Contextualized Recommendations for the Socio-Semantic Web

Vom Fachbereich Informatik der
Technischen Universität Kaiserslautern
zur Verleihung des akademischen Grades
Doktor der Ingenieurwissenschaften (Dr.-Ing.)

genehmigte Dissertation

von
Dipl.-Inf. Rafael Schirru
aus
Lahnstein

Erstgutachter: Prof. Dr. Prof. h.c. Andreas Dengel
Zweitgutachter: Prof. Dr. Sandra Zilles
Vorsitzender: Prof. Dr. Paul Lukowicz
Dekan: Prof. Dr. Arnd Poetzsch-Heffter

Tag der mündlichen Prüfung: 2013/05/08

Zeichen der TU Kaiserslautern im Bibliotheksverkehr: D 386

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit mit dem Titel

Contextualized Recommendations for the Socio-Semantic Web

selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Die Arbeit wurde in dieser oder ähnlicher Form noch in keinem anderen Prüfungsverfahren eingereicht.

Berlin, den 10.05.2013

Rafael Schirru

Abstract

In recent years, recommender systems have been widely used for a variety of different kinds of items such as books, movies, and music. However, current recommendation approaches have often been criticized to suffer from overspecialization thus not enough considering a user's diverse topics of interest. In this thesis we present a novel approach to extracting contextualized user profiles which enable recommendations taking into account a user's full range of interests. The method applies algorithms from the domain of topic detection and tracking to automatically identify diverse user interests and to represent them with descriptive labels. That way manual annotations of interest topics by the users, e. g., from a predefined domain taxonomy, are no longer required. The approach has been tested in two scenarios: First, we implemented a content-based recommender system for an Enterprise 2.0 resource sharing platform where the contextualized user interest profiles have been used to generate recommendations with a high degree of inter-topic diversity. In an effort to harness the collective intelligence of the users, the resources in the system were described by making use of user-generated metadata. The evaluation experiments show that our approach is likely to capture a multitude of diverse interest topics per user. The labels extracted are specific for these topics and can be used to retrieve relevant on-topic resources. Second, a slightly adapted variation of the algorithm has been used to target music recommendations based on the user's current mood. In this scenario music artists are described by using freely available Semantic Web data from the Linked Open Data cloud thus not requiring expensive metadata annotations by experts. The evaluation experiments conducted show that many users have a multitude of different preferred music styles. However a correlation between these music styles and music mood categories could not be observed. An integration of our proposed user profiles with existing user model ontologies seems promising for enabling context-sensitive recommendations.

Recommender Systems, Socio-Semantic Web, Topic-based Resource Recommendations, Mood-based Music Recommendations

Zusammenfassung

In den letzten Jahren fanden Empfehlungssysteme für eine Vielzahl unterschiedlicher Objekte wie Bücher, Filme und Musik, weite Verbreitung. Aktuelle Ansätze werden dabei häufig kritisiert an dem Problem der Überspezialisierung zu leiden und somit die diversen Interessensgebiete eines Benutzers nicht hinreichend einzubeziehen. In dieser Arbeit stellen wir einen neuen Ansatz zur Extraktion kontextualisierter Benutzerprofile vor. Diese Profile ermöglichen Empfehlungen, welche die unterschiedlichen Interessen eines Benutzers berücksichtigen. Die Methode wendet Algorithmen aus der Domäne der Themenextraktion und -verfolgung an, um diverse Benutzerinteressen automatisch zu erkennen und repräsentiert diese mit einem beschreibenden Label. Auf diese Weise werden keine händischen Annotation von Interessensgebieten, z.B. aus einer vorgegebenen Domänen-Taxonomie, durch die Benutzer mehr benötigt. Der Ansatz wurde in zwei Szenarien getestet: Zunächst haben wir ein inhaltsbasiertes Empfehlungssystem für eine Enterprise-2.0-Resource-Sharing-Plattform implementiert, in dem die kontextualisierten Benutzerprofile verwendet wurden, um Empfehlungen mit einem hohen Grad an Themen-Diversität zu generieren. Zur Beschreibung der Inhalte im System sollte die kollektive Intelligenz der Benutzer genutzt werden, indem die Ressourcen durch Benutzer-generierte Metadaten beschrieben wurden. Die durchgeführte Evaluation hat gezeigt, dass unser Ansatz eine Vielzahl unterschiedlicher Benutzerinteressen erkennen kann. Die extrahierten Labels sind spezifisch für die erkannten Themen und können verwendet werden, um thematisch passende Ressourcen zu finden. Als Zweites wurde eine leicht veränderte Variante des Algorithmus getestet, um Musikempfehlungen zu generieren, die die aktuelle Stimmung des Benutzers berücksichtigen. In diesem Szenario wurden Künstler mittels frei verfügbarer semantischer Daten aus der Linked Open Data Cloud beschrieben, so dass teure Metadaten-Annotationen durch Experten nicht mehr benötigt werden. Die Evaluationsexperimente haben gezeigt, dass viele Benutzer unterschiedliche Musikrichtungen hören, jedoch konnte eine Korrelation zwischen diesen Musikrichtungen und bestimmten Stimmungen nicht beobachtet werden. Eine Integration der von uns vorgeschlagenen Benutzerprofile mit existierenden Benutzermodell-Ontologien scheint vielversprechend, um kontextsensitive Empfehlungen zu ermöglichen.

Empfehlungssysteme, Socio-Semantic Web, themenbasierte Empfehlungen von Ressourcen, stimmungsbasierte Musikempfehlungen

Acknowledgments

This thesis would not have been possible without the help of many different people. I want to thank Stephan Baumann for his inexhaustible support during the past years, for all the controversial and fruitful discussions as well as proof-reading of the thesis. Further, I'd like to thank Professor Sandra Zilles for proof-reading, helpful suggestions and guidance. Special thanks belong to Professor Andreas Dengel for his feedback on my work and for providing me with an environment that made this thesis possible. Further I want to thank my colleagues Tatjana Scheffler, Michael Sintek, and Thomas Roth-Berghofer for guiding me with my first scientific publications. I also want to thank the professors and attendees of the doctoral symposium at the RecSys 2010 conference for their critical questions and their encouragement to continue my work on diversification in recommender systems. My sincere thanks go to my student workers Bernhard Streit for his work on artist similarities based on metadata from the Semantic Web and to Christian Freye for his work on the context-based identification of music preferences. Special thanks belong to my family for their support during the years and to my friends Karsten Adler, Peter Kretzschmar, Bernhard Streit, Inessa Seifert, and Kinga Schumacher for believing in me.

This research has been financed to a large extend by the IBB Berlin in the project "Social Media Miner", and co-financed by the EFRE fonds of the European Union.

Contents

List of Figures

List of Tables

Part I

Foundations

Chapter 1

Introduction

Nowadays, recommender systems are omnipresent on the World Wide Web. They support people to discover a variety of different kinds of items such as music, books, and movies in a vast and almost unmanageable information space. The two predominant recommendation technologies applied today are content-based and collaborative filtering. Content-based recommender systems have their roots in information retrieval research [Belkin and Croft, 1992]. They recommend items that are similar to those items a user has liked in the past. Collaborative filtering was invented in the early 1990ies [Goldberg et al., 1992, Resnick et al., 1994] and has been adapted successfully in many large scale online portals like Amazon[1] and Last.fm.[2] Collaborative filtering systems recommend such items that an active user's peers have preferred in the past.

With the advent of the Web 3.0 new and exciting opportunities for recommender systems have emerged. [Wahlster et al., 2006] defined the Web 3.0 as the convergence of the Web 2.0 [O'Reilly, 2005b] and the Semantic Web [Berners-Lee et al., 2001]. Recommender systems can profit from both worlds. On the one hand we have the Web 2.0 comprising technologies such as wikis, blogs, and resource sharing platforms in which we find user-generated content and social metadata such as tags. Tagging is a lightweight approach to collaboratively categorize items being widely adapted due to its low cognitive cost [Sinha, 2005]. Despite well-known problems like synonymous and ambiguous tags or tag assignments on different levels of specificity ([Begelman et al., 2006]), tags still provide a valuable

[1] http://www.amazon.com/

[2] http://www.last.fm/

source of information about items that can be used for recommendations. On the other hand the Semantic Web provides us with structured metadata that can be understood by machines. The Linking Open Data project[3] identifies existing data sets under open licenses and publishes them on the Web, according to the Linked Data principles (see Chapter 3.2). This data can be used to obtain descriptions for items from different domains (e. g., music, movies, and books), that way opening new possibilities for content-based recommender systems.

Traditional content-based and collaborative recommender systems tend to recommend similar items predominantly (e. g., [Bradley and Smyth, 2001, Zhang and Hurley, 2009]) thus not taking a user's full range of interests into account. In this thesis we propose an approach extracting contextualized user interest profiles that can be used to enhance the diversity of recommendation lists by recommending items from different user interest topics. So far, recommender systems have mostly been evaluated by assessing each recommended item separately and calculating an aggregated score for the whole system. However judging recommendation lists as a whole has been identified as an important issue in recent years [McNee et al., 2006a]. It has also been addressed in previous work, e. g., by [Ziegler et al., 2005] showing that recommendation lists with a higher degree of diversity can improve user satisfaction with the recommender system. In Section 1.1 we introduce our idea of topic-based recommendations in Enterprise 2.0 resource sharing platforms. Our goal here is the provision of recommendations according to a knowledge worker's full range of interests. Next, in Section 1.2 we present another application domain for our algorithm. In the second use case we identify the different music styles a user prefers, that way aiming at context-sensitive music recommendations based on the present mood of the active user.

1.1 Topic-based Resource Recommendations

Nowadays, social media technologies are increasingly often deployed to foster the knowledge transfer in the Enterprise. McAfee introduced the concept of the Enterprise 2.0 as a collection of Web 2.0 technologies for generating, sharing, and refining information [McAfee, 2006]. Companies can buy or build these technologies in order to uncover the practices and outputs of their knowledge workers. At the German

[3] http://esw.w3.org/SweoIG/TaskForces/CommunityProjects/LinkingOpenData

Research Center for Artificial Intelligence[4] we have developed the ALOE[5] system, a social resource sharing platform for bookmarks, files, and their associated metadata that can be deployed in such scenarios.

