering relies on the number of songs assigned to each mood label. Given two moods, M_1 and M_2, N_1 and N_2 are the corresponding number of songs assigned by the *AllMusic.com* experts to each of these moods respectively. N_{12} is the number of common songs associated to both M_1 and M_2. We cluster M_1 and M_2 if the clustering score C_{12} satisfies:

$$C_{12} = \frac{N_{12}}{min(N_1, N_2)} > 0.1 \tag{5.1}$$

If condition (5.1) is satisfied, M_1 and M_2 are placed into the same class M', containing the union of the two sets of songs corresponding to M_1 and M_2. This process is repeated several times over all possible pairs of moods, until no more possible clustering candidates are found, or until resulted clusters contain already 5 moods (as it would otherwise lead to one very large cluster). With this approach we reduce the mood space to 77 classes. To formalize, the co-occurrence - based clustering looks as follows:

Alg. 2. Moods Co-occurrence - based clustering

```
While no more changes to mood set MS
    For i=1 .. |MS| − 1
        For j=(i+1) .. |MS|
            Compute clustering score C_ij for M_i and M_j (Eq. 5.1)
            If (C_ij > 0.1 and |M_i| + |M_j| ≤ 5)
                cluster M_i and M_j into M'
    Recompute MS based on already clustered moods
        Remove from MS all clustered themes (e.g. M_i, M_j,...)
        Add to MS all resulted cluster sets (e.g. M',...)
```

Themes. For the case of themes, the co-occurrence - based algorithm is similar, with the only difference that we operate on the set of themes and the resulting set of theme clusters contains 44 classes.

WordNet - based Clustering

Moods. This clustering approach exploits the semantic meanings of the *AllMusic.com* moods. For each of the 178 moods, we first extract from WordNet[10] the corresponding set of synonyms, $syns\,[i]$. Then the same procedure is applied for each of the synonyms appearing in the synset of a mood. Thus, for each mood label M_i we have a set of words $s\_syns\,[i]$ representing its synonyms and the synonyms of the synonyms. These sets of words, s_syns, are compared pairwise and if there is any overlap among them, the corresponding mood labels are collapsed. With this procedure we reduce the mood space to 154 categories. This algorithm is presented below:

Alg. 3. Moods WordNet - based clustering

```
1:  syns[] = new syns[|MS|], MS – set of moods
2:  s_syns[] = new s_syns[|MS|]
3:  For i=1 .. |MS|
4:      syns[i] = {WordNet_synonyms (M_i)}
5:  For i=1 .. |MS|
6:      words_i = syns[i]
7:      s_syns[i] = syns[i]
8:      For j=1 .. |words_i| − 1
10:         synW = {WordNet_synonyms (words_i [j])}
11:         add(s_syns[i], synW)
12:  For i=1 .. |MS|-1
13:      For j=i+1 ... |MS|
14:      Compute OV = overlap(s_syns[i], s_syns[j])
15:      If (OV > 0) cluster M_i and M_j into M'
```

Themes. The WordNet-based clustering of themes, like in the case of the moods, aims at clustering semantically related theme labels. The approach is however a bit different than in the case of moods, as theme labels are not all the time single-word concepts. On average, the 73 themes have 1.6 words (including stopwords; and 1.55 when discarding the stopwords). For each of the 73 themes we first process the corresponding words this theme consists of. All stop words are removed, and for the remaining words we extract the corresponding WordNet synonyms. All resulted synsets are compared pairwise and if the overlap between two sets is at least two words, the corresponding themes are clustered. With this procedure, the resulted set of themes contained 58 entries. Below we present all steps of this algorithm:

Alg. 4. Themes WordNet - based clustering

```
1:  For i=1 .. |TS|, TS - set of themes
2:      WS[i] = {word_k|word_k ∈ T_i, k = 1..m}, m=|T_i|
3:  For i=1 .. |TS|
4:      For j=1 .. |WS[i]|
5:          If (!stopWord(WS[i][j])) add(WS'[i], WS[i][j])
6:  For i=1 .. |TS|-1
```

[10] WordNet. http://wordnet.princeton.edu

7: For j=1 .. $|TS|$
8: For j=(i+1) .. $|MS|$
9: Compute OV = overlap(WS'[i], WS'[j])
10: If ($OV > 2$) cluster T_i and T_j into T'

Comparing and Pruning the Clusters

The different clustering methods lead to quite distinct cluster types, characterized by different sizes and different numbers of songs captured by the clusters' labels (see examples in Table 5.3). Classes containing less than 30 songs are discarded in order to have a minimal representative learning corpus for the classifier. This means that the number of classes that will be used for deriving themes / moods / genre / styles labels will be less than the number of *AllMusic.com* original classes, and less than the resulting number of classes after applying some clustering method. In Table 5.4 we present the overview of the number of classes resulted after using one of the clustering methods (rows "Clusters") and after discarding those classes with less than 30 tracks (rows "Pruned"). For comparative reasons we also include the original number of *AllMusic.com* classes (when no clustering method is applied – column "None") and the corresponding numbers of clusters that are kept considering the 30 tracks per class constraint. In case of genres and styles no clustering method was applied and additionally, for the case of moods we present the number of clusters and pruned number of classes when applying both first and second level manual clustering methods (column "Man. $1^{st}/2^{nd}$"). Only the classes remaining after the pruning step will be used for our experiments. As described in Section 5.3.2, Weka's Naïve Bayes Multinomial implementation was used to train and recommend moods, themes and genres / styles.

By inspecting the different resulted clusters, we observe that when using WordNet, due to the fact that only direct synonyms are found, but not strongly related or associated terms (*e.g.* "sexy" is merged with "intimate", but not with "sexual") the set of moods and themes was only slightly reduced. Most often, the synonymous labels combined via WordNet were also clustered together through the manual method, or at least belonged to the same primary emotion, *i.e.* superclass. Interestingly, this was not the case for co-occurrence clustering, which means the synonyms were rarely used together on the same songs – probably due to redundancy / information gain considerations. Co-occurrence clustering, seems to be a good compromise in terms of class number, 43% of mood labels and about 60% of themes remained. In general, resulting clusters were slightly bigger (on average 2.3 labels/class) and almost half of these clusters contained more than one term. In contrast to WordNet, instead of taking context-free direct synonyms, near-synonyms and strongly associated words were grouped together based on frequent co-usage for music. Manual clustering leads to the greatest reduction of classes as for themes it tried to build upon both, synonymy relationships as well as loose (usage/context) associations like "Party Time",

MOOD CLUSTERS – MANUAL CLUSTERING
Love.Lust: Passionate / Sexual / Sensual / Sexy
Anger - Rage: Angry / Fierce / Outrageous / Bitter / Outraged / Hostile / Aggressive / Acerbic / Rambunctious / Thuggish / Malevolent
Sadness.Sadness: Somber / Bittersweet / Wistful / Gloomy / Self-Conscious / Bleak / Sad / Brooding / Melancholy
THEME CLUSTERS – MANUAL CLUSTERING
Day Driving / Road Trip / Night Driving
Guys Night Out / Girls Night Out / Party Time / Cool&Cocky / Drinking
Freedom / Maverick / Revolutionary / Patriotic / Victory
MOOD CLUSTERS – CO-OCCURRENCE CLUSTERING
Party / Celebratory / Fun / Happy / Carefree
Soothing / Romantic / Calm / Peaceful / Sentimental
Somber / Sad / Gloomy / Brooding / Angst-Ridden
THEME CLUSTERS – CO-OCCURRENCE CLUSTERING
Guys Night Out / Drinking
Heartache / Feeling Blue / D-I-V-O-R-C-E
Slow Dance / Seduction/ Romantic Evening / In Love / Sexy
MOOD CLUSTERS – WORDNET CLUSTERING
Exciting / Rousing
Elegant / Elaborate / Refined / Mannered
Intimate / Sexual
THEME CLUSTERS – WORDNET CLUSTERING
Heartache / Loss / Grief
Early Morning / Monday Morning
In Love / New Love / Stay in Bed

Table 5.3 Samples of resulted moods and themes clusters, applying different clustering methods

"Drinking" and "Girls Night Out".

For moods, manual clustering focused on characterizing the underlying concepts (*i.e.* emotions) along dimensions like pleasantness and activity. Looking at the discussions about how many universal basic emotions exist, it is not a surprise that the result consists of few clusters with many labels that are very similar in nature. Especially when it comes to music perception it is unclear how many different moods people really distinguish – in terms of linguistic description, physiological reaction *etc.* The results of our evaluation experiments will also shed a little more light on that.

5.3.4 Automatic Evaluation

With our first evaluation we aim at automatically measuring the quality of our tag prediction algorithms. As ground truth data we use a subset of the *AllMusic.com* data set. Being manually created by music experts, the assignments of songs to classes of moods, themes or genres / styles can be considered correct and thus accepted as ground truth. Given a *Last.fm* music track, we predict possible mood / theme / genre / style annotations and compare our output against the AllMusic experts'

		Clustering Method			
		None	**Man. $1^{st}/2^{nd}$**	**Co-occ.**	**WordNet**
#Moods	Clusters	178	6 / 23	77	154
	Pruned	89	6 / 22	39	82
#Themes	Clusters	73	20	44	58
	Pruned	11	12	12	9
#Genres	Clusters	20	-	-	-
	Pruned	18	-	-	-
#Styles	Clusters	633	-	-	-
	Pruned	109	-	-	-

Table 5.4 Nr. of moods / themes resulted clusters based on the applied clustering method and remained number of clusters after pruning

assignments for the same song.

Since our goal is recommendation of relevant annotations, we choose the following metrics for the evaluation of our results:

- **Hit rate at rank k ($H@k$).** Hit rate at rank k is defined as the probability of finding a good descriptive tag among the top-k recommended tags. For our evaluation, we consider $k = 3, 5$.

- **R-Precision (RP).** The idea here is to generate a single value summary of the ranking by computing the precision at the R-th position in the ranking, where R is the total number of relevant documents. This value also equals to the recall at rank R. The R-precision measure is useful for observing the behavior of an algorithm for each individual.

- **Mean Reciprocal Rank (MRR).** Mean reciprocal rank [Voo99] is a statistical measure used to evaluate a process which outputs a list of possible hits for a query, ordered by the probabilities of their correctness. The reciprocal rank of a query response is the multiplicative inverse of the rank of the first correct answer and the mean reciprocal rank is the average of the reciprocal ranks of results for a sample of queries Q.

For this first automatic evaluation we perform, we concentrate on the $H@3$ metric, as we recommend three annotations to the users to choose from. We consider three annotations a good compromise, between providing enough suggestions and at the same time not overwhelming the users with too much information. We present the results for all our experimental runs in Table 5.5. These runs correspond to the different combinations of classes, features and optionally a clustering algorithm applied to the initial set of classes, in total resulting 33 different recommendation methods. For an easier overview, Figure 5.5 summarizes $H@3$ values for recommending annotations based on tags. This figure also contains results presented in Section 5.3.5, "User Evaluation", for direct comparison.

	Clustering	Features	H@3	H@5	RP	MRR
Theme	-	Tags	0.80	0.92	0.49	0.67
	-	Lyrics	0.56	0.72	0.26	0.46
	-	Tags+Lyrics	0.80	0.94	0.48	0.67
	manual	Tags	0.73	0.85	0.41	0.61
	manual	Lyrics	0.59	0.73	0.31	0.49
	manual	Tags+Lyrics	0.74	0.86	0.44	0.63
	co-occ.	Tags	0.76	0.88	0.48	0.65
	co-occ.	Lyrics	0.54	0.71	0.28	0.47
	co-occ.	Tags+Lyrics	0.78	0.90	0.47	0.65
	WordNet	Tags	0.85	0.94	0.47	0.66
	WordNet	Lyrics	0.72	0.85	0.38	0.59
	WordNet	**Tags+Lyrics**	**0.88**	**0.96**	**0.48**	**0.69**
Mood	-	Tags	0.39	0.51	0.17	0.34
	-	Lyrics	0.17	0.25	0.06	0.17
	-	Tags+Lyrics	0.37	0.48	0.15	0.32
	Man. 1^{st}	Tags	0.88	0.99	0.49	0.71
	Man. 1^{st}	Lyrics	0.82	0.98	0.42	0.65
	Man. 1^{st}	**Tags+Lyrics**	**0.89**	**0.99**	**0.52**	**0.73**
	Man. 2^{nd}	Tags	0.63	0.76	0.31	0.53
	Man. 2^{nd}	Lyrics	0.49	0.65	0.21	0.41
	Man. 2^{nd}	**Tags+Lyrics**	**0.64**	**0.78**	**0.31**	**0.52**
	co-occ.	Tags	0.58	0.71	0.30	0.50
	co-occ.	Lyrics	0.38	0.51	0.16	0.34
	co-occ.	Tags+Lyrics	0.58	0.70	0.29	0.49
	WordNet	Tags	0.39	0.52	0.18	0.34
	WordNet	Lyrics	0.19	0.28	0.07	0.19
	WordNet	Tags+Lyrics	0.38	0.50	0.16	0.33
Genre	**-**	**Tags**	**0.97**	**0.98**	**0.83**	**0.91**
	-	Lyrics	0.85	0.93	0.60	0.75
	-	Tags+Lyrics	0.93	0.98	0.76	0.86
Style	**-**	**Tags**	**0.76**	**0.85**	**0.48**	**0.65**
	-	Lyrics	0.22	0.29	0.10	0.21
	-	Tags+Lyrics	0.62	0.72	0.37	0.54

Table 5.5 Experimental results: $H@3$, $H@5$, RP, MRR for the different algorithms

We observe that the best performing methods are those using tags as input features for the classifiers. The methods using only lyrics as features perform worst. When combining tags and lyrics as features, the corresponding methods perform much better than those based only on lyrics and they sometimes also slightly outperform the tag-based methods. These results confirm once more the quality of user provided tags – a result also observed in [BFNP08] – as well as hypothesis 1 on which our approach relies (see Section 5.3.2). Lyrics, in contrast to tags, introduce noise, as many song texts contain all sorts of interjections (*e.g.* "hey", "oh", "uh-huh", *etc.*), slang or simply informal English. With lyrics features the best results are obtained for genre and theme recommendations – the second hypothesis on which we built our approach. Though alone they are obviously not descriptive enough to decide well upon genre or theme, by setting the topic, lyrics seem to help removing some tag ambiguity for identifying appropriate themes. In contrast, lyrics do not seem to be indicative of the mood of a song.

As expected, the best results we obtain are for the genre-tag recommendations:

H@3 of 0.97 for the case of tags as features. Styles do not perform as well as genres (H@3 of 0.76), mostly due to the fact that the AllMusic labels are too fine-grained to clearly distinguish between them (109 classes). Given the difficulty of agreeing on a single, appropriate music genre taxonomy, some of these fine distinctions may also be worth discussing. For the case of theme recommendations, the best results, H@3 of 0.88, are achieved for the algorithm using a combination of tags and lyrics as features and applying a WordNet-based clustering on the theme classes. Overall, theme recommendations using WordNet-based class clustering perform best, compared to the other methods applying either no clustering, manual- or co-occurrence-based methods.