As the amount of content in these information systems grows, there is an increasing need for recommender systems that keep the users informed about resources matching their needs and preferences. However, traditional recommender systems based on collaborative filtering suffer from sparsity issues, particularly in scenarios where the amount of items is much larger than the amount of users. Content-based recommender systems on the other hand suffer from the overspecialization problem thus not considering a user's full range of interests (e. g., [Adomavicius and Tuzhilin, 2005]). Let's assume a knowledge worker is interested in Java programming, Perl scripting, and Linux operating system. She uses an Enterprise 2.0 resource sharing platform to share resources with her colleagues according to these topics. We envision a recommender system that identifies these topics and provides recommendations accordingly.

After a warm-up phase in which traditional item-based collaborative filtering recommendations are provided for new users, our proposed approach applies algorithms from the domain of topic detection and tracking to identify a knowledge worker's different topics of interest. The method analyzes the metadata profiles of the user's preferred resources and derives per topic a weighted term vector as a label. When the user requests recommendations these vectors are used to query an index in order to find previously unknown resources matching the respective interest topics.

The underlying approach (namely clustering and cluster label extraction) has further been applied in a use case for music recommendations based on the present mood of the active user. The idea will be introduced subsequently.

1.2 Mood-based Music Recommendations

Music classification and recommendation based on mood has been a growing research area in recent years (e. g., [Rho et al., 2009, Lee and Lee, 2006]). This is also reflected in the MIREX (Music Information Retrieval Evaluation eXchange) challenge[6] where the Audio Mood Classification task has been added in 2007 ([Hu et al., 2008]).

[4] `http://www.dfki.de/`

[5] `http://aloe-project.de/AloeView/`

[6] `http://music-ir.org/mirex/wiki/MIREX_HOME`

In this thesis we investigate two assumptions: First, we examine the hypothesis that many people listen to different styles of music in terms of genres, instruments, release years, etc. While this hypothesis might seem intuitively plausible, current music recommender systems do not particularly consider the different music styles a user prefers. Many music recommender systems are based on collaborative filtering techniques that tend to recommend similar items only. Other recommender systems for entertainment items try to infer a user's taste from identified personality traits [Hu and Pu, 2009]. Advocates of theses systems often claim that a user might prefer different music styles within a given genre, however preferences across various very different genres are often neglected. Second, we analyze whether a user's currently preferred music style depends on her present mood. There is evidence in the literature that a user's music preferences change depending on her mood (e. g., [Mortensen et al., 2008]). For that reason we try to find correlations between the user's current mood and her preferred music style in terms of music attributes such as genres and instruments. In our approach we use metadata from the Semantic Web to describe a user's preferred artists. These artists are then clustered according to the music styles they are associated with. Then we check whether the identified groups overlap with mood categories we found in the literature. The primary goal for this scenario is the provision of context-sensitive music recommendations. Depending on the user's mood we aim at recommending items from the music style which is most appropriate in the given situation.

In contrast to traditional collaborative filtering methods that tend to recommend popular items and are not well suited for users with extraordinary tastes ([McNee et al., 2006b]) approaches based on metadata from the Semantic Web can provide recommendations with a high degree of novelty without the need of finding peers with a similar taste for the active user [Baumann et al., 2010].

1.3 Research Hypotheses

For topic-based resource recommendations we will analyze the following hypotheses:

H1 Knowledge workers have different topics of interest.

H2 By applying topic detection algorithms on the users' preferred resources we can detect these topics.

H3 The detected topics can be used to generate recommendation lists with a high degree of diversity.

To the best of our knowledge these hypotheses have not been examined in the Enterprise 2.0 context so far. For mood-based music recommendations we will investigate the following hypotheses:

H4 Many people listen to different styles of music.

H5 An active user's preferred style of music depends on her mood.

1.4 Outline

The thesis is divided into four major parts: Part I describes the motivation for this work and provides the reader with relevant information about the environment in which the thesis is set.

Chapter 2 presents the state of the art in recommender systems. It gives an overview of the different kinds of recommendation algorithms with content-based and collaborative approaches as the predominant ones. Different hybridization methods are discussed that combine the strengths of single recommendation methods while at the same time alleviating their deficiencies. Finally, the chapter states two areas where the extraction of user profiles and recommendation algorithms proposed in this thesis go beyond the current state of the art.

Chapter 3 describes the idea of the Web 3.0 as the convergence of the Web of people (i. e., the Web 2.0) and the Web of data (i. e., the Semantic Web). A focus is put on collaborative tagging systems and the Enterprise 2.0 as well as Linked Data as these concepts are particularly important for the use cases in which the proposed recommendation approaches have been applied.

Chapter 4 depicts the ALOE system, an Enterprise 2.0 platform that has been implemented in the Knowledge Management group at DFKI as well as the C-LINK system which is a conference organization system that has been built on top of ALOE. In C-LINK content-based recommendations for conference events and participants have been implemented. The chapter discusses the C-LINK system as well as lessons learned.

In Part II we present the proposed approaches together with their evaluations.

Chapter 5 introduces our approach for topic-based resource recommendations in Enterprise 2.0 platforms. It depicts in detail our idea on how to extract user profiles that appropriately represent a user's different topics of interest as well as the derived content-based recommender system.

Chapter 6 describes our method extracting multifaceted profiles representing a user's different preferred music styles. First, we depict Semantic Web data sources that can be used to describe music artists. Then we present our approach to extracting contextualized music preference profiles and suggest possible fields of application.

Chapter 7 covers the evaluation of the ideas presented in this thesis. First, it presents goals from the literature which are commonly addressed when recommender systems are evaluated. Then it explains the reasoning behind the evaluation method applied. Next, the results for our topic-based resource recommendations and mood-based music recommendations are presented respectively.

Chapter 8 presents research in the fields of topic-based resource recommendations as well as mood-based music recommendations.

Part III concludes the findings of our work and presents ideas for future work.

Chapter 9 summarizes the work performed in this thesis, discusses the research hypotheses that have been set in the introduction, and depicts our research contributions. Finally, limitations of the approach as well as the evaluation experiments conducted are discussed.

Chapter 10 suggests ideas for future work. In particular we discuss the use of our multifaceted user profiles for context-sensitive recommendations.

In the appendix we present background knowledge and technical details of the techniques used in this work.

Appendix A depicts similarity and distance measures that are often used for recommender systems or clustering algorithms to determine the similarity between users, items, or between whole item sets.

Appendix B describes statistical error measures that are commonly used to evaluate the quality of clustering results or the accuracy of recommendation algorithms.

Appendix C provides an overview of clustering algorithms. First, hierarchical clustering methods are described that allow for a visual representation of clustering results. Second, we present the K-Means algorithm as an example for flat clustering together with a heuristic to determine a reasonable cluster number. Third, co-clustering techniques are depicted that perform clustering of the rows and columns of a matrix at the same time. Finally, we present two methods that are frequently used to extract cluster labels.

Appendix D presents two schemes for music mood representations based on social tags.

Appendix E summarizes important aspects for the evaluation of recommender systems and of collaborative filtering in particular. It compares live user experiments to offline analyses and the use of synthesized vs. natural data sets. Further properties of recommender system data sets are discussed and different evaluation measures are presented.

Chapter 2

State of the Art in Recommender Systems

Providing useful recommendations is a challenging task. A recommender system needs to learn about the preferences of users as unobtrusively as possible and provide real-time recommendations in a vast information space of potentially millions of items. Following [Adomavicius and Tuzhilin, 2005] we state the recommendation problem as follows: Let U be the set of all users and I be the set of all items in the system. Let φ be a utility function to measure the usefulness of item i to user u, $\varphi : U \times I \rightarrow R$ with R being a totally ordered set. For each user $u \in U$ we want to select items $i' \in I$ that maximize the utility for the user:

$$\forall u \in U, i'_u = \underset{i \in I}{\arg\max} \varphi(u, i) \tag{2.1}$$

As depicted in [Montaner et al., 2003] there are three main information filtering methods for recommender systems on the Web today: content-based, collaborative and demographic filtering. Furthermore hybrid approaches exist that combine several techniques in order to overcome the particular weaknesses of the individual approaches [Burke, 2002]. Subsequently content-based and collaborative filtering are briefly introduced.

For both techniques memory-based and model-based approaches exist. In large scale environments with millions of items and millions of users memory-based approaches often suffer performance issues. Model-based algorithms remedy these problems. They use the set of available ratings to learn a model offline which can be used to make on demand rating predictions. In order to depict the functionality of the algorithms we mainly concentrate on the memory-based approaches.

2.1 Content-Based Approach

Content-based (CB) methods estimate the utility of an item according to its similarity to items for which the user has expressed a preference in the past. The content-based methods have their roots in information retrieval and information filtering research hence being prevailingly implemented for resources having textual features available (either directly extracted from documents or in the form of metadata). However content-based recommender systems have also been implemented for multimedia items for which no textual features are available. For instance, [Liu and Huang, 2000] and [Logan and Salomon, 2001] implemented content-based retrieval of similar audio items by using automatically extracted signal features such as mel-frequency cepstral coefficients. Subsequently we will focus on content-based systems for items for which textual features are available. We will use the expressions "items with textual features" and "metadata profiles of items" synonymously with the term "documents."

For every item that can possibly be recommended content-based approaches compose a profile ($\mathit{ItemProfile}\,(i)$) consisting of its features (i. e., the attributes that characterize the item). For textual resources this is usually a set of keywords with an "importance" weight attached. For this weight the term frequency/inverse document frequency measure is widely used [Sparck Jones, 1972]:

Let N be the number of all items that can be recommended and k_a a keyword which is contained in n_a of them. Further $f_{a,b}$ is the number of times k_a is contained in i_b. The term frequency is computed as follows:

$$TF_{a,b} = \frac{f_{a,b}}{\max_z f_{z,b}} \tag{2.2}$$

with the maximum computed over the frequencies $f_{z,b}$ of all keywords which are contained in item i_b. Keywords that appear in many documents are not discriminative. The measure of the inverse document frequency ($IDF_a = \log \frac{N}{n_a}$) is used to cope with this problem. The combined TF-IDF weight for a keyword is determined by multiplying its term frequency in a document with its inverse document frequency:

$$w_{a,b} = TF_{a,b} \times IDF_a \tag{2.3}$$

An item profile might be composed of these weights:

$$\mathit{ItemProfile}\,(i) = (w_{1i}, w_{2i}, ..., w_{Ki}) \tag{2.4}$$

with K as the total number of all keywords. Further, we need for every user a profile

$$\mathit{UserProfile}\,(u) = (w_{u1}, w_{u2}, ..., w_{uK}) \tag{2.5}$$

that describes her interests. Usually this profile is composed of weighted terms from items for which the user has expressed a preference in the past.