Compared to themes, mood recommendations do not perform as well when using many classes, they achieve only a H@3 of 0.64. The best performing algorithm uses manual clustering of the moods, and more specifically, the method using tags and lyrics as input features. For the case of moods, we present the results corresponding to both first and second level manual clustering of the original AllMusic classes (rows "Man. 1^{st}" and "Man. 2^{nd}"). Reducing the cluster number to the 6 first level classes ("Man. 1^{st}") corresponding roughly to basic human emotions, boosts the performance considerably and for this case we achieve a H@3 value of 0.89. Though having a larger mood vocabulary for recommendations should be aimed at, trade-offs are necessary. It is an interesting question for future work, how many classes are appropriate to describe what mood distinctions people actually do when listening or referring to music.

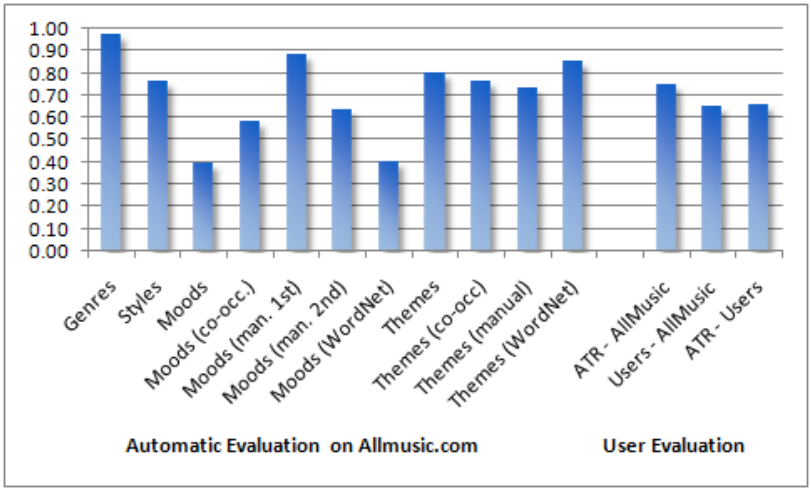

Figure 5.5 H@3 values for different types of recommended annotations, when using various clustering methods and tags as features

The results presented so far (Table 5.5, Figure 5.5) indicate the performance of our algorithms in correctly recommending moods, themes or genre / style annotations, *i.e. macro* evaluation results. However, we were also interested in *micro*-evaluating our algorithms. More specifically, we also analyzed the results per specific annotation

	Best	H@3	Worst	H@3
Theme	Slow Dance	0.97	Late Night	0.52
	Romantic Evening	0.89	Summertime	0.62
	Autumn	0.89	Party Time	0.72
Mood	Ethereal	0.65	Precious	0.00
	Hypnotic	0.64	Calm/Peaceful	0.00
	Angst-Ridden	0.57	Rambunctious	0.00
Genre	Electronica	0.99	Easy Listening	0.00
	Rap	0.96	Cool & Gospel	0.65
	Country	0.96	Comedy	0.70
Style	Grunge	0.98	Chicago Soul	0.00
	Trash	0.98	Blue-Eyed Soul	0.03
	Industrial Metal	0.97	Nashville Sound/Countrypolitan	0.03

Table 5.6 Examples of best and worst performing (by $H@3$) classes, without clustering, learned using tags as features

class to find out which classes offer the best performances and which classes are more difficult to annotate with. Table 5.6 shows $H@3$ values for the different classes without applying any clustering method and using tags as features.

The differences show that while some classes are relatively easy to recommend, others may require special attention or some level of disambiguation. Also, classes which are hard to recommend are ambiguous and the annotations are mostly subjective. Themes like "Late Night" or "Summertime" strongly depend on the person and what s/he is used to be doing late night or in summer. The same is true for moods like "Precious" or "Rambunctious". They can be subjectively interpreted in several ways. In the case of genres, "Easy Listening" shows a perfect example of extremely subjective annotation. We also find it hard to distinguish among very fine grained styles like "Chicago Soul" and "Blue-Eyed Soul". On the other hand, classes which can be recommended with high accuracy are also very clearly defined, may it be a theme like "Slow Dance", a mood like "Hypnotic", or even genres and styles like "Electronic" or "Grunge".

It is difficult to directly compare our results to the related work cited in Section 5.2.1, as each paper uses a different number of classes. Moreover, experimental goals, ground truth and evaluation procedures vary as well, or detailed descriptions are missing – like strict classification into one class or proposing up to many classes for one piece of music.

5.3.5 User-based Evaluation

For evaluating the quality of our recommended themes for music tracks in terms of user judgments, we set up a user survey, where users had to manually label songs with one or more theme classes used in our algorithms and in *AllMusic.com*. Having already compared our approach against the expert annotations from *AllMusic.com*, we want to also analyze how average users perform, compared to the same ground

truth.

Due to the rise of social networking platforms and the enormous amount of people that can be reached by mechanisms of viral marketing, we decided to perform our experiments within Facebook[11], to easily acquire new users. Facebook is a social networking platform launched in 2004 that has more than 110 million users. Facebook's success is also due to their opening of the platform – Facebook provides an API to enable everyone to write their own applications. This has the potential that the participants in our evaluation will be more representative of all computer users and less biased with respect to Computer Science (*i.e.* co-workers, students).

We developed a small Facebook application, "Mood Mates!" (see Figure 5.6 for a screenshot), in which users can choose from eleven given theme labels those that they think best describe the given song. The song can be played back at least in a 30 seconds preview, in case it was not known to the user. Users can tag as many songs as they want, and to motivate them to invest this effort we offer a feature for "social comparison" with their friends, as well as with unknown - but like minded - application users and music experts (from *AllMusic.com*). Thus, after labeling the first 10 songs a user automatically sees the results of the social comparison, his matches with friends and other people already having taken this "test". Besides, the "Hall of Fame" feature provides the users a list with the best users' scores and their placement in this list – again a motivating factor for attracting more participants.

Given the fact that we had to motivate users to participate in our experiment, we had to restrict the number of tag labels the users had to inspect and choose from. We decided to perform the evaluation for the case of theme recommendations, as these labels are the easiest to understand, especially for non-native English speaking users. We restricted the test set of songs to only those tracks having theme labels assigned by AllMusic experts. 11 themes and 315 different tracks met this restriction. Though acquiring users proved out to be more difficult than we thought, after two months we had 61 users for the application. They annotated 113 songs, resulting in a total of 956 user-song-theme annotations. On average, users assigned 2 themes per song, while for the same data, on *AllMusic.com* the average was 1.08 themes per song.

With the Facebook-based user survey, we aim to compare not only the performance of normal users against the *AllMusic.com* experts, but also the results of our algorithm (ATR – Automatic Theme Recommendation) against the choices of the users. These two facets of the evaluation are included in Table 5.7, in rows "Users vs. *AllMusic.com*" and "ATR vs. Users", respectively. Values for $H@3$, $H@5$, RP and MRR are summarized in Table 5.7 and a graphical comparison of these values and the results of the automatic evaluation can be found in Figure 5.5. Below we describe the details of the two investigations we performed:

Users vs. *AllMusic.com*. As mentioned already, with this analysis we would like to see the performance of the users compared to that one of the music

[11] Facebook. http://www.facebook.com

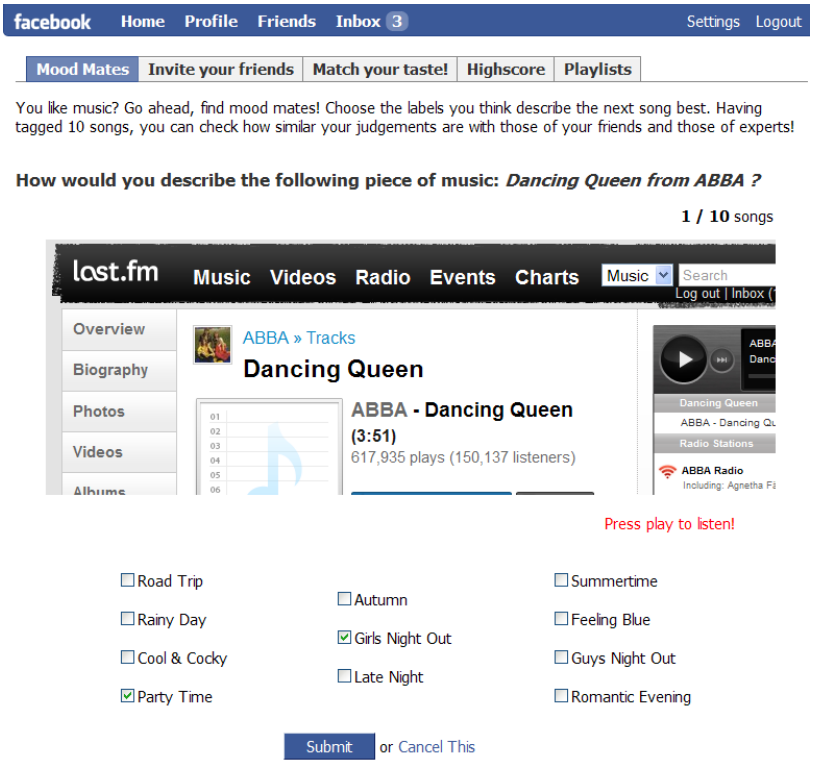

Figure 5.6 Mood Mates! Facebook application

experts from *AllMusic.com*. Given the restriction that the set of songs on which the evaluation was performed was different than in the case of the automatic evaluation (see Section 5.3.4), we re-do the evaluation of our method also on this set of tracks. The same measures are computed for the assignments of the users. For the $H@5$ measure, we cannot provide any significant results, as the number of songs tagged by users with five labels was too small. For the case of MRR, this can also not be computed, as we do not distinguish between the order of theme assignments of the users. We are thus left with 2 measures to compare our performances against those of the users: $H@3$ and RP. In terms of $R-precision$, our method (ATR) shows 108% improvement over the users, at a statistically highly significant level (paired t-Test with $p \ll 0.01$), while $H@3$ has a gain of 16%.

ATR vs. Users. With the second analysis we perform, we aim at letting the users evaluate the theme recommendations we provide. Compared to the previous study, the ground truth is not *AllMusic.com* any more, but the assignments of the participants in our survey. For our approach it is important both to provide accurate recommendations (*i.e.* high overlap with the experts' assignments), as well as recommendations relevant for users annotating the music tracks. As we can see from the results included in Table 5.7, also when compared to the "user ground truth" our method performs very well: $H@3$ has a value of 0.65 and for $H@5$ we achieve 0.83.

	H@3	H@5	RP	MRR
ATR vs. *AllMusic.com*	0.74	0.85	0.45	0.63
Users vs. *AllMusic.com*	0.64	NA	0.22	NA
ATR vs. Users	0.65	0.83	0.35	0.57

Table 5.7 Theme tags: Users vs. *AllMusic.com* vs. ATR

The results show that our method performs well also with respect to the user assignments. The fact that the users perform quite bad compared to the AllMusic experts, but our method (ATR) performs well both compared to the users and to the experts, indicates that our method provides theme labels that are easier to recognize by users than the labels assigned by AllMusic experts.

5.4 Identifying Potential Music Hits

Automatic prediction of hit songs is currently turning into a hot topic, the main reasons being: (1) the money that music record companies are willing to pay for such services and (2) the increased computing power allowing the development of powerful tools for solving this problem. The subject has been approached in many ways and some companies, such as Polyphonic HMI, make good money from it.

The benefits of being able to predict which songs are likely to become hits is various and is of big interest for both music industry and artists, as well as for listeners. In their attempt to release only profitable music, producers may want to have an indication of the potential of the music songs they will work with. Artists can profit from the results of such techniques by identifying the most suitable markets for their songs, music lovers' niches and by choosing the best channels and targets. Last but not least, normal music listeners can enjoy good music as a benefit of accurate hit predictions on a daily basis – radio stations can use such methods in order to improve their program by playing only songs which are highly likely hits.

Most previous attempts to identify hit songs have focused on intrinsic characteristics of songs, such as lyrics and audio features. In the prevailing view it is all about musical quality, so the task is to reveal the audience's preferences about music - *e.g.* by finding the similarity to what they liked before. However, it is often neglected that people are not independently deciding on what they like, but rather they like what they think other people may also like [Wat07]. Despite 'intrinsic quality' success seems also to depend on the already known or assumed popularity, *i.e.* we find a rich get richer effect (aka preferential attachment or cumulative advantage). Thus, subjective opinions of a few early-arriving individuals account for hit potential as well, introducing a certain amount of randomness like in the famous 'butterfly effect' [Lor63].

As social networks become more and more popular and some specialize on certain

topics, information about users' music tastes becomes available and easy to exploit. Web 2.0 applications such as *Flickr*, *Del.icio.us* and *Last.fm* are well-known for fast-growing online data production via their network effects. The wisdom of the crowds has become a famous notion of the collective intelligence manifesting itself in such collaborative tagging systems. Here, people organize and share resources by providing valuable semantic annotations. Especially for multimedia resources, accurate annotations are extremely useful, as these additional textual descriptions can be used to support multimedia retrieval. Most important in our case, these networks set and identify trends and hot topics.

We propose a method for predicting the success of music tracks by exploiting social interactions and annotations, and without relying on any intrinsic characteristics of the tracks. We predict the potential of music tracks for becoming hits by directly using data mined from a music social network (*Last.fm*) and the relationship between tracks, artists and albums. The social annotations and interactions enable both measuring similarity (*i.e.* intrinsic quality) of songs and finding those critical early-stage effects of cumulative advantage. Our approach requires only the social data corresponding to a track's first week life in *Last.fm*, in order to be able to make good predictions about its potential and future evolution[12].

5.4.1 Datasets

Last.fm

The method we propose for predicting music hits relies on external social information extracted from the popular music portal, *Last.fm*. For gathering the *Last.fm* data, we start from a previous crawl, described in Section 3.3.1 and also included in [FNP]. For the purpose of the present study, we also needed information regarding the weekly charts of the *Last.fm* users. One of the most popular features of *Last.fm* user profiling is the weekly generation and archiving of detailed personal music charts and statistics. Users have several different charts available, including Top Artist, Top Tracks and Top Albums. Each of these charts is based on the actual number of times people listened to the track, album or artist. Similar global charts are also available and these are created based on the total number of individual listeners. For gathering this information, we started from the initial set of 12,193 crawled users (included in the initial *Last.fm* crawl from May 2007) and for all of them we downloaded all their available weekly charts. For this task we made use of the Audioscrobbler[13] web services. As not all of the 12,193 users from our initial set have been active since May 2007, we could gather charts for only 10,128 of them. A weekly chart consists of a list of songs that the user has listened during that particular week. The weekly

[12] *Last.fm* offers to artists the possibility to upload their own music to the portal (http://www.last.fm/uploadmusic?accountType=artist).