The utility function is defined as follows:

$$\varphi\,(u, i) = \mathit{score}\,(\mathit{UserProfile}\,(u)\,, \mathit{ItemProfile}\,(i)) \tag{2.6}$$

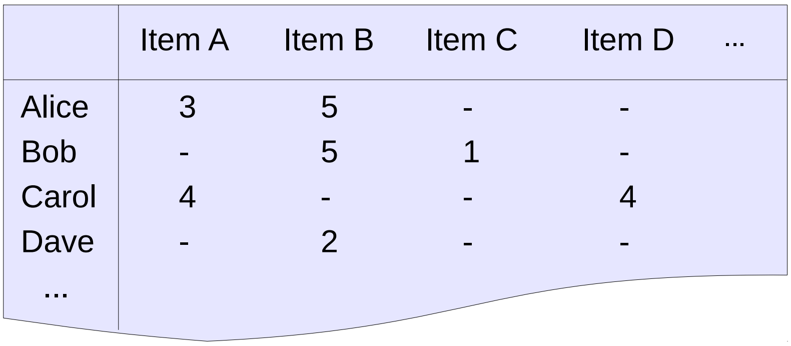
For $\mathit{ItemProfile}\,(i) = \overrightarrow{w_{item}}$ and $\mathit{UserProfile}\,(u) = \overrightarrow{w_{user}}$ a common scoring heuristic is the cosine similarity measure:

$$\varphi\,(u, i) = \cos\,(\overrightarrow{w_{user}}, \overrightarrow{w_{item}}) = \frac{\overrightarrow{w_{user}} \cdot \overrightarrow{w_{item}}}{\|\overrightarrow{w_{user}}\|_2 \times \|\overrightarrow{w_{item}}\|_2} \tag{2.7}$$

A content-/model-based approach for website recommendations is presented in [Pazzani and Billsus, 1997]. In their system *Syskill & Webert* users can initialize their profiles by assigning keywords and probabilities of their occurrences for a topic in order to distinguish interesting pages on the topic from uninteresting ones. This profile is later revised when users provide ratings for Web pages they have visited. Website recommendations are treated as a classification problem that is solved with a naïve Bayesian classifier.

2.2 Collaborative Filtering

The concept of collaborative filtering (CF) was first introduced by [Goldberg et al., 1992]. In the Tapestry system eager users are expected to annotate eMails (in particular newsgroup messages). More casual users will profit



	Item A	Item B	Item C	Item D	...
Alice	3	5	-	-	
Bob	-	5	1	-	
Carol	4	-	-	4	
Dave	-	2	-	-	
...					

Figure 2.1: User preferences are represented in a user-item rating matrix.

from these annotations and will read messages based on the reviews. The casual users install filters that use these annotations, documents matching such filters will be returned.

Modern collaborative filtering systems calculate recommendations based on the users' explicit or implicit ratings for items in a system. Two approaches are distinguished: the user-based and the item-based collaborative filtering method. Both will be introduced subsequently.

2.2.1 User-Based Collaborative Filtering

User-based collaborative filtering systems recommend items that users with similar tastes as the active user liked in the past (e. g., [Resnick et al., 1994, Konstan et al., 1997, Shardanand and Maes, 1995]). User preferences are usually represented in a user-item rating matrix (see Figure 2.1). Memory-based approaches predict ratings based on the entire set of available rating values of all users in the system. The process is divided into two steps. First the similarity between the active user and all other users that rated a predefined number of items in common with the active user is computed. In collaborative filtering, Pearson correlation and cosine similarity are commonly used similarity measure for this purpose. The former will be described subsequently, the letter is described in Appendix A. Let r_{ai} be the rating value of user u_a for item i and $\bar{r}_a$ be the average rating of user u_a. Further let I_{ab} be the set of items that both users u_a and u_b rated. Then the Pearson correlation between the two users can be computed as follows.

$$sim_{Pearson}\left(u_a,u_b\right)=\frac{\sum_{i\in I_{ab}}\left(r_{ai}-\bar{r}_a\right)\left(r_{bi}-\bar{r}_b\right)}{\sqrt{\sum_{i\in I_{ab}}\left(r_{ai}-\bar{r}_a\right)^2}\sqrt{\sum_{i\in I_{ab}}\left(r_{bi}-\bar{r}_b\right)^2}} \tag{2.8}$$

A neighborhood of users that are responsible for the rating predictions of the current user can be selected by choosing the k most similar users to the current user.

In the next step the ratings of the nearest neighbors are aggregated to calculate rating predictions for the current user. Let $\hat{U}$ be the set of neighbors of the current user. A rating prediction for an item can be calculated as follows:

$$r_{u,i}=\lambda\sum_{u'\in\hat{U}} sim(u,u')\cdot r_{u',i} \tag{2.9}$$

with normalization factor $\lambda = 1/\sum_{u'\in\hat{U}}|sim(u,u')|$.

2.2.2 Item-Based Collaborative Filtering

In large scale Web platforms with millions of users and millions of items, user-based collaborative filtering as described above suffers serious performance issues. Item-based collaborative filtering is a technique that helps to overcome these performance problems while at the same time providing better recommendation accuracy [Sarwar et al., 2001]. The approach is based on the assumption that item similarities are more static than similarities between users. For that reason the pairwise similarities between items can be computed offline and can then be accessed quickly when recommendations have to be provided.

In item-based collaborative filtering only the similarities between items are computed that are co-rated by at least a predefined number of users. The calculations are performed by using the column vectors of the user-item rating matrix. Frequently used similarity metrics are again cosine similarity as well as correlation-based similarity. In [Sarwar et al., 2001] it was shown that the adjusted cosine similarity (see Appendix A) can significantly reduce the mean absolute error thus improving recommendation accuracy.

Once the similarities between the items are calculated we get for the current item the most similar items and isolate the items which are co-rated by the active user.

Let $\hat{I}$ be the set of co-rated items. The rating prediction for the current item can be calculated as follows:

$$r_{u,i} = \frac{\sum_{i' \in \hat{I}} sim(i, i') \cdot r_{u,i'}}{\sum_{i' \in \hat{I}} |sim(i, i')|} \tag{2.10}$$

The perhaps most popular item-based recommender system might be the one of online retailer Amazon [Linden et al., 2003].

2.2.3 Improving Collaborative Filtering - The Netflix Competition

In October 2006 the online DVD retail service Netflix[1] announced a competition exposing a price of 1,000,000 USD for the best team implementing a collaborative filtering algorithm that could achieve a reduction of the root mean squared error (RMSE, see Appendix B.3) by at least 10% compared to their own Cinematch algorithm. They released a training data set of 100,480,507 ratings by 480,189 users for 17,770 movies. Each rating was also associated with a timestamp indicating when the rating was contributed. Further a separate data set was provided containing the titles and release years of the movies. In the view of privacy concerns no information about the users was given [Wikipedia, 2010c]. The team "BellKor's Pragmatic Chaos" was announced as the winner of the competition on September 18, 2009. Their algorithm is described in detail in [Koren, 2009].

[Amatriain, 2009] summarizes what the collaborative filtering community has learned from the competition:

- *RMSE is not a valid success measure:* There is no direct correlation between RMSE and the end-user satisfaction with the recommender system (see also the discussion on this topic in Appendix E).

- *Time matters:* Modeling the temporal evolution of user preferences has been found to be of major importance. Just because someone liked the first Harry Potter book in 1997 does not necessarily mean that the person will like the seventh book of the series today. It does not even mean that he/she still likes the first book today.

[1] `http://www.netflix.com/`

- *Matrix factorization methods work best:* Methods such as singular value decomposition and non-negative matrix factorization do not only improve the recommendation results but they also provide insights into the problem and most importantly they can be implemented in a very efficient way. We discuss these techniques in the context of clustering in Appendix C.3.
- *One method is not enough (nor 100):* To improve the rating predictions of a system it is usually easier to add another prediction method instead of trying to improve the old one. In the winning solution of the Netflix competition many predictors have been blended requiring the learning of millions of parameters thus bringing the algorithm close to being a black box. The scientific insights and the knowledge learned from it was very limited and the portability of the approach is rather questionable.
- *The importance of data and noise:* Good improvements with rating predictions can be made when the data is first cleaned from noise, e. g., when asking the users to re-rate some items.

2.3 Discussion of Content-Based and Collaborative Approaches

Content-based and collaborative information filters both have their particular strengths and weaknesses. We will summarize them in this section according to [Adomavicius and Tuzhilin, 2005]. A common issue of both approaches is the *new user problem*. Providing high quality recommendations to a user is only possible when the user has expressed preferences for a sufficiently large amount of items. In CF there is also the *new item problem*, i. e., items for which a minimum number of ratings is not available cannot be recommended. CB approaches select items depending on their features thus not suffering this problem. However the need for features (e. g., text or manually annotated features) is also a drawback of content-based approaches as such features might not always be available, e. g., features describing the content of video data or images. Also content-based approaches cannot take into account the *quality* of items. Items with similar content but different in quality cannot easily be distinguished. As collaborative filtering relies on the ratings of items the approach inherently considers the quality of items. Another problem of content-based approaches is *overspecialization.* CB recommender systems suggest items that are thematically similar to items for which a user has expressed a preference in the past that way missing interesting items from other topics. Collaborative filtering is capable of recommending items from different topics thus increasing the serendipity aspect of the system. However CF cannot always be applied. In systems where the user-item rating matrix is sparsely populated the recommendation quality of collaborative information filters decreases significantly.

2.4 Model-based Approaches

Recommender systems face many challenges. They have to produce high quality recommendations, perform many recommendations per second for potentially millions of users and items, and they have to achieve high coverage in spite of data sparsity. Pure memory-based approaches as described before suffer scalability issues when the number of users and items becomes too large. It is for this reason that approaches have been implemented which learn a model of the available data (offline) in order to enable efficient recommendations for systems with many users and items.