[13] Audioscrobbler. http://www.audioscrobbler.net

charts we could gather span over 164 weeks and our final data collection consisted of 210,350 tracks, played by 37,585 unique artists. 193,523 unique tags are associated with the tracks, 163,483 of these tags occurring as well along with artists' names.

The distribution of charts per user fits a power law curve, as depicted in Figure 5.7(a). For the set of songs identified in the users' charts, the distribution of the number of times played per user follows a power law distribution as well (Figure 5.7(b)). Following directly from this finding, the distribution of artists occurring in the users' chart lists presents the same characteristics (Figure 5.7(c)). The sharp drop at the end of all curves is due to the crawling method and sampling.

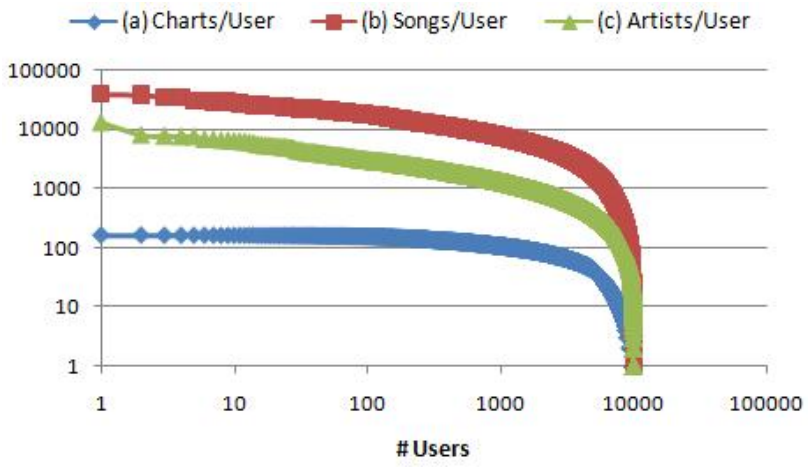

Figure 5.7 Log scale distributions of: (a) Charts/User; (b) Nr. of artists/User; (c) Nr. or songs/User

Billboard Charts

For being able to asses the quality of our predictions, we also needed a good ground-truth dataset. The most suitable for our purposes was the data exposed by *Billboard.com*. Billboard is a weekly American magazine devoted to the music industry, which maintains several internationally recognized music charts that track the most popular songs and albums in various categories on a weekly basis[14]. Billboard utilizes a system called Nielsen SoundScan to track sales of singles, albums, videos and DVDs, so that it can register sales when the product is purchased at the cash register of SoundScan-enabled stores. Moreover, it also utilizes a system called Broadcast Data Systems (BDS) for tracking radio airplays. Each song has a musical "fingerprint" which, when played on a radio station that is contracted to use BDS, is detected. This information is added up every week among all radio stations to determine airplay points. Arbitron statistics are also factored in to give "weight" to airplays based on audience size and time-of-day.

[14]Billboard Charts. http://www.billboard.biz/bbbiz/index.jsp

Billboard produces charts for music albums and singles, as well as a large number of specializations of those, depending on several characteristic like genres, countries, compilations, *etc.* – each of these Billboard charts using this basic formula. What separates the charts is what stations or stores each chart uses – each musical genre having a core audience or retail group. The charts express thus music popularity in means of sells, and other factors, and are considered the best measure of success for a track in an universal and heterogeneous environment. The Billboard charts are not biased towards any community and we believe that they express the popularity of a track in a useful, objective way. They are released weekly as .html pages and represent the top tracks of the previous week. Every chart has associated a name, an issue date, and stores information about the success of the songs in form of rank, artist name and album/track name. Moreover, each chart entry has a previous week rank, as well as a highest rank field – *i.e.* the highest Billboard position ever reached by that song.

There are 70 different charts available for singles and 57 different ones for albums and a detailed list can be found at [Bila] and [Bilb], for albums and singles, respectively. We collect all these Billboard charts and aggregate the information, the resulting charts thus spanning over a range of almost 50 years, namely between August 1958 and April 2008. In total, the aggregated Billboard single chart contained 1,563,615 entries, 68,382 of them being unique songs. With respect to albums, the aggregated chart had 1,200,156 entries and among those, only 49,961 proved to be different albums. The big amount of duplicates comes from the fact that many of the songs or albums occur in the charts corresponding to several weeks, as well as in different types of charts, *e.g.* "European Hot 100 Singles", "The Billboard Hot 100", *etc.* In Figure 5.8 we present the distribution of music tracks across several top rank ranges in the Billboard charts. Similarly, Figure 5.9 depicts the distribution of albums over several rank ranges in Billboard.

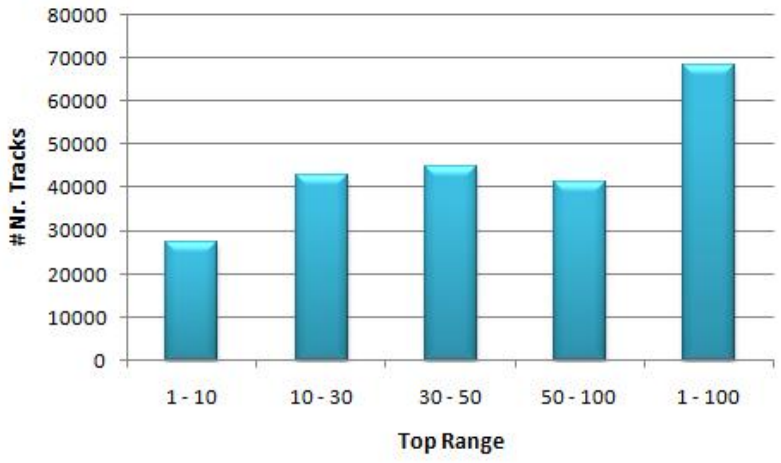

Figure 5.8 Tracks' distribution over several top rank ranges

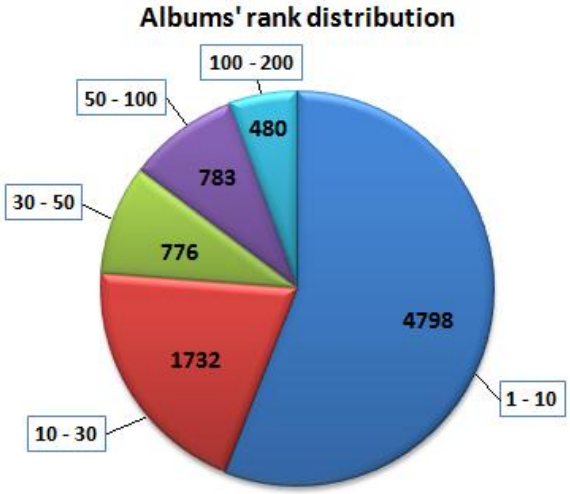

Figure 5.9 Albums' distribution over several top rank ranges

The final set of tracks and the corresponding information around these tracks (artist, album, Billboard rank, *etc.*) is represented by the intersection of the set of unique tracks gathered from the *Last.fm* users' weekly charts and the set of tracks included in the Billboard charts. This intersection resulted in 50,555 unique music songs, on which we will perform our experiments.

5.4.2 Predicting Music Hits

As already discussed, existing attempts to automatically identify hit-songs, rely mostly on finding specific acoustic patterns in the songs and/or specific themes by analyzing the lyrics of the music tracks. Such approaches would then never predict a purely instrumental track as a potential hit, nor would they predict as hit a song promoting new and revolutionary sounds. With the method we propose in the present paper we tackle exactly these two shortcomings. Our solution applies neither low level feature extraction and analysis on the music tracks, nor data mining techniques on the songs' lyrics. Instead, we make use of the social information around the tracks, which we gather from the popular music portal *Last.fm*. This information is processed and transformed into a list of features, which is fed to a classifier for training it to discover potentially successful songs. The approach we propose relies on the following assumptions:

- The initial popularity (*i.e.* the popularity among listeners after only one week after the upload) of a track is indicative of its future success.
- Artists interpreting the tracks have a direct influence on the future success of the songs.

- Previous albums of the same artist have a direct influence on the future success of the songs.
- The popularity of other tracks produced by the same artist and included on the albums we consider have also an impact on the future success of the song.

With these hypotheses fitting perfectly to the principles of preferential attachment/cumulative advantage, we now proceed describing the details of our music hit prediction algorithm based on social media data.

Feature selection

The features used for training the classifiers are chosen such that the assumptions listed above are supported. It is thus natural to build a model where the main entities correspond to the interpreting *Artist*, previous popular *Albums* of the same artist and *Tracks* included on the albums considered. Moreover, each of these entities has associated a set of attributes, which are as well taken as input features for our classifiers. In Figure 5.10 we present the complete set of features considered.

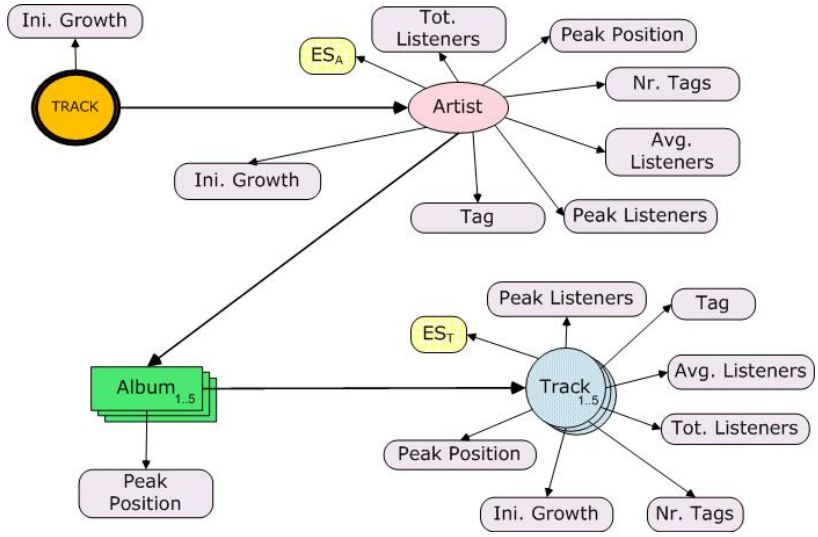

Figure 5.10 Features used for training the classifiers

All entities and their associated attributes related to a particular track, for which we would like to predict whether it will be a hit or not, form a tree having the $TRACK$ as root. Each of the features can be reached by starting from the root of the feature-tree and following the corresponding branches. We now discuss in detail the main feature entities and their associated attributes composing the feature tree.

Artist-relevant Features.

Artists, as the performers of the songs we make predictions for, are likely to have an influence on their hit potential. Usually, artist entities have associated a set of tags assigned by *Last.fm* users – we consider the top 5 most used tags, $Tag_{1..5}$. In case the artist does not yet have 5 tags, we exploit as many as available. Besides, we also include the total number of tags available for an artist, $Nr.\ Tags$, as well as its overall number of listeners, $Tot.\ Listeners$. $Ini.\ Growth$ represents the number of listeners for this artist during the first week it appeared in *Last.fm* charts (Note that these are *Last.fm* user charts.). The higher this number, the bigger the probability that this artist is quite popular and his future songs will become hits with a high probability. The *Peak Listeners* feature measures the maximum number of listeners over all *Last.fm* user charts. With $Avg.\ Listeners$ we capture the average number of listeners over all *Last.fm* user charts and the value is computed as:

$$Avg.\ Listeners = \frac{Tot.\ Listeners}{\#weeks\ in\ Last.fm\ user\ charts} \tag{5.2}$$

The *Peak Position* represents the highest Billboard position this artist reached so far (for new artists, this value will not be known).

An *Artist* is directly connected to an *Album*-entity, since the performer might have produced several albums already. Thus, we also include as features the artist's top-5 albums, or as many as currently available.

Track-relevant Features.

In the model presented in Figure 5.10, containing the features used for training the classifiers, the *Track* entity occurs twice. It is important to distinguish between the two different instances: $TRACK$ represents the song for which we aim to automatically predict whether it will belong to the class "HIT" or "NHIT", which is also the root of the tree resulted from the complete set of features. Beside this, we also consider top-5 tracks, $Track_{1..5}$, appearing on the albums we include as feature for the given $TRACK$.

For $TRACK$, the track for which we want to make the predictions, only the $Ini.\ Growth$ feature is considered (the maximum number of *Last.fm* listeners after the first week this song appeared on the *Last.fm* portal). The artist of the track, *Artist* represents an entity directly connected to $TRACK$ and for the case that the song has more authors (*e.g.* Madonna featuring Justin Timberlake) we consider only the first artist.

For $Track_{1..5}$, the tracks associated to other albums of the same artist, we include as feature the overall number of listeners on *Last.fm*, $Tot.\ Listeners$, as a strong indicator of its popularity. The tags given by the *Last.fm* users to a track are as well good popularity indicators. We consider the top-5 tags, $Tag_{1..5}$, or like in the case of the artists, if there are less than 5 tags, we consider as many as available. $Peak\ Position$, $Avg.\ Listeners$, $Peak\ Listeners$ and $Ini.\ Growth$ have the same meaning as the corresponding artist-related features.

Album-relevant Features.

Similar to the case of artist-like entities, albums also have associated a series of features: their popularity can be measured based on the highest position reached in Billboard, the *Peak Position* feature. Since for some artists the previously released albums can be quite many, we include only the top-5 albums which reached positions in the Billboard charts. Besides, from each album we also consider the top-5 Billboard listed tracks, $Track_{i_{1..5}}$.

Additional Features.

In addition to the direct features discussed above, we also extract some implicit features for the artist and track entities. We associate *ES*-Entity Scores features, as a combination of the entities' Billboard top reached position and their HITS scores [Kle99] – computed by applying HITS on a graph using artists, tracks and tags as nodes. Given an artist A, a track T and a tag TG, we create links as follows:

- From A to T, if track T is played by artist A;
- From T to TG, if track T has been tagged with tag TG;
- From A to TG, if artist A has been tagged with tag TG.

On the resulted graph we apply the HITS algorithm and compute the corresponding hub and authority scores. We present below the formulas for computing the HITS scores for artists, HS_A and tracks entities, HS_T:

$$HS_{A|T} = \begin{cases} 0, \text{ if } hubS_{A|T} == 0 \wedge authS_{A|T} == 0; \\ authS_{A|T}, \text{ if } hubS_{A|T} == 0 \wedge authS_{A|T}! = 0; \\ hubS_{A|T}, \text{ if } hubS_{A|T}! = 0 \wedge authS_{A|T} == 0; \\ authS_{A|T} \cdot hubS_{A|T}, \text{ otherwise.} \end{cases} \tag{5.3}$$

$hubS_{A|T}$ and $authS_{A|T}$ represent the hub and authority scores of the artist and track (represented as subscripts A or T respectively).