[Sarwar et al., 2000] present a model-based collaborative filtering algorithm that applies Latent Semantic Indexing/Singular Value Decomposition to reduce the dimensionality of the data in recommender systems. They use the low dimensional representation of the data to compute the neighborhood of the active user, that way improving the efficiency of the collaborative filtering algorithm. Two experiments were conducted: in the first experiment rating predictions were calculated, in the second Top-N recommendation lists were generated. The evaluation of the system shows that the approach results in good quality predictions and has the potential to provide better online performance than pure memory-based approaches.

Model-based, content-based recommender systems are described, e. g., by [Pazzani and Billsus, 2007]. They present the task of learning a user model for content-based recommendations as a form of classification learning. In such systems user feedback on items is used as training data for classification learners. The data is divided into categories such as “items the user likes” and “items the user doesn’t like”. Content descriptions of items can either be structured or unstructured (free text). The classification learners try to predict whether a user will like an item or not. Many of these algorithms also provide an estimate of the probability that the user will like an unseen item. These estimates can be used to rank a list of recommendation candidates. Some algorithms directly try to predict a user's degree of interest by providing numeric values that estimate a user's rating for an unseen item. As suitable machine learning algorithms for model-based, content-based recommendations Pazzani and Billsus present among others: decision trees (prevailingly for structured data with few attributes), rule induction (particularly suited for semi-structured data), Rocchio's algorithm, linear classifiers, and probabilistic methods such as naïve Bayes.

2.5 Further Approaches

Knowledge-based recommender systems are closely related to Case Based Reasoning research. Their four main characteristics are: centrality of examples, conversational navigation via tweaks, knowledge-based similarity metrics and task-specific retrieval strategies. The FindMe approach as proposed, e. g., in [Burke, 2000], has two fundamental retrieval modes: The first is similarity finding. In this mode the user selects an item from a catalog and requests similar items. Alternatively it may also be possible to just specify desired features of the searched items. Second is the tweak mode where features of candidate items can be adjusted (e. g., lower price) in order to get better recommendations. In contrast to collaborative information filters, knowledge-based recommender systems do not a have a ramp-up problem and also do not suffer from the new user problem as the user directly tells the system what kind of item she is searching for. However the knowledge engineering task (i. e., describing items with high-quality up-to-date data) is often a bottleneck in these systems.

Another type of recommender systems uses *demographic information* (such as gender, age, or education) to identify types of users that like certain objects. However obtaining demographic information can be difficult. [Pazzani, 1999] trains a classifier for each recommendable item with the homepages of users that liked the respective item and the homepages of users that didn't like the item.

Utility-based recommender systems try to capture a user's preferences in a utility function (cf. Equation 2.1). While with other recommendation approaches this function is usually known in advance and the same for all users, learning such a function for each user is the biggest issue of utility-based recommender systems.

[Yi and Deng, 2009] propose an approach for utility-based recommendations in E-Commerce based on Bayesian networks. User utilities are represented as probabilities over attributes. First a common utility function for all users is build by a domain expert. A prior Bayesian network is established based on this function. In a second step the Bayesian network is adapted according to the implicit feedback that a user provides for items (e. g., the user purchases an item or saves a reference to an item). The utility function that is learned that way, will be used to recommend items that are supposed to have the highest utility for a user. An important advantage of utility-based recommender systems is that they can incorporate item features that are not related to the item itself (such as delivery schedule or warranty terms) but are important to the user.

2.6 Hybrid Systems

To overcome the limitations of individual recommendation approaches and to exploit the advantages of two or more methods, often hybrid systems are built. [Burke, 2002] describes seven methods to combine different recommendation algorithms:

Weighted In a weighted hybrid recommender system the results of all available recommendation techniques are combined to calculate the score of an item. In its simplest form the final score is a linear combination of the recommendation scores of the available methods. In case that some or all of the involved methods do not produce a rating prediction score the recommendations of each method may be considered as votes for the respective items. In the hybrid system these votes will then be combined in order to produce the final recommendation list. There are systems that also adapt the influence of single recommendation techniques based on the user feedback for the recommended items. A drawback of the weighted hybridization method is its implicit assumption that the relative value of each recommendation technique is more or less uniform across the item space. However this assumption does not always hold true as, e. g., collaborative approaches perform worse on seldom rated items.

Switching [Tran and Cohen, 2000] present a hybrid recommender system that switches between knowledge-based and collaborative recommendations based on a predefined criterion. New users receive knowledge-based recommendations. As soon as a sufficiently large user profile of preferred items is available the system can also provide recommendations based on a collaborative recommendation method. The system determines automatically whether the knowledge-based or the collaborative method can provide the most useful recommendations. For that purpose it calculates rating predictions according to the collaborative method. If the average rating prediction exceeds a predefined threshold the recommendations of the collaborative method will be presented to the user, otherwise the knowledge-based recommendations will be used. If the actual user ratings for the knowledge-based recommendations are worse than the average rating prediction of the collaborative method the threshold for the switching criterion will be lowered and the selection process starts over. Another switching criterion might be the confidence of the recommendations provided by single methods (see Appendix E.5).

Mixed In cases where many recommendations are needed simultaneously, it might be useful to have a mixed hybrid that presents recommendations from different systems. For instance, a mixed hybrid of collaborative and content-based filters is likely to overcome the new item problem as the content-based technique does not suffer from this problem. It might be able to alleviate the overspecialization problem of pure content-based approaches as the collaborative filter is likely to also recommend interesting items from other domains than those for which the user already has expressed a preference. However it still suffers from the new user problem as both content-based as well as collaborative recommender systems have this problem.

Feature Combination The feature combination approach allows to mix content-based and collaborative filters by treating user ratings as additional features for items. It applies content-based techniques on the augmented data set. Such a hybrid allows the consideration of collaborative data without relying exclusively on it. That way it reduces the sensitivity of the system for the number of users that rated an item. Moreover the content-based features allow the system to have information about the inherent similarity of items that would otherwise be concealed by pure collaborative approaches.

Cascade The cascade hybrid involves a staged process. First, one recommendation technique is employed to produce a roughly ranked list of candidate items. In a second step another recommendation technique is used to refine the recommendations from the candidate set. Two advantages of the cascade hybrid should be highlighted here: The first advantage is that cascading allows to avoid employing the second recommendation technique on items that are already well-differentiated by the first, higher-priority, technique or not sufficiently often rated. Second the cascade hybrid is tolerant of noise produced by the second, lower-priority technique as the second technique can only refine, not upset the results of the first technique.

Feature Augmentation Feature augmentation hybrids take rating predictions or classifications of items as additional input for the next recommendation step. Burke mentions the LIBRA system [Mooney and Roy, 2000] as an example for such a hybrid. LIBRA is a content-based book recommender system that uses the metadata that is associated with books (titles, authors, synopses, ...) to learn user profiles. The metadata incorporates collaborative content such as related authors and titles as determined by Amazons collaborative recommender system. Mooney and Roy have evaluated the role of the collaborative features and have found that they have a significant positive effect on the quality of the generated recommendations.

Meta-level The last kind of hybrid recommender systems that Burke presents, are the meta-level hybrids. These systems take the model learned by one recommendation technique as the input for the second technique. As one example Burke mentions the method of *collaboration via content* [Pazzani, 1999] that addresses the sparsity problem of traditional collaborative filtering systems. To determine the similarity between users, collaboration via content uses content-based user profiles. The profiles consist of weighted terms that indicate that a user will like an item. The prediction value of an item is calculated as the weighted average of all users' predictions for the item. Therefore, the correlation between profiles is used as weight.

2.7 Beyond the State of the Art

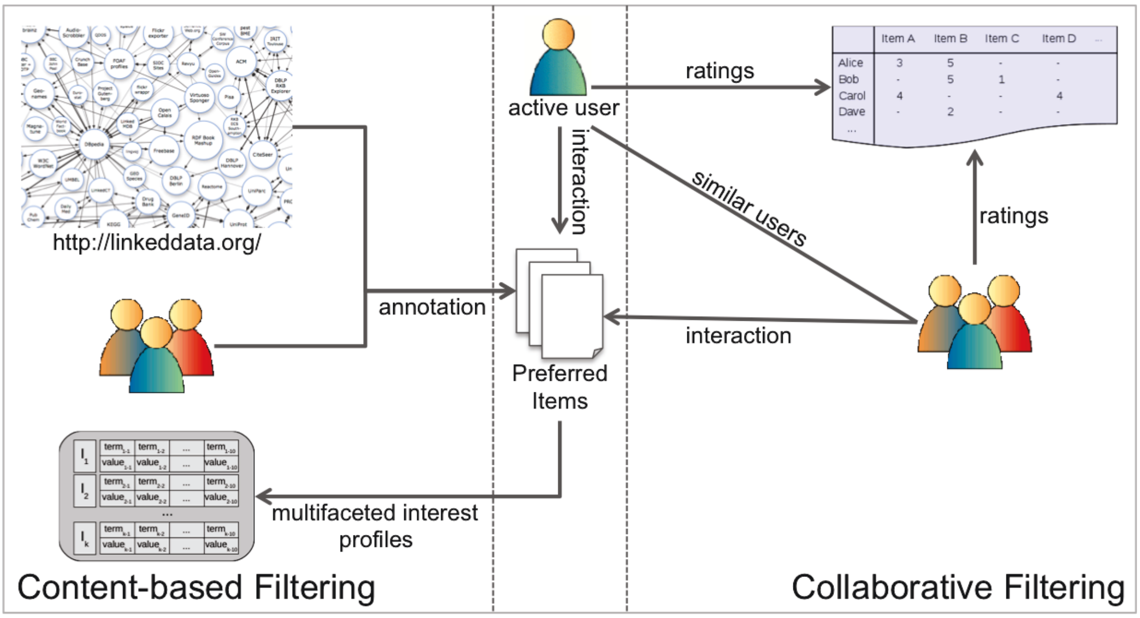
The methods for user profile extraction and item recommendations based on these profiles that are described in this thesis go beyond the current state of the art at least in two areas: First, we tackle the problem of limited content analysis for multimedia items. As described before capturing the topics of non-textual items automatically is still difficult today and thus complicates the use of content-based systems in scenarios where such items have to be recommended. We investigate the use of metadata annotated by the user community for the extraction of contextualized user profiles in an Enterprise 2.0 platform thus being able to provide content-based recommendations of items independent of their format. That way we can provide an alternative for scenarios where collaborative filtering recommendations are problematic due to sparsity issues of the user-item rating matrix. In the second use case analyzing the users' music preferences, we made use of Linked Data[2] to describe the artists a user prefers. Providing music recommendations based on manually annotated metadata has been done successfully in recent years. Platforms such as Pandora[3] have put a huge amount of effort into the manual annotation of music items. Exploiting freely available data from the Semantic Web could help to make recommendations based on metadata descriptions of multimedia items cheaper and applicable for a larger amount of items.

Besides the annotation of items with community metadata, the second issue tackled by our approach is the problem of overspecialization which is a major concern particularly for content-based recommender systems that tend to recommend items from the user's predominant interest topic mostly. By applying clustering algorithms on the profiles of a user's preferred items we can identify groups (topics of interest and preferred music styles) that cover a broad range of the user's preferences thus allowing us to improve the diversity of recommendation lists or to provide context-sensitive recommendations according to a user's current needs and preferences.

Figure 2.2 shows an overview of the data sources used for the extraction of the user profiles and for the recommendation algorithms proposed in this thesis. In Chapter 5 we present a switching hybrid recommender system providing collaborative filtering recommendations for new users and content-based recommendations based on our contextualized user profiles for users that interacted with the system over a longer period of time. In this scenario the items are annotated by the users of the Enterprise 2.0 platform. In the second scenario we describe a user's preferred artists by making use of Linked Data. The artists are clustered and for each group a label is extracted

[2] http://linkeddata.org/

[3] http://www.pandora.com



	Item A	Item B	Item C	Item D	...
Alice	3	5	-	-	
Bob	-	5	1	-	
Carol	4	-	-	4	
Dave	-	2	-	-	
...					

Figure 2.2: Data sources used for the extraction of contextualized user profiles and the recommendation algorithms proposed in this thesis.

describing the respective music style. In Chapter 6 we depict how the user profiles obtained that way can be used for the recommendation of internet radio stations and how they can be integrated into existing content-based recommender systems.

Chapter 3

The Web 3.0

In recent years the Web has evolved in two directions: First is the development towards a Web of people. With the arising of the Web 2.0 an increasing number of people have become producers of content and also of metadata describing the content on the Web. Second is the development towards a Web of data. The Semantic Web initiative aims at making data on the Web processible by computers in a meaningful way. In [Wahlster et al., 2006] the Web 3.0 is defined as the convergence of the Web 2.0 and the Semantic Web (see Figure 3.1).

The current chapter presents in detail the constituent parts of the Web 3.0. We depict in Section 3.1 the principles of the Web 2.0 and pick collaborative tagging systems as one phenomenon of the Web 2.0 that is of particular importance for our work. Further the concept of the Enterprise 2.0 is described as it constitutes an application domain for our topic-based recommendation approach. In Section 3.2

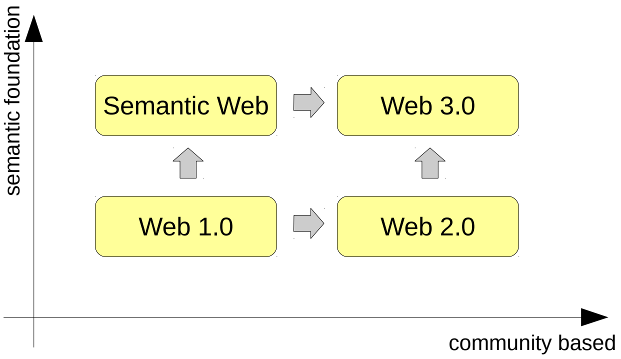


Figure 3.1: The Web 3.0 as the convergence of the Web 2.0 and the Semantic Web ([Wahlster et al., 2006]).

we depict the characteristics of the Semantic Web and briefly introduce the concept of Linked Data. Finally the convergence of the Web 2.0 and the Semantic Web in the Social Semantic Web is described in Section 3.3.

3.1 The Web 2.0

A turning point for the Web was marked with the bursting of the dot-com bubble in the autumn of 2001. The Web was judged as overrated by many people, but bubbles and consequent shakeouts are typical features of all technological revolutions [O'Reilly, 2005b].

The concept of the Web 2.0 started with a conference brainstorming session between O'Reilly and MediaLive International. As stated by Dale Dougherty, a Web pioneer and O'Reilly VP, the Web was far from having crashed. In contrast it was more important than ever. New applications and sites arose with a surprising regularity. It seemed that the companies that had survived the bursting of the dot-com bubble had certain things in common. [O'Reilly, 2005a] gives the following definition of the term Web 2.0:

> *"Web 2.0 is the network as platform, spanning all connected devices; Web 2.0 applications are those that make the most of the intrinsic advantages of that platform: delivering software as a continually-updated service that gets better the more people use it, consuming and remixing data from multiple sources, including individual users, while providing their own data and services in a form that allows remixing by others, creating network effects through an "architecture of participation," and going beyond the page metaphor of Web 1.0 to deliver rich user experiences."*

According to [O'Reilly, 2005b] seven principles which are included in the above definition and which characterize Web 2.0 applications will be presented subsequently.

The Web As Platform Based on the example of Google the first principle shall be explained. Google was started as a native Web application which was delivered as a service. The Web application itself was never packaged or sold and there were no scheduled software releases. The Web application was just continuously improved. The application was never licensed, just used. For this reason the application didn't have to be ported to different platforms so that customers could run the software on their own equipment. The application was run on a scalable collection of commodity PCs running open source operating systems in addition with homegrown applications and utilities which no one outside the company ever gets to see.

Harnessing Collective Intelligence The power of the Web to harness collective intelligence seems to be the central principle for those who survived the Web 1.0 era and are leaders in the Web 2.0 era. Segaran states that collective intelligence is about drawing new conclusions from independent contributors ([Segaran, 2007], page 2). A major difference between Amazon and competitors like Barnesandnoble.com is that Amazon made a science of user engagement. They have much more user reviews, participation possibilities on almost every page, and they make use of user activities to produce better search results.

Much attention has been received by sites like Delicious[1] which have pioneered a concept that is called "folksonomy". The term stands for a style of collaborative categorization of resources by making use of freely chosen keywords which are often referred to as tags. The concept of folksonomies will be described in more detail in Section 3.1.1.

Data is the Next Intel Inside A specialized data base constitutes the backend of every current significant internet application. This makes data base management a core competency of Web 2.0 companies. The importance of the data can again be illustrated by making use of the example of Amazon.com. Amazon's original data base came from the ISBN registry provider R. R. Bowker. In contrast to its competitors Amazon enhanced the data by adding publisher-supplied data (e. g., cover images, table of contents, and sample material) and they further encouraged their users to annotate the data. By effectively embracing and extending their data suppliers, Amazon became the primary source for bibliographic data on books, a reference source for scholars, librarians, and consumers.

[1] Delicious is a social sharing platform for bookmarks. For further information see: `http://delicious.com/`.

End of the Software Release Cycle As described above, the software of the internet era is delivered as a service and not as a product. Two points are important for this kind of software: First operations must be a core competency and the software has to be maintained on a daily basis, otherwise it will cease to perform. For instance, Google has to crawl the Web in order to update its indices, filter out link spam and other attempts to influence its results continuously. It further has to dynamically respond to hundreds of millions of user queries that must be matched with context-appropriate advertisements. Second it is important to treat users as co-developers, i. e., monitoring of user behavior in order to observe which new features are used and how they are used. On this basis it can be decided which features are kept and which are dismissed.

Lightweight Programming Models Amazon provides its Web services in two forms: The first adheres to the formalisms of the SOAP[2] Web service stack, the second provides XML data over HTTP in a lightweight approach called REST.[3] High value B2B connections (e. g., between Amazon and retail partner ToysRUs) usually make use of the SOAP stack, however Amazon reports a usage of 95% of the lightweight REST service.

Software Above the Level of Single Devices The Web 2.0 is no longer limited to the PC platform. A good example of this principle is iTunes.[4] With the PC acting as the local cache and control station the application reaches from the handheld device to a massive Web backend. Long time Microsoft developer Dave Stutz stated that "useful software written above the level of the single device will command high margins for a long time to come."

[2] SOAP is a protocol intended for exchanging structured information in a decentralized, distributed environment. For further information see: `http://www.w3.org/TR/soap/`.

[3] REST stands for Representational State Transfer and is an architectural style for distributed hypermedia systems. For further information see: `http://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf`.

[4] iTunes is an application which enables its users to buy music, movies, TV shows, and audiobooks, or download free podcasts from the iTunes Store. For further information see: `http://www.apple.com/itunes/`.

Rich User Experiences The potential of the Web to deliver full scale applications hit the mainstream when Google introduced Gmail[5] which is a free Webmail service with a rich user interface and PC-equivalent interactivity. In order to realize Gmail Google made use of a collection of technologies known as Ajax[6] which is a key component of the Web 2.0.

3.1.1 Collaborative Tagging Systems

With the arising of the Web 2.0 tagging systems have come up that allow their users to share various kinds of content. Such content can either be already available on the Web (e. g., Delicious and Diigo[7] for bookmarks) or it can be uploaded by the users (e. g., Flickr[8] for photos and YouTube[9] for videos). When contributing resources to such systems, users usually enter tags (i. e., freely chosen keywords) to describe and classify the content they provide thus improving the retrievability of resources for themselves and also for other users.

Collaborative Tagging

In Wikipedia a *tag* is defined as

> *"a non-hierarchical keyword or term assigned to a piece of information (such as an internet bookmark, digital image, or computer file). This kind of metadata helps describe an item and allows it to be found again by browsing or searching. Tags are chosen informally and personally by the item's creator or by its viewer, depending on the system."[Wikipedia, 2008]*

[Golder and Huberman, 2006] identify seven functions that tags perform for bookmarks:

- *Identifying what (or who) it is about:* The majority of tags identifies the topics of bookmarked items.

[5] `http://mail.google.com/`

[6] Ajax stands for Asynchronous JavaScript and XML and is a collection of several technologies. For further information see: `http://www.adaptivepath.com/publications/essays/archives/000385.php`.