The final Entity Scores (ES) will be based both on the outcome of calculating the HITS scores and the corresponding best positioning in any of the Billboard charts ever. This score will give an estimation of the popularity of certain artists and tracks in relation to the tags used, between themselves and in the opinion of a recognized authority in the domain, as the Billboard charts are. The formula for computing ES_A and ES_T, the entity scores for artists and tracks is given below:

$$ES_{A|T} = \begin{cases} \frac{1}{1000} \cdot HS_{A|T}, \text{ if } PeakPos_{A|T} \text{ is missing;} \\ \\ \frac{1}{PeakPos_{A|T}} \cdot HS_{A|T}, \text{ otherwise.} \end{cases} \tag{5.4}$$

$PeakPos_{A|T}$ represents the best reached position by the artist or track in all considered Billboard charts. If these entities do not occur in any of the charts (they never

got that successful as to be included in the music tops), we consider a very large number (1000) to substitute their missing Billboard rank. The inverse of this number or of the best Billboard position is considered for the computation of the final Entity Score (see Equation 5.4). The resulting $ES_{A|T}$ scores for artists and tracks will be used as features for our music hits prediction algorithm. They will be attached to the corresponding entities depicted in the feature graph from Figure 5.10.

Music Hit Prediction Algorithm

The core of our music hit prediction method is a classifier trained on the *Billboard.com* ground truth and using as features social media data extracted from *Last.fm* or inferred from it. We experiment with a number of different classifiers (Support Vector Machines, Naïve Bayes, Bayesian Networks and Decision Trees) and for building the classifiers we use the corresponding implementations available in the open source machine learning library Weka[15]. Given the three hypotheses mentioned above, the classifiers learn a model from a training set of data. Once the model is learned, it can be applied to any unseen data from *Last.fm* and predict whether the corresponding songs have the potential of becoming hits or not. Below we present the main steps of our music hit prediction algorithm.

Alg. 1. Music hit prediction

1: Split song set S_{total} into
S_{train} = songs used for training the classifier
S_{test} = songs used for testing the hit predictions
2: Select features for training the classifier
2a: For each song $s_i \in S_{train}$
2b: Create feature vector $F(s_i) = \{f_j | f_j \in FS\}$, where
FS = feature set from all songs, computed as:
$FS = FS(s_i) \cup FS(Artist) \cup FS(Album_{1..5}) \cup FS(Track_{1..5})$
$FS(s_i) = \{Ini.\ Growth\}$
$FS(Artist) = \{Ini.\ Growth, Tag_1, .., Tag_5, Peak\ Listeners,$
$Avg.\ Listeners, Nr.\ Tags, Peak\ Position, ES_A\}$
$FS(Album) = \{Peak\ Position\}$
$FS(Track) = \{Ini.\ Growth, Tag_1, .., Tag_5, Peak\ Listeners,$
$Avg.\ Listeners, Nr.\ Tags, Peak\ Position, ES_T\}$
3: Train classifiers on S_{train} using $\{F(s_i); s_i \in S_{train}\}$
4: For each song $s_i \in S_{test}$
4a: Compute probabilities $p(HIT|s_i)$ and $p(NHIT|s_i)$
4c: $PMAX = max(p(HIT|s_i), p(NHIT|s_i))$
4d: $Class = \begin{cases} HIT, \text{if } PMAX = p(HIT|s_i); \\ NHIT, \text{if } PMAX = p(NHIT|s_i). \end{cases}$

The set of songs described in the Datasets Section (Section 5.4.1) is split into two partitions: one partition for training and one for testing the classifiers (step 1). Then, for the songs in the training set, we build the set of corresponding features (step 2) according to the attributes attached to the main entities (artist, albums, tracks) as

[15] Weka. http://www.cs.waikato.ac.nz/~ml/weka

depicted in Figure 5.10. The classifier is trained on the resulting set of features and a model is learned from it (step 3). After this step, the model is applied to all songs from the test data and a prediction is made (step 4). In the next section we present the evaluation of our algorithm.

5.4.3 Experiments and Results

For measuring the performance of our prediction algorithm we will use the following metrics:

- Accuracy (Acc) – Statistical measure of how well the classifier performs overall;
- Precision (P) – Probability for items labeled as class C of indeed belonging to C;
- Recall (R) – Probability of all items belonging to class C of being labeled as C;
- F1-measure (F1) – Weighted harmonic mean of Precision and Recall;
- Area under ROC (AUC) – Area under the Receiver Operating Characteristic (ROC) curve obtained by plotting the fraction of true positives vs. the fraction of false positives.

We experiment with several multi-class classifiers: Support Vector Machines, Naïve Bayes, Decision Trees and Bayesian Networks with 1 or 2 parents, but only the best results are presented – this was the case of Bayesian Networks with 2 parents. We train classifiers for several rank ranges, such that the partitioning of the data satisfies the following: For hit class $1-1$, we consider as hit songs only those tracks which have reached top-1 in Billboard charts. All other songs starting with the second position in Billboard are considered non-hits. Similarly, other hit rank ranges are considered: $1-3$ (*i.e.* tracks which have reached top-3 Billboard positions are regarded hits, while the rest, starting from position 4, are non-hits), $1-5$, $1-10$, $1-20$, $1-30$, $1-40$ and $1-50$. The number of hit and non-hit instances is approximately the same for all classifiers. We select as many songs as available from the rank ranges considered as hits. For non-hits, we randomly pick about the same number of songs from the set of music tracks with Billboard positions greater than the right margin of the hit class or from the set of tracks not appearing at all in the Billboard charts (*i.e.* "clear" non-hits). We summarize in Table 5.9 the resulting number of instances for each of the hits' rank ranges.

Each classifier is trained and tested on the total set of instances (both hits and non-hits), corresponding to each of the hit class ranges. The results of the classification are evaluated in terms of Accuracy, Precision, Recall, F1-measure and AUC. In Table 5.8 we present the averaged results of the 10-fold cross validation tests.

Hits' Rank Range	Acc[%]	Hits				Non-Hits			
		P	R	F1	AUC	P	R	F1	AUC
1 – 1	81.31	0.788	0.858	0.821	0.883	0.844	0.768	0.804	0.883
1 – 3	79.73	0.768	0.852	0.808	0.875	0.833	0.742	0.785	0.875
1 – 5	79.57	0.765	0.854	0.807	0.871	0.834	0.737	0.783	0.87
1 – 10	79.24	0.771	0.835	0.801	0.857	0.818	0.75	0.783	0.856
1 – 20	75.84	0.804	0.688	0.741	0.848	0.724	0.83	0.773	0.848
1 – 30	75.87	0.808	0.684	0.741	0.85	0.722	0.835	0.774	0.85
1 – 40	75.28	0.802	0.679	0.735	0.843	0.716	0.829	0.768	0.843
1 – 50	75.19	0.803	0.676	0.734	0.84	0.714	0.83	0.768	0.84

Table 5.8 Classifiers' evaluation for predicting Hits/Non-Hits, considering different rank intervals for the hit-classes

Hits' Rank Range	# Hit Inst.	# Non-Hit Inst.
1 – 1	2,335	2,331
1 – 3	3,607	3,594
1 – 5	4,354	4,339
1 – 10	5,553	5,515
1 – 20	7,016	6,913
1 – 30	8,035	7,897
1 – 40	8,744	8,538
1 – 50	9,024	8,807

Table 5.9 Distribution of instance numbers for Hits/Non-Hits for different hit-class ranges

As observed from Table 5.8, the best results are obtained for the classifier built for detecting top-1 music hits. For this case, we obtain a value of 0.883 for the AUC measure, 0.788 precision and 0.858 recall for hits, while the overall accuracy is 81.31%. In [DL05] the authors reported AUC values of 0.69 for the best performing classifiers, trained to recognize top-1 hits from charts in Unites States, UK and Australia. Having similar datasets' sizes and song sets with no bias on any particular music genre (though the tracks might be different), our results for class $1-1$ are comparable with the ones reported by [DL05]. However, our approach performs better, providing $\approx 28\%$ improvement in terms of AUC values. It has been argued that AUC values between $0.5 - 0.7$ are an indicator of low accuracy, while AUC values between $0.7 - 0.9$ indicate good accuracy [FBJ03].

For all other classifiers, the results present as well characteristics which indicate good classification accuracy. In terms of AUC values, the performance is a bit worse than for the very restrictive case of hits taken only from top-1 Billboard charts (class $1-1$). The main reason for this is the fact that as we increase the rank range for what we call hits, the tracks begin to have a more heterogeneous set of features making it more difficult for classifiers to distinguish the correct hits from the rest of the songs. However, as we increase the interval ranges, precision improves in the detriment of recall, the best value being achieved for hit predictions from the interval $1 - 30$.

For the scenarios we consider, precision is actually more important than recall: a music label would be interested in promoting as far as possible only those music tracks which definitely have the potential of becoming hits; most radio stations try to play only music tracks which are already popular or already in top positions of music charts. Such radio stations would be thus more concerned about only supplying their audience with music hits, rather than being sure that they cover all hits. If some "missed" music tracks turn eventually to chart-hits, the radio stations can still introduce them in their airplay programs. The main advantage of relying on such an approach is the fact that they can easily identify new and fresh sounds after just one week of letting the song "in the hands" of the *Last.fm* users.

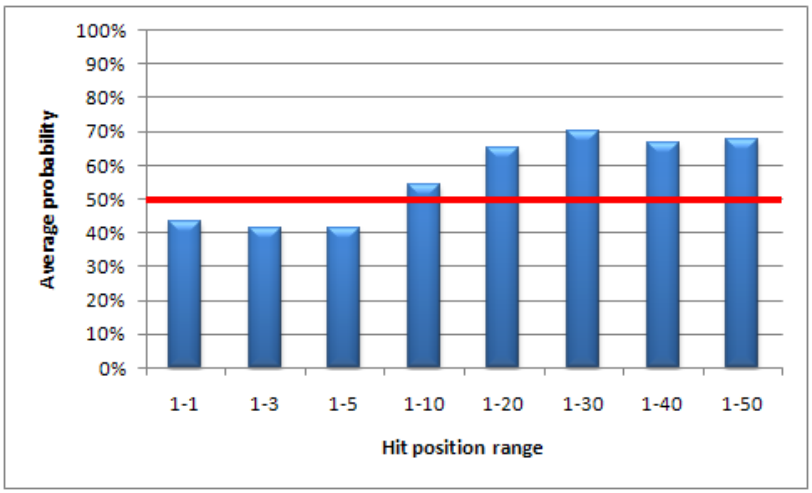

Figure 5.11 Classification probability for chart position 7 averaged over 100 songs

In addition to the experiments described above, we also tested the accuracy of the built classifiers on a concrete scenario: we created a set of 100 songs, all having reached position-7 in Billboard, as their best rank. The resulted set of tracks was afterwards used for testing all classifiers (the set of 100 rank-7 songs was removed from all training sets of all classifiers). In Figure 5.11 we present the average probabilities for the 100 rank-7 tracks as assigned by the different classifiers and indicating the likelihood of the tracks to belong to the particular hit range class. The thick line at the 50% average probability corresponds to random class assignment. We observe that classifiers corresponding to classes $1-1$, $1-3$ and $1-5$ all have probabilities below the threshold, which is perfectly correct since all tested tracks have rank position 7. Starting with the classifier for the range $1-10$, the average probabilities are showing the track position to be included in the respective intervals.

5.5 Identifying Landmark Pictures

Given the current widespread usage of digital photography, we observe users willing to share their photos and experiences within social platforms, such as *Flickr*. As *Flickr* already contains millions of photos, the tasks of searching and navigating photos of interest become very difficult. To simplify these tasks, users adopted tagging, adding to each photo a set of freely chosen keywords. Still, simple tag matching does not give satisfactory results for particularly complex search tasks. Given that digital photography and social photo-sharing services continue to grow rapidly, these tasks of effective photo search and navigation are getting more and more attention from the research community. One of such tasks is creating a photo summary of landmarks for a city, which is referred in the literature as *landmark finding* problem. The World Explorer application [ANNY07] is the current state-of-art system which provides a landmark finding solution for *Flickr*. The system has a reasonable performance, but it only works with geo-tagged photos (supplied with geographical coordinates). The problem is that many interesting places around the world are still represented by photos without geo-tags and their landmarks cannot be found using World Explorer. We propose to exploit the tagging features and social *Flickr* groups to train a classifier with minimum effort.

Recognizing a landmark on a photo is a hard task: First, content-based image analysis has very limited capabilities to solve this problem in general, given that photos are taken in different light and weather conditions, from different viewpoints and angles. Second, text-based or tag-based methods are much more appropriate for this task, but they do not have extra information if a tag represents a landmark or a family photo taken in a city. We propose to obtain this extra information from social groups in which users are involved. Nowadays *Flickr* is enriched with specific photo groups related to landmarks, cars and other types of objects and themes, which can be used to distinguish the main topic of the photo. With our approach we thus aim at exploiting exactly this valuable type of information.

Our method contains two main parts: First, we exploit tags and social *Flickr* groups to train a classifier to identify landmark photos and tags. The method requires minimum human efforts, by manually providing input links to relevant *Flickr* groups. Afterwards, the system automatically trains a classifier based on the data retrieved from the specified *Flickr* groups. In the second part, our method ranks all suggested relevant tags by their representativeness of a landmark.

This approach is also generalizable to other problems such as car finding, mobile phone finding, *etc.* However, due to the high cost of user studies, in this paper we test the performance of our method on the landmarks only. To our best knowledge, our solution is the first to solve the landmark finding problem based on photo communities information. The current method is limited to users' tags and social *Flickr* groups and does not make use of low level image features or geo-tags. The user study we conducted for evaluating the proposed algorithm shows that our approach

outperforms World Explorer even on geo-tagged photos.

5.5.1 Formalizations and Problem Statement

For the rest of the paper we will consider that the landmark finding application has to automatically create a summary of *Flickr* photos, giving a comprehensive overview of landmarks at some location of interest. We will decompose this task into several sub-problems, as presented in Figure 5.12.

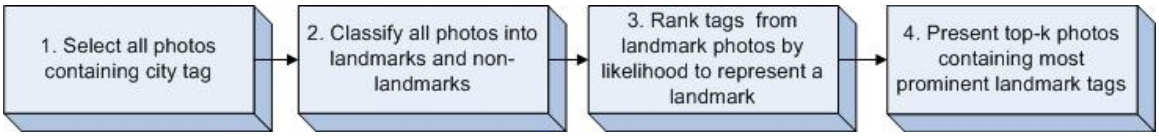

Figure 5.12 Decomposition of Landmark Finding Problem

The first step consists of selecting a set of photos related to a particular city. Since we do not consider geo-tagged photos, we rely on a simple heuristic of having the city name as a tag associated with a photo. This way we may miss many relevant photos, but for our task it is not a problem, since we still get a lot more photos than we need for a summary generation. In the second step all collected photos are automatically classified as either landmarks or non-landmarks. It is important to understand that at this point, we do not have a summary of city landmarks. We have just a list of pictures with or without landmarks, according to the classifier. What we want to achieve is a list of names representing city landmarks and based on these names create a comprehensive city landmark summary. In the third step, tags of the photos classified as landmarks are ranked according to their likelihood of representing city sights. Once a ranking score is available for all tags in the set, in the fourth step we select top-k most representative tags. For each of these k tags we retrieve the top-1 *Flickr* photo which has as tags both the name of the city, as well as the landmark tag. We consider that photos should not contain several representative tags at once, since we aim at showing a single landmark with each photo. For returning the set of top-1 *Flickr* pictures satisfying the conditions described above we make use of the *Flickr* API[16] for tag-based search and sort the pictures by relevance. As the steps 1 and 4 presented in Figure 5.12 are quite simple, therefore we will not focus on them. In the following sections we discuss in detail the sub-problems of classification and tag ranking.