[7] `http://www.diigo.com/`

[8] `http://www.flickr.com/`

[9] `http://www.youtube.com/`

- *Identifying what it is:* Tags may describe the kind of thing a bookmarked item is (e. g., an article, a blog, or a book).
- *Identifying who owns it:* Some tags describe who owns or created the bookmarked content. With regard to the apparent popularity of blogs knowing the content ownership can be particularly important.
- *Refining categories:* Some tags do not establish categories themselves. They rather refine or qualify existing categories.
- *Identifying qualities or characteristics:* Users express their opinions about the tagged content by assigning adjectives as tags such as scary, funny, or inspirational.
- *Self reference:* Some tags identify content in terms of its relation to the tagger. Such tags usually begin with "my" like, e. g., mystuff.
- *Task organizing:* In order to group information together that is related to performing a task, the information may be tagged according to the task (e. g., toread or jobsearch).

[Marlow et al., 2006] distinguish two high level categories that motivate people to annotate tags: *Organizational practices* rise from the use of tagging as an alternative to structured filing. *Social practices* consider the communicative nature of tagging, i. e., users express themselves, their opinions, etc. through the tags they use. These categories are then further refined as follows: *Future retrieval* of individual resources or collections of resources, *contribution and sharing*, *attract attention* to the own resources, *play and competition*, e.g., in the ESP game,[10] *self presentation* to write a user's identity into the system (e.g., the "seen live" tag in Last.fm) and *opinion expression.*

Folksonomies

The term *Folksonomy* was coined on July 24, 2004 by Thomas Vander Wal [Wal, 2007]. According to Vander Wal a

[10] In the ESP game two users see the same image and are asked to type in a tag for it. When they agree on a tag, they move on and are awarded with points. The ESP game is available at `http://www.gwap.com/gwap/`.

> *"Folksonomy is the result of personal free tagging of information and objects (anything with a URL) for one's own retrieval. The tagging is done in a social environment (usually shared and open to others). Folksonomy is created from the act of tagging by the person consuming the information."*

Following [Hotho et al., 2006b] a folksonomy can formally be defined as follows:

Definition 1 (Folksonomy) *A folksonomy is a tuple* $\mathbb{F} := (U, T, R, Y)$ *where*

- U*,* T*, and* R *are finite sets (users, tags, and resources)*
- Y *is a ternary relation between them (i. e., the tag assignments),* $Y \subseteq U \times T \times R$

Equivalently the folksonomy can also be seen as a tripartite (undirected) hypergraph with $G = (V, E)$, where $V = U \dot{\cup} T \dot{\cup} R$ is the set of nodes and $E = \{\{u, t, r\} \,|(u, t, r) \in Y\}$ is the set of hyperedges.

Taxonomy of Design Options for Tagging Systems

[Marlow et al., 2006] present a taxonomy of tagging systems. They describe dimensions of tagging systems and how the location of a system on the respective dimension may impact the behavior of the system. It should also be noted that some of the dimensions interact, i. e., a decision along one of them determines (or at least can be correlated with) the system's placement in another. This taxonomy will be depicted subsequently:

Tagging Rights Three types of tagging rights can be distinguished: With *self-tagging* users only tag their own resources. In the *permission-based* approach resource contributors may specify who is allowed to tag their resources (e. g., friends, family, or contacts). The photo sharing platform Flickr is an example for such a permission-based system. In tagging systems that allow *free-for-all* tagging, every user is allowed to tag any resource. Analogously the rights to delete a resource may be determined (no one, anyone, the tag creator, or the resource contributor/owner). The tagging rights influence the nature and type of the resultant tags as well as the role of the tags in the system. For instance the tags emerging in a free-for-all system are normally broad both in the number of tags assigned and in the nature of the tags.

Tagging Support Three categories to configure the process of adding tags to a resource can be observed: With *blind tagging* a user can't see the tags other users assigned to the same resource during the tagging process. *Viewable tagging* allows the user to see the tags which are already associated with a resource. With *suggestive tagging* the system proposes possible tags which a user may take over to annotate a resource.

[Sen et al., 2006] analyzed the evolution of the vocabulary in tagging communities based on community influence and personal tendency. In their experimental setup they had four groups. To the "unshared group" no community tags were shown. The "shared group" could see the tags applied by other members of their group. To the "shared-pop group" the most frequent tags for a resource were shown and the "shared-rec group" was presented with recommended tags that were annotated often for a resource or similar resources. The experiments indicate that viewing the community tags has an indirect impact on the user's tag applications by changing the personal tendency.

It is assumed that suggestive tagging may lead to a quicker convergence of the folksonomy, i. e., it supports the consolidation of the tag usage for a resource. However it is still unclear whether this is a good thing. For instance, when a system suggests tags that have already been annotated for a resource, early tag assignments might strongly influence the evolution of tags for a resource. Users might be detained to come up with their own ideas on how to tag a resource thus aggravating the process of harnessing collective intelligence.

In [Memmel et al., 2008] a prototypical implementation of a tag recommender for the ALOE system was introduced. As main sources for the generation of tag recommendations information about the user (her tags, profile, the tags of her contacts, etc.), the system (e. g., existing tags in the system) and the resources (existing resource tags, content, etc.) were identified. A first evaluation showed that the provided recommendations were perceived as helpful by the users of the system.

Aggregation Model A tagging system may support one of two aggregation models: First there is the *bag-model* which allows duplicate tags for the same resource from different users. Such a model is implemented in the Delicious system. The collective opinions of the taggers can be displayed in aggregated statistics for each resource. The tag data gathered that way may serve to more accurately find relationships between users, tags, and resources. Second there is the *set-model* where the users are asked to collectively tag an individual resource. Repetition of tags by different users is not allowed in this model. A set-model approach is implemented in the platforms Flickr and YouTube.

Object type There are tagging systems for a multitude of resource types. Any object that can be virtually represented can also be tagged. Besides the well known examples for photos, videos, bookmarks, and songs there are systems that allow the tagging of bibliographic material (e. g., CiteULike[11] or BibSonomy[12]), blog posts (Technorati[13]), architectural content (MACE[14]), and so forth. It is assumed that the type of object has implications for the nature and type of the tags being used, however, to our best knowledge this assumption has not been empirically tested, yet.

Source of Material Resources in a tagging system can stem from different sources. They can either be uploaded by the participants (e. g., Flickr, YouTube) or may be provided by the system (e. g., Last.fm). Some systems are also open for any resource which is available on the Web (e. g., Delicious, Digg[15]). The source of material may be restricted by the system architecture or through social norms (e. g., CiteULike).

Resource Connectivity Independent of the user tags, resources can be connected in different ways. For instance, Web pages are connected by direct links and many tagging systems allow the sharing of resources to groups. Such connections may have an impact on the convergence of tags for the affected resources, in particular in suggested or viewable scenarios.

Social Connectivity Just like resources also users of tagging systems may be connected. Links are either directed (i. e., a connection between users is not necessarily symmetric) or undirected and some systems even allow typed links (e. g.,

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

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

[13] `http://technorati.com/`

[14] `http://www.mace-project.eu/`

[15] `http://digg.com/`

contacts/friends in Flickr). The connection of users may possibly lead to an adoption of localized folksonomies that is based on the social structure in the system.

Subsequently we will present how technologies that are successful on the Web 2.0 may be deployed in enterprise scenarios to support harnessing collective intelligence and knowledge sharing between employees.

3.1.2 The Enterprise 2.0

McAffee spotted the potential of Web 2.0 technologies to foster the knowledge transfer in companies. He introduced the concept of the *Enterprise 2.0* as a collection of Web 2.0 technologies for generating, sharing, and refining information [McAfee, 2006]. Companies can buy or build these technologies in order to uncover the practices and outputs of their knowledge workers. His proposed SLATES framework consists of the following six components:

- **Search:** McAfee cites a Forrester study [Morris et al., 2005] which revealed that less than 50% of the intranet users reported to find the content they were looking for. Searches on the internet however are more likely to lead to successful search experiences (87%). This indicates that besides good intranet page layouts and navigation aids, there is a demand for improved keyword search on many platforms.

- **Links:** Google showed that the exploitation of the link structure between Web pages can significantly improve search results ranking. Intranets could also profit from this approach however it requires that many people can add links, not only the small group of people that develop the portal.

- **Authoring:** The example of Wikipedia has shown that group authorship can have convergent, high-quality content as output. In enterprises blogs and wikis should enable every staff member to share knowledge, insights, experiences, and the like.

- **Tags:** Besides improved keyword search, the study found that staff members would appreciate an improved categorization of content. Web 2.0 resource sharing platforms usually collect a large amount of resources and outsource the process of categorization (tagging) to their users. In enterprise platforms this could reveal patterns and processes in knowledge work by means of social navigation (see which tags the colleagues used, which pages they visited, and so on).

- **Extensions:** Often tagging is extended by automating categorization and pattern matching. Recommender systems serve as a well-known example. Based on the preferences a user expressed in the past, they recommend resources with similar content, resources that are preferred by the user's peers and the like.

- **Signals:** Checking the intranet for new content of a certain topic regularly is a tedious task. Feed technologies such as RSS and Atom can be used to inform the users of new content matching their topics of interest automatically.

So far we described the constituent parts of the Web of people and an application domain of Web 2.0 technologies in the enterprise. This is particularly relevant for our first use case targeting topic-based resource recommendations that was developed for an Enterprise 2.0 resource sharing platform. The algorithm uses social metadata such as tags and titles to obtain resource descriptions which are used to extract a user's topics of interest as well as for the retrieval of resources matching particular interest topics. We will now go on to describe the principles of the Web of data.

3.2 The Semantic Web

The Semantic Web is an initiative of the World Wide Web Consortium (W3C). It aims at presenting information on the Web in a way so that it can be processed by computers in a meaningful way. The main concepts of the Semantic Web will be described according to [Berners-Lee et al., 2001] subsequently.

Expressing Meaning Nowadays, the content of the Web is designed to be read by humans, not to be processed by computer programs meaningfully. The Semantic Web aims to bring a meaningful structure to the content of Web pages, that way creating an environment in which software agents that roam from page to page can carry out sophisticated tasks for users. It is an extension of the current Web, giving well-defined meaning to information in order to improve the cooperation between people and computers.

Knowledge Representation To make the Semantic Web function, it is necessary that computers have access to structured collections of information as well as sets of inference rules that can be exploited for automated reasoning. A fundamental technology for developing the Semantic Web is the Resource Description Framework (RDF).[16] RDF encodes meaning in sets of triples (like subject, verb, and object). The triples can be written using XML tags. With RDF, a document asserts that certain things, such as people, Web pages, etc., have properties (e. g., "is a sister of", "is the author of") with specific values (e. g., another person). Subjects, predicates, and objects are identified by Universal Resource Identifiers (URIs). The use of URIs allows everybody to define a new concept or verb by simply defining a URI for it on the Web.