For understanding the algorithms presented in section 5.5.2 we need to first introduce a number of formalizations and definitions. In the following definitions U represents the set of users, T stands for the set of tags, R is the set of resources and $Y \subseteq U \times T \times R$ is ternary relation over U, T and R representing a user's tag assignment.

[16] http://www.flickr.com/services/api

Definition 1. Tag Frequency Normalization (TF) We define the number of times a tag t appears with a resource r as frequency of the tag t with resource r, $f_r(t)$:

$$f_r(t) =\mid \{(u,t,r) \in Y, u \in U\} \mid \tag{5.5}$$

Normalized tag frequency $TF_r(t)$ of a tag t in a resource r is then computed as follows:

$$TF_r(t) = \frac{f_r(t)}{\sum f_r(t')}, (u,t,r) \in Y, (u,t',r) \in Y, t' \in T, u \in U, \tag{5.6}$$

Definition 2. Inverse Resource Frequency (IRF)

Inverse Resource Frequency, like Inverse Document Frequency in IR, is computed as below:

$$IRF(t) = log\left(\frac{\mid R \mid}{\mid \{(t,r), u \in U, r \in R, (u,t,r) \in Y\} \mid}\right) \tag{5.7}$$

Definition 3. Inverse User Frequency (IUF)

Similar to *IRF*, we define Inverse User Frequency (*IUF*), which is formally defined *IUF* as follows:

$$IUF(t) = log\left(\frac{\mid U \mid}{\mid \{(u,t), u \in U, r \in R, (u,t,r) \in Y\} \mid}\right) \tag{5.8}$$

5.5.2 Landmark Finding Methodology

In the following we present the details of the main sub-problems composing our landmark finding method. We focus on step 2, classification of photos into landmarks and non-landmarks, and step 3, selecting the most representative landmark tags.

Landmark Classification

From the set of pictures containing city tag, we want to select photos representing landmarks. For this task we make use of a SVM binary classifier [Vap99][17], in particular its SVMLight implementation (see [Joa02]). For every picture we create a feature vector based on the tags which were used to annotate it and the SVM classifier assigns each photo to either "landmark" or "non-landmark" category. We assign weights to the tags in the feature vectors based on the usage of tags among resources and users, presented below. Formally we define a feature vector for a photo r as following:

$$F(r) = [w(t_1,r), w(t_2,r), \cdots, w(t_{|T|},r)], u \in U, t \in T \tag{5.9}$$

where $w(t,r)$ is defined as:

$$tf\_irf(t,r) = TF_r(t) \cdot IRF(t) \tag{5.10}$$

[17]While in general it is possible to apply any other classifier, we rather try to test the hypothesis about the applicability of tags for landmark photos identification.

Several weighting schemes have been tested for the feature vector, however, the combination given by Eq. 5.10 provided best results.

One of the main challenges for SVM or any machine learning technique is to create a good training set. Once a model is created based on the labeled data from the training set, the SVM can classify unseen examples based on the model. Our hypothesis is that some of the *Flickr* groups like "Landmarks around the world" can serve as positive examples, while arbitrary general groups, like "Birds" or "Airplanes" represent negative examples.

The idea to use *Flickr* groups as training data is quite simple and can be used for any arbitrary photo classification task beyond the landmark finding problem. If a relevant group of photos exists on *Flickr*, one can use it as a training data to find more photos on the same topic within *Flickr*. For example "CAR [directory]" or "Mobile Phones" groups can be helpful for finding thousands of car and mobile photos. Nevertheless, applicability of *Flickr* groups for such tasks needs to be studied with additional experiments.

Measuring Tags' Representativeness

Once we have selected a set of city photos and filtered only landmark-related ones, the third step consists of ranking all tags by how well they represent landmarks. What we would like to achieve is a ranked set of tags representing landmarks specific to a particular city. For example, one can intuitively mark the tag "sky" as a poor evidence of landmark, "bridge" is somehow better and "goldenbridge" is the most promising one. However, we need to be able to generalize this over the whole set of *Flickr* tags for finding the most probable tags as being landmark annotations. Several intuitions for discovering the most representative tags were presented in [ANNY07]. We consider the following properties of tags for computing tags' representativeness as landmarks: global properties consider the complete dataset, while local properties are related to the tags representing landmarks of a particular city.

When looking at the whole dataset, we would like to give low scores to common tags. The assumption is that representative landmark tags appear relatively often along with landmark photos, but are not very common among the rest of the collection. Let us consider R the set of all photos (both landmark and non-landmark related ones), and T the associated set of tags. Supporting this first assumption, we compute IRF (Eq. 5.7) of the considered tag. If a tag is frequently used to tag photos in the dataset, it has a low $IRF_{R,T}(t)$[18] value and vice versa. Similarly, if a tag is globally very common amongst users, it must be scored low. This is achieved by computing IUF, $IUF_{R,T}(t)$ (Eq. 5.8).

After defining global scoring factors, we come to local measures computed on a part of the collection with landmark photos only. When considering the dataset

[18] Computation is relative to R and T

containing only pictures associated to a particular city and classified as landmarks, our assumption is that common tags should be scored high. Let us represent the set of landmark-related photos selected for a city as R_c and the corresponding tag set as T_c. If a tag is common among the photos for a particular city, probably this tag represents some feature of the city, *e.g.* some museum, or an old and famous building. Let $nrt_c(t)$ be a number of times a tag t appears within landmark photos for a city c. Then we can compute normalized *City Tag Frequency*, $CTF(t)$, as follows (Eq.5.11):

$$CTF(t) = \frac{nrt_c(t)}{MAX(nrt_c(t'))}, t, t' \in T_c \tag{5.11}$$

Similarly, if a tag is used frequently by users, then it is probably a feature of the city. Let $nut_c(t)$ be the number of users using a tag t for the landmark photos for a city c. We compute the normalized *City User Tag Frequency*, $CUTF$, using (Eq.5.12):

$$CUTF(t) = \frac{nut_c(t)}{MAX(nut_c(t'))}, t, t' \in T_c \tag{5.12}$$

The decision values returned by the SVM classifier against the classified photos represent a confidence measure of the classification. Let d_r be the decision value for the photo r and let R_t be all the resources associated with a tag t. The confidence value $CONF(t)$ for the tag t is calculated as:

$$CONF(t) = log\left(\sum_{r \in R_t} d_r\right) \tag{5.13}$$

We combine all the above mentioned factors that affect the ranking of the tags and compute a representativeness score for each tag t occurring along with the resources classified as landmarks of a city c. The representative score of each tag for a city c is computed as follows:

$$SCORE(t) = IRF_{R,T}(t) \cdot IUF_{R,T}(t) \cdot CTF(t) \cdot CUTF(t) \cdot CONF(t), t \in T_c \tag{5.14}$$

5.5.3 Experiments and Results

Given the methodology described in Section 5.5.2, we now proceed with the description of the evaluation we performed.

Datasets

For our experiments we needed two sets of data: a training set and a test set of data.

Training Data (DS_1). The training dataset was used for training the landmark vs. non-landmark classifier. DS_1 was constructed by downloading 430,282

photos from several *Flickr* groups, uploaded by 57,581 different users. For positive examples we manually picked several groups, such as "Landmarks", "Landmarks around the world", "City Landmarks", *etc.* As negative examples we used groups like "Airplanes", "Birds", "Cars", "Mobile Phones", *etc.* The dataset thus created contains 14,729 positive examples (related to landmark groups) and 415,553 negative examples (related to general groups). None of these 430,282 photos was included in the test dataset. This is real-world data, so "positive groups" might also contain some non-landmark photos and vice versa. However, no additional noise reduction technique has been applied.

Test Data (DS_2). This dataset consists of pictures corresponding to 50 randomly picked cities (for which World Explorer [ANNY07] has at least 10 landmark tags), 60% European ones and the rest of 40% representing Asian, North-, South-American and Australian cities. We downloaded 4,000 to 5,000 photos/city, so that in total we gathered 232,265 photos, uploaded by 32,409 different users. Pictures from dataset DS_2 were used for testing the classifier, after a model was learned based on DS_1.

Evaluation Setup

The goal of our experiments is to evaluate the performance of the algorithm in finding city landmarks. We evaluate the accuracy of city landmark findings for the list of 50 different cities, included in the testing set DS_2, thus having in total 232,265 images at our disposal. The results of this analysis have been collected through a user survey. Additionally, with this user study we also compared our results against results produced by an existing system trying to solve the same problem, World Explorer [ANNY07]. Since World Explorer uses as input for its algorithms *Flickr* pictures with GPS data – *i.e.* richer input data than we need – our aim was to obtain at least comparable quality.

For the user survey we designed a simple application to evaluate landmark summaries created for the list of 50 cities. Most of the cities were European, since we expected our users to be more familiar with them. For the same list of cities, we also obtain city summaries from the World Explorer application. These summaries are also evaluated by our users and we compare the results for the two systems: ours - "TG-SVM" (TagGroups-SVM) and World Explorer. Since World Explorer needs as input geographical information instead of city names, for all 50 cities, we collected their associated GPS coordinates from the World Gazetteer database[19]. For retrieving tags representing city landmarks, we then made use of the World Explorer's Web API[20], specifying city zoom level 5 [ANNY07]. While World Explorer has 16 different zoom levels, we concentrated only on the single city-level zoom (level 5).

For each city, one needs to specify two pairs of GPS coordinates defining a rect-

[19] http://www.world-gazetteer.com
[20] http://developer.yahoo.com/yrb/tagmaps/

angular area inside the city (bottom - left and top - down corners), for which the World Explorer web service then returns the corresponding landmark-related tags. Given the fact that for the cities we have selected, we only had one single pair of GPS coordinates - basically specifying the city center - we had to define two additional pairs, such that the resulting rectangle had its' center coinciding with the city's. We assumed that most of the city landmarks are located around the city center. We experimented with two sizes of the rectangles' sides: 10x10 km or 5x5 km. However, the case where the rectangle sides were 10 km produced the best results.

For the evaluation setup we recruited 20 volunteers among our colleagues, who were familiar with photo sharing and search services. Each user was asked to evaluate two result sets for 10 randomly selected cities out of the set of 50, and the selection process picked each city so that by the end of the experiment it was evaluated by at least 4 users. Two photo summaries were mixed on a single screen, with one result set created using our algorithm and one coming from the World Explorer API. The users did not know which system produced each photo, as the photos from the two systems were randomly interleaved. Each photo was supplied with a title and a single landmark tag produced by either World Explorer or by our algorithm and used to retrieve this photo. A radio button was placed near each photo, where users could select between "landmark", "non-landmark", and "don't know" options. The users were asked to judge if a photo is a landmark or not, in total producing between 400 and 500 judgments per user. The experiment took about 30 minutes per user.

Participants were instructed that a landmark photo must (1) contain a whole landmark or large part of it and (2) the landmark must be a main topic, not just a background for a person photo. Users were allowed to use photo title and tag as hints when they could not decide based on the picture only.

Evaluation Results

We observed quite different user assessment patterns, some participants considered as landmarks lots of photos, while some others accepted only few of them. As a first analysis, we measured the performance of the two algorithms for each city separately. Having each city assessed by 4 users, we applied majority vote, such that each picture corresponding to a city was assigned to either "landmark", "non-landmark" or "don't know". We assigned 1 to "landmark" judgments, -1 to "non-landmark" and 0 to "don't know". If for a picture the sum of the 4 judgments was greater than 0, we considered that photo a "landmark", a score lower than 0 resulted in labeling the picture as "non-landmark". A majority score of 0 was never obtained for any of the pictures assessed in the user survey.

In Table 5.10 we present micro-average (averaged across all judgments per city) precision for each of the 50 analyzed cities. The results show that our method, TG-SVM, outperformed World Explorer on 30 out of 50 cities, *i.e.* 60% of the cases. On average over all 50 cities, World Explorer has a precision value of 0.32, and our

Nr.	City	PR (WE)	PR (TG-SVM)	Nr.	City	PR (WE)	PR (TG-SVM)	Nr.	City	PR (WE)	PR (TG-SVM)
1	amsterdam	0.33	**0.40**	18	stockholm	0.14	**0.16**	35	palermo	**0.50**	0.40
2	athens	0.21	**0.28**	19	helsinki	0.23	**0.30**	36	paris	**0.45**	0.16
3	barcelona	0.37	**0.44**	20	hongkong	0.16	**0.21**	37	riodejaneiro	**0.38**	0.20
4	beijing	0.27	**0.29**	21	istanbul	0.40	**0.60**	38	singapore	**0.21**	0.13
5	berlin	0.25	**0.48**	22	shanghai	0.43	**0.50**	39	sydney	**0.26**	0.08
6	birmingham	0.19	**0.28**	23	liverpool	0.47	**0.56**	40	tokyo	**0.25**	0.19
7	brasilia	0.40	**0.52**	24	yokohama	0.10	**0.16**	41	vienna	**0.37**	0.30
8	moscow	0.50	**0.75**	25	losangeles	0.09	**0.16**	42	bucharest	**0.62**	0.36
9	buenosaires	0.06	**0.28**	26	rome	0.42	**0.52**	43	cairo	**0.73**	0.56
10	naples	0.13	**0.40**	27	rotterdam	0.25	**0.48**	44	chicago	**0.33**	0.28
11	oslo	0.16	**0.17**	28	santiago	0.23	**0.28**	45	cologne	**0.53**	0.48
12	prague	0.20	**0.48**	29	saopaulo	0.04	**0.13**	46	florence	**0.67**	0.48
13	dresden	0.56	**0.75**	30	seville	0.38	**0.46**	47	genoa	**0.50**	0.42
14	toronto	0.05	**0.24**	31	madrid	**0.41**	0.32	48	hannover	**0.75**	0.33
15	turin	0.25	**0.48**	32	mexicocity	**0.32**	0.08	49	leeds	**0.28**	0.24
16	glasgow	0.39	**0.40**	33	munich	**0.26**	0.25	50	london	**0.29**	0.16
17	hamburg	0.19	**0.36**	34	newyork	**0.41**	0.27				

Table 5.10 Micro-Average Precision for 50 Cities

method, TG-SVM, 0.34. Results in Table 5.10 show an interesting aspect: for some of the cities, *e.g.* Moscow, Dresden, Istanbul or Liverpool, the precision values were very good, while for others, such as Mexico City, London, Paris, or Tokyo the obtained precision was quite low. The reason for this situation is the fact that results are strongly dependent on the quality of the pictures included in the corresponding city set. If the images we retrieve through Flick API do not really represent a view of the requested landmark, but just by chance contain the tags we used for querying the API and according to the *Flickr* ranking strategy they also have high scores, they will be retrieved and we will include them in the interface of the user experiment. By inspecting the pictures corresponding to London, Paris, Tokyo or Mexico City we could observe that the majority represented aerial views of the city where the landmarks were extremely difficult to identify, or were not present at all. In contrast to these, for Moscow, Dresden, Istanbul, *etc.* the corresponding images depicted indeed the landmarks they also have been tagged with.