Ontologies In [Wikipedia, 2009] the term ontology is defined as follows:

> *"Ontology [...] is the philosophical study of the nature of being, existence or reality in general, as well as of the basic categories of being and their relations."*

In Artificial Intelligence an ontology is usually understood as an explicit specification of a conceptualization in which shared knowledge is represented [Gruber, 1993]. Typically it consists of a taxonomy and a set of inference rules. A taxonomy is a definition of classes of objects together with the relations among them, e. g., an address may be modeled as a type of location, city codes may be modeled to apply only to locations, etc. Inference rules help to deduce further information from the ontology. For instance, a city code is associated with a state code and an address uses that city code. From this it follows that the address has the associated state code. Ontologies can be used, e. g., to improve the accuracy of Web searches. Instead of searching for ambiguous keywords, search programs can look for pages referring to a precise concept. Ontologies providing equivalence relations can help to resolve the problem of different URIs referring to a common concept.

Agents The potential of the Semantic Web will surface when programs are created that collect Web content from different sources, process the information, and exchange it with other programs. Some example applications will be pointed out subsequently:

[16] `http://www.w3.org/RDF/`

- *Proofs*, e. g., verification that a person is the one you were looking for.
- *Digital signatures* which can be used to verify automatically that some information has been provided by a specific trusted source.
- *Service discovery*, requires a description of the service that lets agents understand the function provided by the service as well as the way how to use it.

Evolution of Knowledge Besides developing tools for specific tasks, the Semantic Web can assist the evolution of human knowledge. The Semantic Web enables everybody to express new concepts by just naming them with a URI. These concepts can then be progressively linked into a universal Web, that way opening up the knowledge of humans for meaningful analysis by software agents. On this basis new tools can be developed which support people to live, work, and learn together.

To approach the vision of the Semantic Web the first step is to publish data that can be naturally understood by machines ([Berners-Lee and Fischetti, 2000], p. 177). Linked Data[17] is the means by which the Semantic Web can be realized [Bizer et al., 2009]. In [Wikipedia, 2010a] Linked Data is described as a method of exposing, sharing, and connecting data via URIs on the Web. [Berners-Lee, 2009] defines four Linked Data principles as follows:

1. Use URIs in order to name things.
2. HTTP URIs should be used thus enabling the look up of these names.
3. For agents that look up a URI, useful information should be provided according to standards such as RDF or SPARQL.[18]
4. Include links to other URIs, so that agents can discover more things.

[17] `http://linkeddata.org/`

[18] SPARQL is an RDF query language. For further information see: `http://www.w3.org/TR/rdf-sparql-query/`.

In January 2007 the Linking Open Data project was founded with the goal to bootstrap the Web of data. Supported by the W3C Semantic Web Education and Outreach Group[19] the initiative identifies existing data sets under open licenses, converts them to RDF according to the afore mentioned Linked Data principles and publishes them on the Web. Figure 3.2 presents a diagram of the Linking Open Data cloud of published data sets as well as the interlinkage between them. The content of the cloud is diverse comprising domains such as geographic locations, people, companies, books, scientific publications, movies, drugs and clinical trials, and many more [Bizer et al., 2009].

We use Linked Data for our second use case aiming at mood-based music recommendations. Here metadata from the Linking Open Data cloud is used to describe the artists a user listens to in terms of genres, instruments, etc. That way we identify groups of artists that represent a user's different preferred styles of music. In the last part of this chapter we will describe the convergence of the Web 2.0 and the Semantic Web into the Social Semantic Web.

3.3 The Social Semantic Web

[Weller, 2010] points at a trend that has been observed by several researchers in recent years, namely the influence of the Web 2.0 on the Semantic Web (e. g., [Ankolekar et al., 2007, Wahlster et al., 2006]). The convergence of the Social and the Semantic Web is often referred to as the Social Semantic Web or the Web 3.0. In her argumentation Weller cites the editorial by [Greaves and Mika, 2008] where the authors state that both the Web 2.0 and the Semantic Web have the common concept of "socially shared meaning". It is assumed that the Social Semantic Web will be an application area for Semantic Web technologies with the Social Web on the one hand, where value is created by aggregating the contributions of many individual users and the Semantic Web on the other hand integrating structured data from many different sources. Example applications of the Social Semantic Web comprise:

Semantically Interlinked Communities Here we want to briefly present two Semantic Web initiatives that aim at capturing social networks (e. g., people,

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

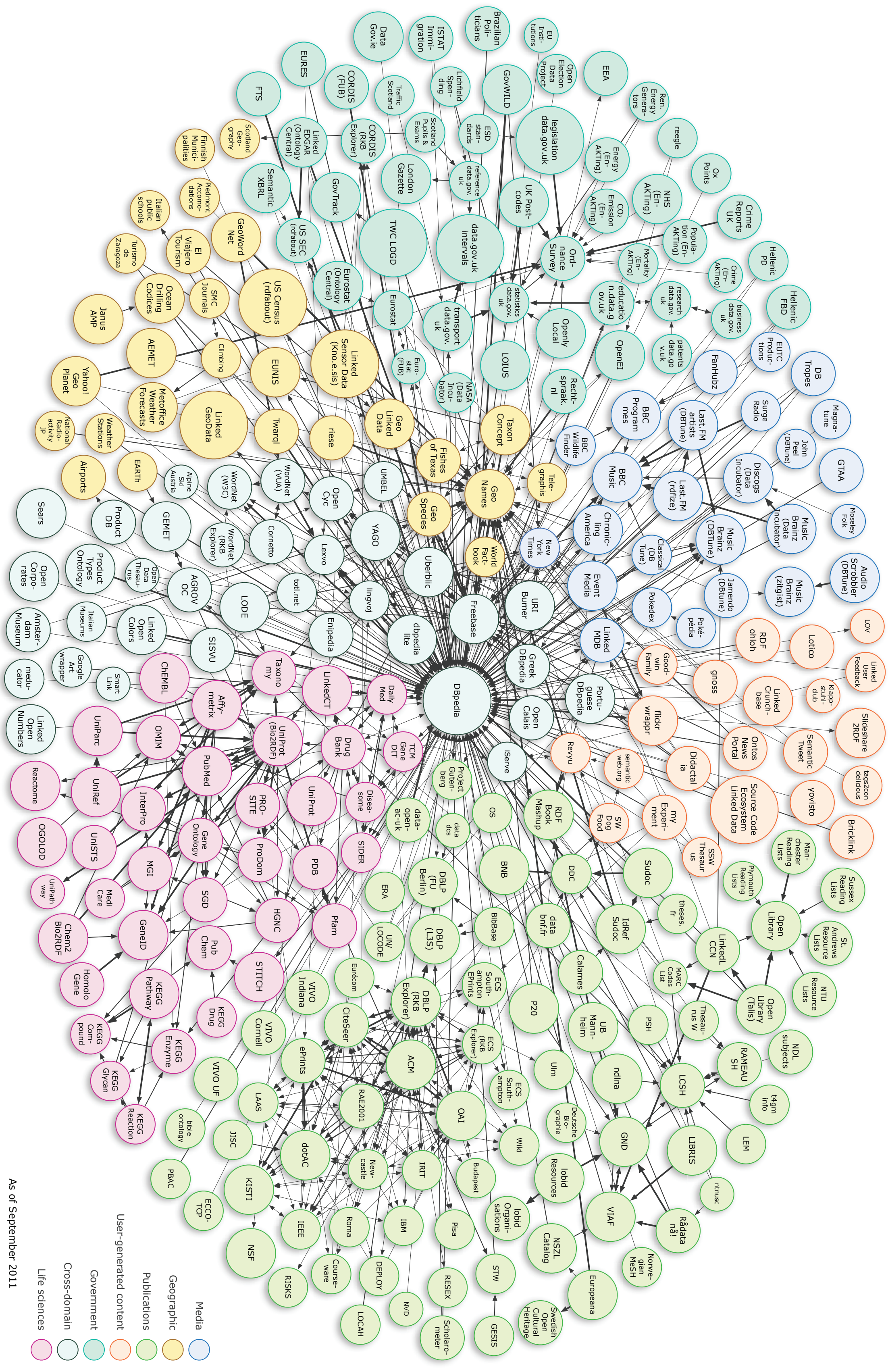


Figure 3.2: Linking Open Data cloud diagram ([Cyganiak and Jentzsch, 2011]).

projects, and events). Friend of a Friend (FOAF[20]) is the most popular such project. It was created in 2000 by Dan Brickley and Libby Miller as an *experimental linked information project*. Its goal is described on the website as "creating a Web of machine-readable pages describing people, the links between them and the things they create and do." Further the SIOC[21] (Semantically-Interlinked Online Communities) project is intended to integrate online community information. It offers an ontology thus enabling the representation of data from the Social Web in RDF format. SIOC is often used together with the FOAF vocabulary and has become a standard way to express user-generated content from Web 2.0 platforms.

Semantic Wikis A *wiki* is a platform allowing the easy creation and editing of interlinked Web pages via a browser either by using a simplified markup language or by making use of a WYSIWYG editor [Wikipedia, 2010e]. Application domains comprise community websites, personal note taking, and knowledge management systems in enterprises. The first wiki software was developed by Ward Cunningham. Semantic wikis such as the Kaukolu semantic wiki component [van Elst et al., 2008] and the Semantic MediaWiki [Krötzsch et al., 2006] add semantic annotations to wiki pages (such as categories, relations, and attributes in the Semantic MediaWiki) thus enabling

- *consistency of content* as information has to be stored only once and can be loaded to different pages,
- *accessing knowledge* in a structured way, e. g., "find all female physicists", and
- *reusing knowledge* in other tools, e. g., media players.

Linking Open Data Also the Linking Open Data project as described in Section 3.2 can be seen as an application domain of the Social Semantic Web representing (social) data and metadata in a structured way by making use of techniques such as RDF (e. g., in the DBpedia[22] project structured information is extracted from Wikipedia and made available on the Web).