User #	PR (WE)	PR (TG-SVM)	User #	PR (WE)	PR (TG-SVM)	User #	PR (WE)	PR (TG-SVM)	User #	PR (WE)	PR (TG-SVM)
1	0.42	**0.44**	6	0.32	**0.39**	11	**0.45**	0.41	16	**0.27**	**0.27**
2	0.45	**0.47**	7	0.26	**0.30**	12	0.77	**0.78**	17	0.35	**0.40**
3	0.38	**0.45**	8	0.29	**0.35**	13	**0.24**	0.29	18	0.18	**0.25**
4	0.26	**0.43**	9	0.11	**0.16**	14	0.22	**0.20**	19	0.15	**0.21**
5	0.23	**0.28**	10	0.22	**0.29**	15	**0.40**	0.37	20	0.62	**0.63**
Avg Prec(WE) = 0.33						Avg Prec(TG-SVM) = **0.37**					

Table 5.11 Macro-Average Precision for 20 Users

In Table 5.11 we present the results from each user using macro-average precision,

when all photos marked by users as landmarks are normalized by the total number of photos returned by an algorithm. Out of 20 users, 16 preferred our algorithm, 3 considered World Explorer-based results better and in one case the algorithms performed equally well. We obtained 12% improvement in precision with our method over World Explorer. We performed a paired t-test over the two outputs and calculated that precision improvement of our algorithm is statistically significant at confidence level $\alpha = 0.001$.

These results support our hypothesis that landmark finding based on photo classification can replace geo-tagging based methods in situations where geo-spatial information is not available. They also show that our algorithm significantly outperforms state-of-the-art algorithms for landmark search. There was no particular tuning of the representativeness score as defined by (Eq. 5.14). Estimating the best combination of these parameters might give additional boost to results' quality.

5.6 Discussion

Collaborative tagging is a valuable source of semantically rich metadata, which is especially useful for digital libraries covering multimedia resources whose non-textual content is not easily indexable / searchable. To tap this potential, we developed a series of algorithms for automatically inferring valuable knowledge and applicable for music resources and pictures. In this chapter we presented three different scenarios, where advanced algorithms can be applied in order to discover some of the hidden features of multimedia content, thus enabling easier access to this kind of content or improved retrieval. All algorithms presented in this chapter, thus address *Problem 3* announced in Chapter 1.

In the first part of this chapter we presented a novel approach for recommending music mood and theme annotations and thus enriching music tracks with tags often used in queries. Given the self-reinforcing nature of user generated tags, suggesting opinion- and usage-related music concepts to users results in a related tag vocabulary which converges to a more diverse set of tags. These will not only enrich our future training set for the learning algorithm, but will probably also enable fully automatic theme or mood tag assignment without user interaction.

The algorithms we proposed rely either on already available user tags, on lyrics, or on combinations of both. The results of our evaluations showed that providing such information is feasible and that we can achieve very good results both comparing our algorithms with user judgments and with the *AllMusic.com* experts' ground truth. Besides genres, themes in particular can be predicted well based on user tags and lyrics. For mood labels, performance is high only when using the first level classes that roughly correspond to basic human emotions. It is an interesting question for future work to investigate how many moods are really distinguishable by people. Our recommendation experiments within Facebook indicate that users are for certain

reasons (like social comparison) quite willing to collaborate in such evaluations. By recommending users such expert music annotations we bridge the gap between the two different vocabularies for describing music and help overcoming the strong bias toward genre tags in music tagging systems. Using our algorithm, music also becomes searchable by associated themes and moods, providing a first step towards effectively searching music inside digital libraries by textual, descriptive queries.

In the second part of this chapter we showed another use of tags for multimedia knowledge discovery: music hit prediction. Previous attempts to identify music hits relied entirely on lyrics or audio information for clustering or classifying song corporas. By using data from a Web 2.0 music site, our approach thus adds a new dimension to this kind of research. Our algorithms exploit social annotations and interactions in *Last.fm* that enable both measuring intrinsic similarity of songs and finding critical early-stage effects of cumulative advantage for tracks assumed to be popular. In order to be able to make accurate predictions about evolution and hit-potential of songs, it only requires those tracks to be inside the portal for one week. The large scale experiments we performed indicate good classification accuracy for our method and compared with previous comparable work we achieve $\approx 28\%$ inprovement in terms of AUC. The applications of our algorithm are manifold: record companies, radio stations, the artists themselves and last but not least, the users.

In the third and last part of this chapter we focused on discovering information for pictures and we addressed the problem of identifying pictures showing landmarks in a certain region / city, by using tag information and without relying on (still sparse) GPS coordinates data. Our algorithms exploit only *Flickr* tags and groups information. For finding relevant landmark-related tags we apply an SVM classifier for which the training data – both positive and negative examples – is extracted from thematical *Flickr* groups. The positive examples are chosen from traveling and landmark related groups, while the negative examples come from groups with generic photographical interests. Our results show that the two-class SVM classifier effectively finds landmark photos based on *Flickr* Groups training data, and is able to recognize landmarks which are not explicitly included in the training set. User evaluation results demonstrate that our method outperforms a state-of-the-art system relying on GPS information for solving the landmark finding task. The algorithms we described have the potential to be generalizable to help identifying not only city landmarks, but also other topical photos, such as “animals”, “flowers” or “cars”.

Chapter 6

Conclusions and Future Work

The amount of data available on the Web, in organizations and enterprises is multiplying at a rapid rate mainly due to the data storage capabilities becoming larger every day. At the same time, the popularity of Web 2.0 sites determined an increased participation of the large public in producing new content on its own and opened new ways for the users in sharing their experiences in form of documents (*e.g.*, pictures, bookmarks, music, *etc.*) with their family and friends. As a result, finding the right information among this vast amount of content available online has become a very tedious task. On the other hand, quite a significant amount of the publicly available content gets enhanced through the manual annotations (*i.e.*, tags) voluntarily provided by Web 2.0 users. Yet, it is not obvious whether and how these tags or subsets of them can be used for improving users' access to information. In this thesis we investigated these questions in detail and, based on the outcomes of this analysis, proposed a number of applications of tags for supporting search and personalization. This section first summarizes our major research contributions, and then discusses some issues which remained open for future investigations.

Summary of Contributions

Tag usage is rapidly increasing on the Web, providing potentially interesting information to improve search. To tap this potential, in Chapter 3 we extended previous preliminary work with a thorough analysis of the use of tags for different collections and in different environments. Our analysis is the first to present an in-depth study of tagging behaviors for very different kinds of resources and systems - Web pages (*Del.icio.us*), music (*Last.fm*), and images (*Flickr*) - and also comparing the results with anchor text characteristics. We analyzed and classified sample tags from these systems, to get an insight into what kinds of tags are used for different resources, and

provided statistics on tag distributions in all three tagging environments. For finding out whether tags bring new information to the content they annotate, we checked the overlap of tags with content, with metadata assigned by experts or coming from other verified sources. Another important aspect which we investigated refers to the potential of different kinds of tags for improving search, and we compared them with user queries posted to search engines. Additionally, we conducted a user survey in order to better understand the perception of users regarding usefulness of tags for search.

Our analysis provided evidence for the usefulness of a common tag classification scheme for different collections, and has shown that the distributions of tag types strongly depend on the resources they annotate: For *Flickr*, *Del.icio.us* and *Anchor Text (AT) Topic*-related tags are appearing in more than 50% of the cases, while for *Last.fm* the *Type* category is the most prominent one. More than 50% of existing tags bring new information to the resources they annotate and for the music domain, this is the case for 98.5% of the tags. A large amount of tags is accurate and reliable, for the music domain for example 73.01% of the tags also occur in online music reviews. Our study proved that most of the tags can be used for search, and that in most cases tagging behavior exhibits approximately the same characteristics as searching behavior. We also observed some noteworthy differences: For the music domain, *Usage context* is very useful for search, yet underrepresented in the tagging material. Similarly, for pictures and music *Opinions/Qualities* queries occur quite often, although people tend to neglect this category for tagging.

These observations motivated us to develop methods for automatically classifying tags into the eight categories building our tag taxonomy and we introduced two types of methods for achieving this goal – rule-based and model-based methods. We compared the automatic tag classification produced by our algorithms against a ground truth data set, consisting of manual tag type assignments produced by human raters. Experimental results showed that our methods can identify tag types with high accuracy (80-90%), thus enabling further improvement of systems making use of social tags.

In Chapter 4 we continued by analyzing the potential of tags for supporting personalization applications. We considered two different aspects: (1) using tags in order to provide personalized music recommendations; and (2) using tags for achieving personalized Web ranking. In the first part of this chapter we analyzed tags from the point of view of their potential of characterizing the users and thus enabling personalized recommendations. Using data from *Last.fm*, we analyzed tag usage and statistics and investigated the use of tag-based user profiles in contrast to conventional user profiles based on song and track usage. We specified recommendation algorithms based on tag user profiles, and explored how collaborative filtering recommendations based on these tag profiles are different from recommendations based on song/track profiles. Finally, we described a set of new search-based methods, which use tags to

recommend songs interesting to a user, yielding substantially improved results – 44% increase in quality for the best algorithm over collaborative filtering.

In the second part of this chapter, we showed how to personalize Web ranking, by relying on annotations produced by human experts and gathered from the ODP catalog. Given that directories like ODP contain only a very small amount of tagged pages, compared to the Google's number of indexed pages, we investigated the impact these annotations have and specifically their feasibility to implement personalized search based on these tags. We introduced an additional criterion for Web page ranking, namely the distance between a user profile defined with taxonomical tags and the sets of topics covered by each URL returned in Web search. The precision achieved by this technique significantly surpassed the precision offered by Google search, reaching up to 63% in quality improvement. Additionally, we showed that extending the manual ODP classifications from 4 million entries to a 8 billion Web is feasible, based on an analysis of how topic classifications for subsets of large page collections can be extended to this large collection via topic-sensitive biasing of PageRank values.

Finally, in Chapter 5 we presented a third use of tags, namely automatically inferring valuable information about the resources tags are attached to. We focused on inferring information for multimedia content and we presented three different scenarios, where such advanced algorithms can be applied in order to discover some of the content's hidden features, thus enabling easier access to information and improved retrieval.

Building on the results of the analysis included in Chapter 3, which showed that the types of music tags which would be really beneficial for supporting retrieval - usage (*theme*) and opinion (*mood*) tags - are often neglected by users in the annotation process, in the first scenario presented in this chapter we address exactly this problem: in order to support users in tagging and filling these gaps in the tag space, we develop algorithms for automatically inferring mood and theme annotations, which are then recommended to the users. Our methods exploit the available user annotations, the lyrics of music tracks, as well as combinations of both. We also compared the results for our recommended mood / theme annotations against genre and style recommendations - a much easier and already studied task. Besides evaluating against an expert (AllMusic.com) ground truth, we also evaluated the quality of our recommended tags through a Facebook-based user study and the results showed that we can achieve very good results both comparing our algorithms with user judgments and with the AllMusic.com experts' ground truth. By recommending users such expert music annotations we bridge the gap between the two different vocabularies for describing music and help overcoming the strong bias toward genre tags in music tagging systems. Using our algorithm, music also becomes searchable by associated themes and moods, providing a first step towards effectively searching music inside digital libraries by textual, descriptive queries.

The second scenario we focused on, centered around the use of tags for identifying

potential music hits. Previous attempts to identify hit songs have mostly focused on the intrinsic characteristics of the songs, such as lyrics and audio features. We introduced a new method for predicting the potential of music tracks for becoming hits, which instead of relying on intrinsic characteristics of the tracks, directly uses data mined from a music social network and the relationships between tracks, artists and albums. We evaluated the performance of our algorithms through a set of experiments and the results indicate good accuracy in correctly identifying music hits, as well as significant improvement over existing approaches ($\approx 28\%$ inprovement in terms of AUC).

In the third scenario we aimed at identifying pictures depicting landmarks in a certain region / city and experimented with a subset of *Flickr* photos. While first algorithms based on geo-tagged photos have been suggested for this task, the majority of pictures is still without GPS coordinates, and therefore neglected by these algorithms. We proposed a new method to identify city landmarks using only common textual tags. For finding photos representing city landmarks we applied a SVM classifier trained with positive examples from landmark related *Flickr* groups and negative examples from general *Flickr* groups. Representative tags are extracted and used to construct a landmark photo summary. We evaluated our algorithms through a user study, results showing that our new method significantly outperformed state-of-the-art geo-tagging based algorithms. Moreover, the algorithms we described have the potential to be generalizable to help identifying not only city landmarks, but also other topical photos, such as "animals", "flowers", "cars", *etc.*

Open Directions

In this thesis we presented a number of applications of tags for search and personalization. Nevertheless, there is still room for improvements and further research directions. Some of future interesting research questions refer for example to detailed investigations regarding which kinds of queries can be best supported by which kind of information, *i.e.* tag-information, content or other metadata. This will help us to strategically extend existing knowledge gathered from different sources and provide better support to queries, especially for pictures and music resources which cannot be handled well enough by existing techniques.

Regarding automatic tag classification, a possible improvement of our current methods refers to exploiting resource features such as title / description for Web pages, lyrics for songs, or attributes extracted by content-based methods in order to learn a tag's type based on the concrete resource tagged. We also intend to extend the model-based methods to enable machine learning of some categories now identified by rules or look-ups. For this, more intensive experimentation with WordNet, as well as incorporation of related approaches exploiting metadata such as time stamps or GPS coordinates seem promising.

The personalized music recommendation algorithms we presented in Chapter 4

can be also further improved by incorporating relevance feedback into search-based recommendations, such that the user is able to select negative tracks or tags, genres s/he does not like, live performances, or even instrumental setups. Similarly, relevance feedback can be also used to enhance the personalized Web search algorithm we introduced in Section 4.4.

Regarding the knowledge discovery methods discussed in Chapter 5, we believe that further improvement can be achieved through better feature selection mechanisms. One possibility to select the most representative features to be fed to the classifiers is to take into account the class type (*e.g.*, Topic, Author, Usage context, *etc.*) of the tags considered as input. For this, accurate automatic classification of tag types is required. Moreover, merging tag-based features with features representing low-level characteristics of the content is also worth examining.

Appendix A

Curriculum Vitae

Raluca Paiu, born on December 2^{nd} 1980, in Bucharest, Romania.

Nov. 2007 -	Project manager at Forschungszentrum L3S, Universität Hannover
Oct. 2004 - Oct. 2007	Junior researcher at Forschungszentrum L3S, Universität Hannover
Mar. - Aug. 2004	Master Studies in Computer Science, Universität Hannover Title of the thesis: *"Semantic Web Search"*
1999 - 2004	Bachelor Studies in Computer Science, Politehnica University, Bucharest, Romania
Oct. 2005 -	Teaching assistant for lectures "Artificial Intelligence I, II", Universität Hannover
Oct. 2003 - Mar. 2004	Web Programmer, NetStyle Ltd., www.netstyle.ro
Mar. - Jul. 2003	Teaching Assistant for the lecture "Analysis and Synthesis of Numerical Devices" Politehnica University, Bucharest, Romania
Aug. 2002 - Mar. 2003	Programmer at Datagram Ltd. www.datagram.ro
Jul. 2002 - Aug. 2002	Practice at Datagram Ltd. www.datagram.ro
Jul. - Aug. 2001	Practice at the National Institute for Research and Development in Informatics, Bucharest, Romania. www.ici.ro

Bibliography

[ACD+08] Eugene Agichtein, Carlos Castillo, Debora Donato, Aristides Gionis, and Gilad Mishne. Finding high-quality content in social media. In *WSDM '08: Proceedings of 1st ACM International Conference on Web Search and Data Mining*, pages 183–194, Stanford, California, USA., 2008.

[ACN+09] Rabeeh Abbasi, Sergey Chernov, Wolfgang Nejdl, Raluca Paiu, and Steffen Staab. Exploiting flickr tags and groups for finding landmark photos. In *ECIR' 09: Proceedings of the 31st European Conference on Information Retrieval*, pages 654–661, Toulouse, France, 2009.

[AN04] Mehmet S. Aktas and Mehmet A. Nacar. Personalizing pagerank based on domain profiles. In *Proceedings of WebKDD 2004: KDD Workshop on Web Mining and Web Usage Analysis*, pages 22–25, Seattle, Washington, USA, 2004.

[AN07] Morgan Ames and Mor Naaman. Why we tag: motivations for annotation in mobile and online media. In *Proceedings of SIGCHI*, pages 971–980, San Jose, California, USA, 2007.

[ANNY07] Shane Ahern, Mor Naaman, Rahul Nair, and Jeannie Hui-I Yang. World explorer: visualizing aggregate data from unstructured text in georeferenced collections. In *JCDL '07: Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*, pages 1–10, Vancouver, BC, Canada, 2007. ACM.

[ASC07] Rabeeh Abbasi, Steffen Staab, and Philipp Cimiano. Organizing resources on tagging systems using t-org. In *Proceedings of Workshop on*

Bridging the Gap between Semantic Web and Web 2.0 at ESWC 2007, Innsbruck, Austria, June 2007.

[AT05] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, 2005.

[BCC+06] Ingo Brunkhorst, Paul A. Chirita, Stefania Costache, Julien Gaugaz, Ekaterini Ioannou, Tereza Iofciu, Enrico Minack, Wolfgang Nejdl, and Raluca Paiu. The beagle++ toolbox: Towards an extendable desktop search architecture. In *Proceedings of the Semantic Desktop and Social Semantic Collaboration Workshop at the International Semantic Web Conference, ISWC '06*, Athens, GA, USA, 2006.

[BdVBF07] Henk Blanken, Arjen P. de Vries, Henk Ernst Blok, and Ling Feng. *Multimedia Retrieval*. Springer, 2007.

[Bes06] David Best. Web 2.0: Next big thing or next big internet bubble. *Technische Universiteit Eindhoven*, 2006.

[BF07] Tamara Lee Berg and David Forsyth. Automatic ranking of iconic images. Technical Report UCB/EECS-2007-13, EECS Department, University of California, Berkeley, Jan 2007.

[BFG+09] Kerstin Bischoff, Claudiu S. Firan, Mihai Georgescu, Wolfgang Nejdl, and Raluca Paiu. Social knowledge-driven music hit prediction. In *ADMA '09: Proceedings of the 5th International Conference on Advanced Data Mining and Applications*, Beijing, China, 2009.

[BFK+09] Kerstin Bischoff, Claudiu S. Firan, Cristina Kadar, Wolfgang Nejdl, and Raluca Paiu. Automatically identifying tag types. In *ADMA '09: Proceedings of the 5th International Conference on Advanced Data Mining and Applications*, Beijing, China, 2009.

[BFNP08] Kerstin Bischoff, Claudiu S. Firan, Wolfgang Nejdl, and Raluca Paiu. Can all tags be used for search? In *CIKM '08: Proceedings of the 17th ACM conference on Information and Knowledge Mining*, pages 193–202, New York, NY, USA, 2008. ACM.

[BFNP09] Kerstin Bischoff, Claudiu S. Firan, Wolfgang Nejdl, and Raluca Paiu. How do you feel about "dancing queen"? deriving mood & theme annotations from user tags. In *JCDL '09: Proceedings of the Joint Conference on Digital Libraries*, Austin, Texas, 2009.

[BFP09] Kerstin Bischoff, Claudiu S. Firan, and Raluca Paiu. Deriving music theme annotations from user tags. In *WWW '09: Proceedings of the 18th international conference on World Wide Web*, pages 1193–1194, Madrid, Spain, 2009. ACM.

[Bila] Billboard.biz. http://www.billboard.biz/bbbiz/charts/currentalbum.jsp.

[Bilb] Billboard.biz. http://www.billboard.biz/bbbiz/charts/currentsingles.jsp.

[Bor93] J. Bortz. *Statistics for Social Scientists*. Springer Verlag, 1993.

[Bri98] Sergey Brin. The anatomy of a large-scale hypertextual web search engine. In *Computer Networks and ISDN Systems*, pages 107–117, 1998.

[BWC07] Andrew Byde, Hui Wan, and Steve Cayzer. Personalized tag recommendations via tagging and content-based similarity metrics. In *Poster Session at the 1st International Conference on Weblogs and Social Media (ICWSM)*, Boulder, Colorado, U.S.A., 2007.

[BXW+07] Shenghua Bao, Guirong Xue, Xiaoyuan Wu, Yong Yu, Ben Fei, and Zhong Su. Optimizing web search using social annotations. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 501–510, Banff, Alberta, Canada, 2007. ACM.

[CBHS08] Ciro Cattuto, Dominik Benz, Andreas Hotho, and Gerd Stumme. Semantic grounding of tag relatedness in social bookmarking systems. In *International Semantic Web Conference*, pages 615–631, Karlsruhe, Germany, 2008.

[CCNP05] Paul-Alexandru Chirita, Stefania Costache, Wolfgang Nejdl, and Raluca Paiu. Semantically enhanced searching and ranking on the desktop. In *Proceedings of the Workshop on the Semantic Desktop - Next Generation Personal Information Management and Collaboration Infrastructure at the International Semantic Web Conference*, Galway, Ireland, 2005.

[CCNP06] Paul-Alexandru Chirita, Stefania Costache, Wolfgang Nejdl, and Raluca Paiu. Beagle^{++}: Semantically enhanced searching and ranking on the desktop. In *ESWC '06: Proceedings of the 3rd European Semantic Web Conference*, pages 348–362, Budva, Montenegro, 2006.

[CF00] William W. Cohen and Wei Fan. Web-collaborative filtering: recommending music by crawling the Web. *Computer Networks (Amsterdam, Netherlands: 1999)*, 33(1–6):685–698, 2000.

[CGG+05] Paul-Alexandru Chirita, Rita Gavriloaie, Stefania Ghita, Wolfgang Nejdl, and Raluca Paiu. Activity based metadata for semantic desktop

search. In *ESWC '05: Proceedings of the 2nd European Semantic Web Conference*, pages 439–454, Crete, Greece, 2005.

[Cha99] Philip K. Chan. Constructing web user profiles: A non-invasive learning approach. In *In Web Usage Analysis and User Profiling, LNAI 1836*, pages 39–55. Springer-Verlag, 1999.

[CNP05a] Stefania Costache, Wolfgang Nejdl, and Raluca Paiu. Semantically rich recommendations in social networks for sharing and exchanging semantic context. In *Proceedings of the 2nd European Semantic Web Conference Workshop on Ontologies in P2P Communities*, Crete, Greece, 2005.

[CNP05b] Stefania Costache, Wolfgang Nejdl, and Raluca Paiu. Semantically rich recommendations in social networks for sharing, exchanging and ranking semantic context. In *ISWC '05: Proceedings of the 4th International Semantic Web Conference*, pages 293–307, Galway, Ireland, 2005.

[CNP07] Stefania Costache, Wolfgang Nejdl, and Raluca Paiu. Personalizing pagerank-based ranking over distributed collections. In *CAiSE '07: Proceedings of the 19th International Conference on Advanced Information Systems Engineering*, pages 111–126, Trondheim, Norway, 2007.

[CNPK05] Paul Alexandru Chirita, Wolfgang Nejdl, Raluca Paiu, and Christian Kohlschütter. Using odp metadata to personalize search. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 178–185, Salvador, Brazil, 2005. ACM.

[Coh60] Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.

[CON04] Paul-Alexandru Chirita, Daniel Olmedilla, and Wolfgang Nejdl. Pros: A personalized ranking platform for web search. In *AH '04: Proceedings of the International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, Eindhoven, The Netherlands, August 2004.

[CRH05] Oscar Celma, Miguel Ramirez, and Perfecto Herrera. Foafing the music: A music recommendation system based on rss feeds and user preferences. In *ISMIR '05: Proceedings of the 6th International Conference on Music Information Retrieval*, London, UK, 2005.

[CSB06] Song Hui Chon, Malcolm Slaney, and Jonathan Berger. Predicting success from music sales data: a statistical and adaptive approach. In *AMCMM '06: Proceedings of the 1st ACM workshop on Audio and music computing multimedia*, pages 83–88, Santa Barbara, California, USA., 2006. ACM.

[CZZM07] Rui Cai, Chao Zhang, Lei Zhang, and Wei-Ying Ma. Scalable music recommendation by search. In *MULTIMEDIA '07: Proceedings of the 15th international conference on Multimedia*, pages 1065–1074, New York, NY, USA, 2007. ACM.

[Dav00] Brian D. Davison. Topical locality in the web. In *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 272–279, Athens, Greece, 2000. ACM.

[DEFS06] Pavel A. Dmitriev, Nadav Eiron, Marcus Fontoura, and Eugene Shekita. Using annotations in enterprise search. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 811–817, Edinburgh, Scotland, 2006. ACM.

[DiN99] Darcy DiNucci. Fragmented future. *Print 53 (4):32*, 1999.

[DKGS04] Marc Davis, Simon King, Nathan Good, and Risto Sarvas. From context to content: leveraging context to infer media metadata. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 188–195, New York, NY, USA, 2004. ACM.

[DKNS01] Cynthia Dwork, Ravi Kumar, Moni Naor, and D. Sivakumar. Rank aggregation methods for the web. In *WWW '01: Proceedings of the 10th international conference on World Wide Web*, pages 613–622, New York, NY, USA, 2001. ACM.

[DL05] Ruth Dhanaraj and Beth Logan. Automatic prediction of hit songs. In *ISMIR '05: Proceedings of the 6th International Conference on Music Information Retrieval*, London, UK, 2005.

[DNP05] Andrei Damian, Wolfgang Nejdl, and Raluca Paiu. Peer-sensitive objectrank - valuing contextual information in social networks. In *WISE '05: Proceedings of the 6th International Conference on Web Information Systems Engineering*, pages 512–519, New York, NY, USA, 2005.

[DS99] Search Engine Watch Danny Sullivan. More evil than dr. evil?, 1999.

[DSW07] Zhicheng Dou, Ruihua Song, and Ji-Rong Wen. A large-scale evaluation and analysis of personalized search strategies. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 581–590, Banff, Alberta, Canada, 2007. ACM.

[EG04] Barbara Di Eugenio and Michael Glass. The kappa statistic: A second look. *Computational Linguistics*, 30(1):95–101, 2004.

[FBJ03] Joachim E. Fischer, Lucas M. Bachmann, and Roman Jaeschke. A readers' guide to the interpretation of diagnostic test properties: clinical example of sepsis. *Intensive Care Medicine Journal*, 29(7):1043–1051, 2003.

[FNP] Claudiu S. Firan, Wolfgang Nejdl, and Raluca Paiu. The benefit of using tag-based profiles. In *LA-WEB '07: Proceedings of the 2007 Latin American Web Conference*, pages 32–41, Santiago, Chile. IEEE Computer Society.

[FZP03] Yazhong Feng, Yueting Zhuang, and Yunhe Pan. Popular music retrieval by detecting mood. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 375–376, Toronto, Canada, 2003. ACM.

[GGK+05] Daniel Gruhl, R. Guha, Ravi Kumar, Jasmine Novak, and Andrew Tomkins. The predictive power of online chatter. In *KDD '05: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 78–87, Chicago, Illinois, USA, 2005. ACM.

[GH06] Scott A. Golder and Bernardo A. Huberman. Usage patterns of collaborative tagging systems. *Journal of Information Science*, 2006.

[Goo] Google search api. http://api.google.com.

[HA99] Bernardo A. Huberman and Lada A. Adamic. Internet: Growth dynamics of the world-wide web. *Nature*, 401(6749):131, 1999.

[Hav02] T. Haveliwala. Topic-sensitive pagerank. In *WWW '02: Proceedings of the 11th International Conference on World Wide Web*, Honolulu, Hawaii, USA, 2002.

[HD07] Xiao Hu and J. Stephen Downie. Exploring mood metadata: Relationships with genre, artist and usage metadata. In *ISMIR '07: Proceedings of the 8th International Conference on Music Information Retrieval*, Vienna, Austria, 2007.

[HHLS05] Tony Hammond, Timo Hannay, Ben Lund, and Joanna Scott. Social bookmarking tools - a case study. In *D-Lib Magazine*, 2005.

[HJSS06a] Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. G.: Emergent semantics in bibsonomy. In *Proceedings of Workshop on Applications of Semantic Technologies, Informatik 2006. Lecture Notes in Informatics*, 2006.

[HJSS06b] Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. Information retrieval in folksonomies: Search and ranking. In *ESWC '06: Proceedings of the 3rd European Semantic Web Conference*, Budva, Montenegro, 2006.

[HKGM07] Paul Heymann, Georgia Koutrika, and Hector Garcia-Molina. Can social bookmarking improve web search? *Technical Report*, 2007.

[HRGM08] Paul Heymann, Daniel Ramage, and Hector Garcia-Molina. Social tag prediction. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 531–538, Singapore, Singapore, 2008. ACM.

[HRS07] Harry Halpin, Valentin Robu, and Hana Shepherd. The complex dynamics of collaborative tagging. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 211–220, Banff, Alberta, Canada, 2007. ACM.

[IKNP05] Tereza Iofciu, Christian Kohlschütter, Wolfgang Nejdl, and Raluca Paiu. Keywords and rdf fragments: Integrating metadata and full-text search in beagle++. In *Proceedings of the Workshop on the Semantic Desktop - Next Generation Personal Information Management and Collaboration Infrastructure at the International Semantic Web Conference*, Galway, Ireland, 2005.

[JK00] Kalervo Järvelin and Jaana Kekäläinen. Ir evaluation methods for retrieving highly relevant documents. In *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 41–48, Athens, Greece, 2000. ACM.

[JL04] Patrik N. Juslin and Petri Laukka. Expression, perception, and induction of musical emotions: A review and a questionnaire study of everyday listening. *Journal of New Music Research*, 33(3):217–238, 2004.

[JNTD06] Alexandar Jaffe, Mor Naaman, Tamir Tassa, and Marc Davis. Generating summaries and visualization for large collections of geo-referenced photographs. In *MIR '06: Proceedings of the 8th ACM international workshop on Multimedia information retrieval*, pages 89–98, Santa Barbara, California, USA, 2006. ACM.

[Joa02] Thorsten Joachims. *Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA, 2002.

[JW03] Glen Jeh and Jennifer Widom. Scaling personalized web search. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 271–279, Budapest, Hungary, 2003. ACM.

[Kan04] Min-Yen Kan. Web page classification without the web page. In *WWW Alt. '04: Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pages 262–263, New York, NY, USA, 2004. ACM.

[KHMG03] Sepandar Kamvar, Taher Haveliwala, Christopher Manning, and Gene Golub. Exploiting the block structure of the web for computing pagerank. Technical report, Stanford InfoLab, 2003.

[Kle99] Jon Kleinberg. Authoritative sources in a hyperlinked environment. In *Journal of the Association for Computing Machinery*, pages 604–632, 1999.

[KN08] Lyndon Kennedy and Mor Naaman. Generating diverse and representative image search results for landmarks. In *WWW '08: Proceedings of the 17th International World Wide Web Conference*, Beijing, China, 2008.

[KNA+07] Lyndon Kennedy, Mor Naaman, Shane Ahern, Rahul Nair, and Tye Rattenbury. How flickr helps us make sense of the world: context and content in community-contributed media collections. In *MULTIMEDIA '07: Proceedings of the 15th international conference on Multimedia*, pages 631–640, Augsburg, Germany, 2007. ACM.

[KPSW07] Peter Knees, Tim Pohle, Markus Schedl, and Gerhard Widmer. A music search engine built upon audio-based and web-based similarity measures. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 447–454, Amsterdam, The Netherlands, 2007. ACM.

[KZ04] Reiner Kraft and Jason Zien. Mining anchor text for query refinement. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 666–674, New York, NY, USA, 2004. ACM.

[LBM03] Y. Li, Z. A. Bandar, and D. McLean. An approach for measuring semantic similarity between words using multiple information sources. *IEEE Transactions on Knowledge and Data Engineering*, 15(4):871–882, 2003.

[LGH08] Cyril Laurier, Jens Grivolla, and Perfecto Herrera. Multimodal music mood classification using audio and lyrics. In *ICMLA '08: Proceedings of the 2008 Seventh International Conference on Machine Learning*

and Applications, pages 688–693, Washington, DC, USA, 2008. IEEE Computer Society.

[LGZ08] Xin Li, Lei Guo, and Yihong Eric Zhao. Tag-based social interest discovery. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 675–684, Beijing, China, 2008. ACM.

[LH07] Cyril Laurier and Perfecto Herrera. Audio music mood classification using support vector machine. In *ISMIR '07: Proceedings of the 8th International Conference on Music Information Retrieval*, Vienna, Austria, 2007.

[LKGO04] Qing Li, Byeong Man Kim, Dong Hai Guan, and Duk whan Oh. A music recommender based on audio features. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 532–533, Sheffield, United Kingdom, 2004. ACM.

[LLZ03] Dan Liu, Lie Lu, and Hong-Jiang Zhang. Automatic mood detection from acoustic music data. In *ISMIR '03: Proceedings of the 4th International Conference on Music Information Retrieval*, Washington, D.C., USA, 2003.

[LM00] R. Lempel and S. Moran. The stochastic approach for link-structure analysis (SALSA) and the TKC effect. *Computer Networks (Amsterdam, Netherlands: 1999)*, 33(1–6):387–401, 2000.

[LOL03] Tao Li, Mitsunori Ogihara, and Qi Li. A comparative study on content-based music genre classification. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 282–289, Toronto, Canada, 2003. ACM.

[Lor63] Edward N. Lorenz. Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*, 20(2):130–141, 1963.

[LYM04] Fang Liu, Clement Yu, and Weiyi Meng. Personalized web search for improving retrieval effectiveness. *IEEE Trans. on Knowl. and Data Eng.*, 16(1):28–40, 2004.

[Mat04] Adam Mathes. Folksonomies - cooperative classification and communication through shared metadata. *Technical report*, 2004.

[McB94] Oliver A. McBryan. GENVL and WWWW: Tools for Taming the Web. In *WWW '94: Proceedings of the 1st International Conference on World Wide Web*, Geneva, Switzerland, 1994.

[Mik07] Peter Mika. Ontologies are us: A unified model of social networks and semantics. *Web Semantics*, 5(1):5–15, 2007.

[Mil95] G.A. Miller. Wordnet: An electronic lexical database. *Communications of the ACM*, 38(11):39–41, 1995.

[Mis06] Gilad Mishne. Autotag: a collaborative approach to automated tag assignment for weblog posts. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 953–954, Edinburgh, Scotland, 2006. ACM.

[MMC+05] Jose P. G. Mahedero, Álvaro MartÍnez, Pedro Cano, Markus Koppenberger, and Fabien Gouyon. Natural language processing of lyrics. In *MULTIMEDIA '05: Proceedings of the 13th annual ACM international conference on Multimedia*, pages 475–478, Hilton, Singapore, 2005. ACM.

[MNBD06a] Cameron Marlow, Mor Naaman, Danah Boyd, and Marc Davis. Ht06, tagging paper, taxonomy, flickr, academic article, to read. In *HYPERTEXT '06: Proceedings of the seventeenth conference on Hypertext and hypermedia*, pages 31–40, Odense, Denmark, 2006. ACM.

[MNBD06b] Cameron Marlow, Mor Naaman, Danah Boyd, and Marc Davis. Position Paper, Tagging, Taxonomy, Flickr, Article, ToRead. In *Collaborative Web Tagging Workshop at WWW2006*, Edinburgh, Scotland, May 2006.

[MPC+09] Enrico Minack, Raluca Paiu, Stefania Costache, Gianluca Demartini, Julien Gaugaz, Ekaterini Ioannou, Paul A. Chirita, and Wolfgang Nejdl. Leveraging personal metadata for desktop search – the beagle++ system. *Journal of Web Semantics. To appear*, 2009.

[MRS08] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[MSDR04] Stuart E. Middleton, Nigel R. Shadbolt, and David C. De-Roure. Ontological user profiling in recommender systems. *ACM Transactions on Information Systems*, (1):54–88, January 2004.

[NHW+04] Mor Naaman, Susumu Harada, QianYing Wang, Hector Garcia-Molina, and Andreas Paepcke. Context data in geo-referenced digital photo collections. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 196–203, New York, NY, USA, 2004. ACM.

[NM07] Michael G. Noll and Christoph Meinel. Web search personalization via social bookmarking and tagging. In Karl Aberer, Key-Sun Choi, Natasha

Noy, Dean Allemang, Kyung-Il Lee, Lyndon Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber, and Philippe Cudré-Mauroux, editors, *The Semantic Web*, volume 4825 of *LNCS*, pages 367–380, Heidelberg, Germany, 2007. Springer.

[NP05a] Wolfgang Nejdl and Raluca Paiu. Desktop search - how contextual information influences search results & rankings. In *Proceedings of the 2nd SIGIR Workshop on Information Retrieval in Context (IRiX)*, Salvador, Brazil, 2005.

[NP05b] Wolfgang Nejdl and Raluca Paiu. I know i stored it somewhere - contextual information and ranking on our desktop. In *Proceedings of the 8th International Workshop of the DELOS Network of Excellence on Digital Libraries on Future Digital Library Management Systems (System Architecture & Information Access)*, Dagstuhl, Germany, 2005.

[NSPGM04] Mor Naaman, Yee Jiun Song, Andreas Paepcke, and Hector Garcia-Molina. Automatic organization for digital photographs with geographic coordinates. In *JCDL '04: Proceedings of the 4th ACM/IEEE-CS Joint Conference on Digital libraries*, pages 53–62, Tuscon, AZ, USA, 2004. ACM.

[NZJ01] Andrew Y. Ng, Alice X. Zheng, and Michael I. Jordan. Stable algorithms for link analysis. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 258–266, New Orleans, Louisiana, United States, 2001. ACM.

[O'R05] Tim O'Reilly. What is web 2.0? *O'Reilly Network*, 2005.

[Pan] Pandora internet radio and music genome project. http://www.pandora.com.

[PBMW98] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford University, 1998.

[PCT06] Greg Pass, Abdur Chowdhury, and Cayley Torgeson. A picture of search. In *InfoScale '06: Proceedings of the 1st international conference on Scalable information systems*, page 1, Hong Kong, 2006. ACM.

[PR08] Francois Pachet and Pierre Roy. Hit song science is not yet a science. In *ISMIR '08: Proceedings of the 9th International Conference on Music Information Retrieval*, Philadelphia, Pennsylvania, USA., 2008.

[PVV06] Steffen Pauws, Wim Verhaegh, and Mark Vossen. Fast generation of optimal music playlists using local search. In *ISMIR '06: Proceedings of the 7th International Conference on Music Information Retrieval*, Victoria, Canada, 2006.

[QC06] Feng Qiu and Junghoo Cho. Automatic identification of user interest for personalized search. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 727–736, Edinburgh, Scotland, 2006. ACM.

[Qui] Quitura. http://quintura.com.

[RD02] Mathew Richardson and Pedro Domingos. The intelligent surfer: Probabilistic combination of link and content information in pagerank. In *Advances in Neural Information Processing Systems 14*, 2002.

[RGN07] Tye Rattenbury, Nathaniel Good, and Mor Naaman. Towards automatic extraction of event and place semantics from flickr tags. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 103–110, Amsterdam, The Netherlands, 2007. ACM.

[Sch06] Patrick Schmitz. Inducing ontology from flickr tags. In *Proceedings of the Collaborative Web Tagging Workshop (WWW '06)*, Edinburgh, Scotland, May 2006.

[Sea] Search cloudlet. http://www.getcloudlet.com.

[SHDK07] Eric Schwarzkopf, Dominik Heckmann, Dietmar Dengler, and Alexander Krner. Mining the structure of tag spaces for user modeling. In *Workshop on Data Mining for User Modeling*, Corfu, Greece, 2007.

[SHJS06] Christoph Schmitz, Andreas Hotho, Robert Jäschke, and Gerd Stumme. Mining association rules in folksonomies. *Data Science and Classification*, pages 261–270, 2006.

[SKR01] J. Ben Schafer, Joseph A. Konstan, and John Riedl. E-commerce recommendation applications. *Data Min. Knowl. Discov.*, 5(1-2):115–153, 2001.

[SLR+06] Shilad Sen, Shyong K. Lam, Al Mamunur Rashid, Dan Cosley, Dan Frankowski, Jeremy Osterhouse, F. Maxwell Harper, and John Riedl. tagging, communities, vocabulary, evolution. In *CSCW '06: Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*, pages 181–190, Banff, Alberta, Canada, 2006. ACM.

[SMvdP07] Janto Skowronek, Martin McKinney, and Steven van de Par. A demonstrator for automatic music mood estimation. In *ISMIR '07: Proceedings of the 8th International Conference on Music Information Retrieval*, Vienna, Austria, 2007.

[SOHB07] Sanjay Sood, Sara Owsley, Kristian Hammond, and Larry Birnbaum. Tagassist: Automatic tag suggestion for blog posts. In *ICWSM '07: Proceedings of the 1st International Conference on Weblogs and Social Media*, Boulder, Colorado, U.S.A., 2007.

[SSKO87] Phillip Shaver, Judith Schwartz, Donald Kirson, and Cary O'Connor. Emotion knowledge: Further exploration of a prototype approach. *Journal of Personality and Social Psychology*, 1987.

[SSYC06] Dou Shen, Jian-Tao Sun, Qiang Yang, and Zheng Chen. A comparison of implicit and explicit links for web page classification. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 643–650, Edinburgh, Scotland, 2006. ACM.

[Sta] Stanford webbase project. http://webbase.stanford.edu.

[Sul04] Danny Sullivan. The older you are, the more you want personalized search. Search Engine Watch. http://searchenginewatch.com/3385131, 2004.

[SvZ08] Börkur Sigurbjörnsson and Roelof van Zwol. Flickr tag recommendation based on collective knowledge. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 327–336, Beijing, China, 2008. ACM.

[TAG] Tagvy. http://www.tagvy.com.

[TLR03] Kentaro Toyama, Ron Logan, and Asta Roseway. Geographic location tags on digital images. In *MULTIMEDIA '03: Proceedings of the eleventh ACM international conference on Multimedia*, pages 156–166, Berkeley, CA, USA, 2003. ACM.

[TM02] F. Tanudjaja and L. Mui. Persona: A contextualized and personalized web search. In *HICSS '02: Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS'02)-Volume 3*, page 67, Washington, DC, USA, 2002. IEEE Computer Society.

[Vap99] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory (Information Science and Statistics)*. Springer, November 1999.

[Voo99] Ellen M. Voorhees. The trec-8 question answering track report. In *In Proceedings of TREC-8*, pages 77–82, 1999.

[Wal05] Thomas Vander Wal. Folksonomy definition and wikipedia :: Off the top :: vanderwal.net, November 2005.

[Wat07] Duncan J. Watts. Is justin timberlake a product of cumulative advantage? *New York Times, April 15*, 2007.

[Wik] Web 2.0 - wikipedia, the free encyclopedia. http://en.wikipedia.org/wiki/web_2.0. ver. 288413238.

[Win62] J. B. Winer. *Statistical principles in experimental design*. McGraw Hill, 1962.

[WL02] Brian Whitman and Steve Lawrence. Inferring descriptions and similarity for music from community metadata. In *Proceedings of the International Computer Music Conference (ICMC)*, Göteborg, Sweden, 2002.

[XBF+08] Shengliang Xu, Shenghua Bao, Ben Fei, Zhong Su, and Yong Yu. Exploring folksonomy for personalized search. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 155–162, Singapore, Singapore, 2008. ACM.

[XFMS06] Zhichen Xu, Yun Fu, Jianchang Mao, and Difu Su. Towards the semantic web: Collaborative tag suggestions. In *Proceedings of the Collaborative Web Tagging Workshop (WWW '06)*, Edinburgh, Scotland, 2006.

[YGK+06] Kazuyoshi Yoshii, Masataka Goto, Kazunori Komatani, Tetsuya Ogata, and Hiroshi G. Okuno. Hybrid collaborative and content-based music recommendation using probabilistic model with latent user preferences. In *ISMIR '06: Proceedings of the 7th International Conference on Music Information Retrieval*, pages 296–301, Victoria, Canada, 2006.

[Zol07] Alla Zollers. Emerging motivations for tagging: Expression, performance, and activism. In *Workshop on Tagging and Metadata for Social Information Organization (WWW '07)*, Banff, Canada, 2007.