[20] http://www.foaf-project.org/

[21] http://sioc-project.org/

[22] http://dbpedia.org/

Chapter 4

Background - The ALOE System

ALOE is an Enterprise 2.0 resource sharing platform designed for content of arbitrary format ([Memmel and Schirru, 2007]). It enables knowledge workers to share files and bookmarks according to their topics of interest hence making it an appropriate use case for our topic-based resource recommendation algorithm. The system has been deployed at the Knowledge Management department of the German Research Center for Artificial Intelligence.

This chapter starts with an overview of the basic functionalities of the ALOE system in Section 4.1. In Section 4.2 the system is classified according to the design options taxonomy for tagging systems proposed by [Marlow et al., 2006]. Then we depict in Section 4.3 the metadata that is annotated for resources in ALOE and how the system harnesses collective intelligence to obtain appropriate resource descriptions. Finally, Section 4.4 describes the conference organization system C-LINK that was built on top of the ALOE platform. In C-LINK we implemented the first recommender system for ALOE.

4.1 Basic Functionalities

The ALOE system supports sharing of bookmarks and all kinds of files (images, audio, video, office documents, etc.). It provides tagging, commenting, and rating functionalities. Search facilities are offered that provide ranking options taking into account the usage of resources (such as most viewed, highest rated, most commented).

In order to meet the users' privacy needs three access levels have been implemented. When contributing resources to the ALOE system, users have to choose one of three visibility options: Resources contributed with visibility *public* are visible for every user of the system. Resources with *closed group* visibility (see next paragraph) can only be retrieved and accessed by members of the respective group. Resources that are contributed with visibility *private* are only accessible by the contributor herself.

We realized a group concept that enables users to contact and exchange resources with other users that share similar topics of interest. Currently there are two different types of groups: *open* groups can be joined by any registered user of the system. Resources posted to such groups as well as the members of the groups are publicly visible. The membership in *closed* groups requires an authorization by the administrator of the respective group. Resources posted to such groups as well as group members are only visible for users that are themselves a member in the group.

While groups allow for a collaborative association of thematically related resources the concept of collections has been implemented to provide a further possibility for the personal organization of resources. Users can create collections according to different topics and share resources to them. Collections also allow for social browsing as users may navigate through the collections of other users. Further ALOE provides facilities to manage contact lists of users and it has a messaging functionality to easily enable users to contact each other.

4.2 System Design Options

Subsequently we will classify the ALOE system according to the taxonomy for tagging systems by [Marlow et al., 2006] as introduced in Chapter 3.1.1.

Tagging Rights ALOE follows the *self-tagging* approach in which users may only tag their own resources. However users can add resources that are already registered in the system to their portfolio. That way multiple users may “own” the same resource and add metadata to it thus enabling an extensive description of each resource.

Tagging Support In ALOE a combination of *viewable tagging* and *suggestive tagging* has been implemented. Users annotate tags when contributing a resource to the system. They may later also add tags on the detail view page of the resource (see Figure 4.1). Here the tags that have already been annotated for the resource are visible. Further the process of adding tags is supported by providing tag recommendations ([Memmel et al., 2008]).

Aggregation Model A *bag-model* for tag assignments has been implemented in the ALOE systems, i. e., one tag may be assigned by different users for the same resource. For each resource a tag cloud is displayed that shows the aggregated tag assignments.

Object Type ALOE is open for potentially all kinds of objects. The users may contribute bookmarks to the system or upload file resources.

Source of Material In ALOE the source of material is either the Web or the company's intranet (in the case when users contribute bookmarks to the system) or it is the participants (in the case when users upload files).

Resource Connectivity Besides connecting resources by tags, they can also be connected by sharing them to a common group or collection. The resources of a group or collection are presented in a list that can be ranked according to different criteria such as contribution date, average rating, or the resource titles.

Social Connectivity Users of the ALOE system are connected by

- *Contact lists:* A user can add another user to her contacts. The connection is directed and not necessarily symmetric.
- *Groups:* Users may also share group memberships. Every member of a group can easily see which other users are members of the group and contact those members with messages either individually or all members of the group at once.

4.3 Resource Metadata

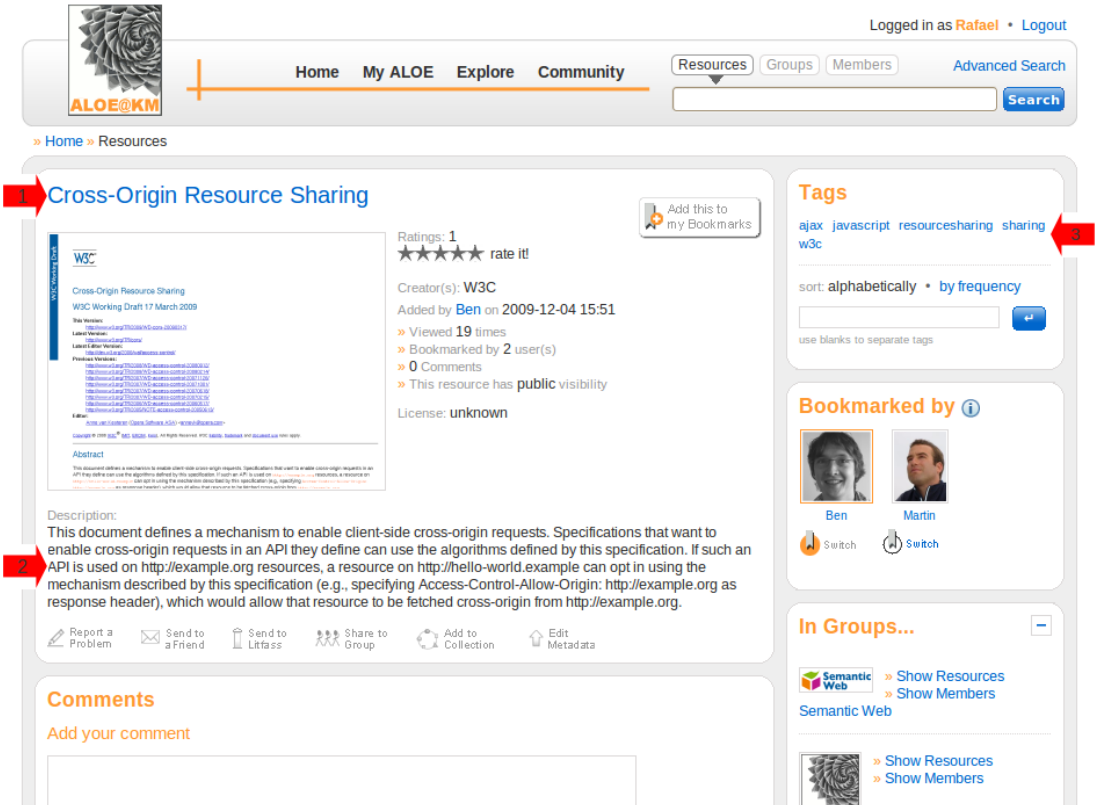
In order to make resources easily retrievable they are annotated with metadata provided by the community of users and with automatically generated metadata. In

ALOE we normalize URLs of bookmarks and represent each URL only once per visibility in the system, i. e., once public, once for each closed group contribution, and once for every private contribution However, each user can contribute a new instance of such a bookmark by adding it to her portfolio. During the contribution process she annotates a title and tags. Optionally a description, author, and licensing information may also be added. Further, we have system-generated metadata for every unique resource (an identifier, its format, and visibility) and system-generated metadata for every contribution (the identifier of the contributor and the contribution date). For file resources duplicates are currently not identified, i. e., each file is treated as a unique resource. Table 4.1 provides an overview of the ALOE resource metadata grouped according to the different metadata types.

The ALOE system supports resource contributions for a variety of content types. Users can either contribute bookmarks to the system or upload files such as office documents, images, audio, and video. Even though automatic approaches to analyze the content of images and videos are currently investigated (e. g., [Ulges et al., 2009] for videos and [Duan et al., 2009] for images), it is still difficult to extract a textual representation of these resources today. Exploiting tags from different users to collaboratively classify a resource has been described in Section 3.1.1 before. Concerning bibliographic metadata such as the title or the creator of a resource [Downes, 2003] states that this should be first party metadata, i. e., metadata provided by the resource author or a proxy (e. g., her company). However in ALOE such metadata is not inevitably available as the resource contributor is not necessarily the author of the resource. For that purpose we aim at harnessing the users' collective intelligence (see Section 3.1) by aggregating their metadata to obtain a proper description of the resources. In [Li et al., 2008] it was shown that social metadata is likely to describe the content of resources appropriately. So we decided to exploit the users' annotations to capture the content of the resources in the system. Figure 4.1 shows the ALOE details page for a resource. Selected metadata fields such as title (1), description (2), and tags (3) have been highlighted.

4.4 The C-LINK System

The idea to automatically extract a user's different topics of interest has risen from the experiences we made in the C-LINK project in 2008. The C-LINK system is a



Cross-Origin Resource Sharing

Ratings: 1
★★★★★ rate it!

Creator(s): W3C
Added by Ben on 2009-12-04 15:51
» Viewed 19 times
» Bookmarked by 2 user(s)
» 0 Comments
» This resource has public visibility

License: unknown

Description:
This document defines a mechanism to enable client-side cross-origin requests. Specifications that want to enable cross-origin requests in an API they define can use the algorithms defined by this specification. If such an API is used on http://example.org resources, a resource on http://hello-world.example can opt in using the mechanism described by this specification (e.g., specifying Access-Control-Allow-Origin: http://example.org as response header), which would allow that resource to be fetched cross-origin from http://example.org.

Comments

Add your comment

Tags

ajax javascript resourcesharing sharing w3c

Bookmarked by

Ben
Martin

In Groups...

Semantic Web
» Show Resources
» Show Members
Semantic Web

» Show Resources
» Show Members

Figure 4.1: Detail view of an ALOE resource. Selected user-generated metadata is highlighted ((1) title, (2) description, (3) tags).

System generated, once per resource	
id	ALOE identifier of the resource.
format	MIME type of the resource.
visibility	One of *public*, *group*, or *private.*
System generated, once per contribution	
contributor	Identifier of the user who contributed the resource to her portfolio.
contribution date	Point in time when the resource was added to the user's portfolio.
User-contributed metadata, once per contribution	
creator	Name of the person that created the resource.
description	A short description what the resource is about.
license	License under which the resource is available.
title	Title of the resource.

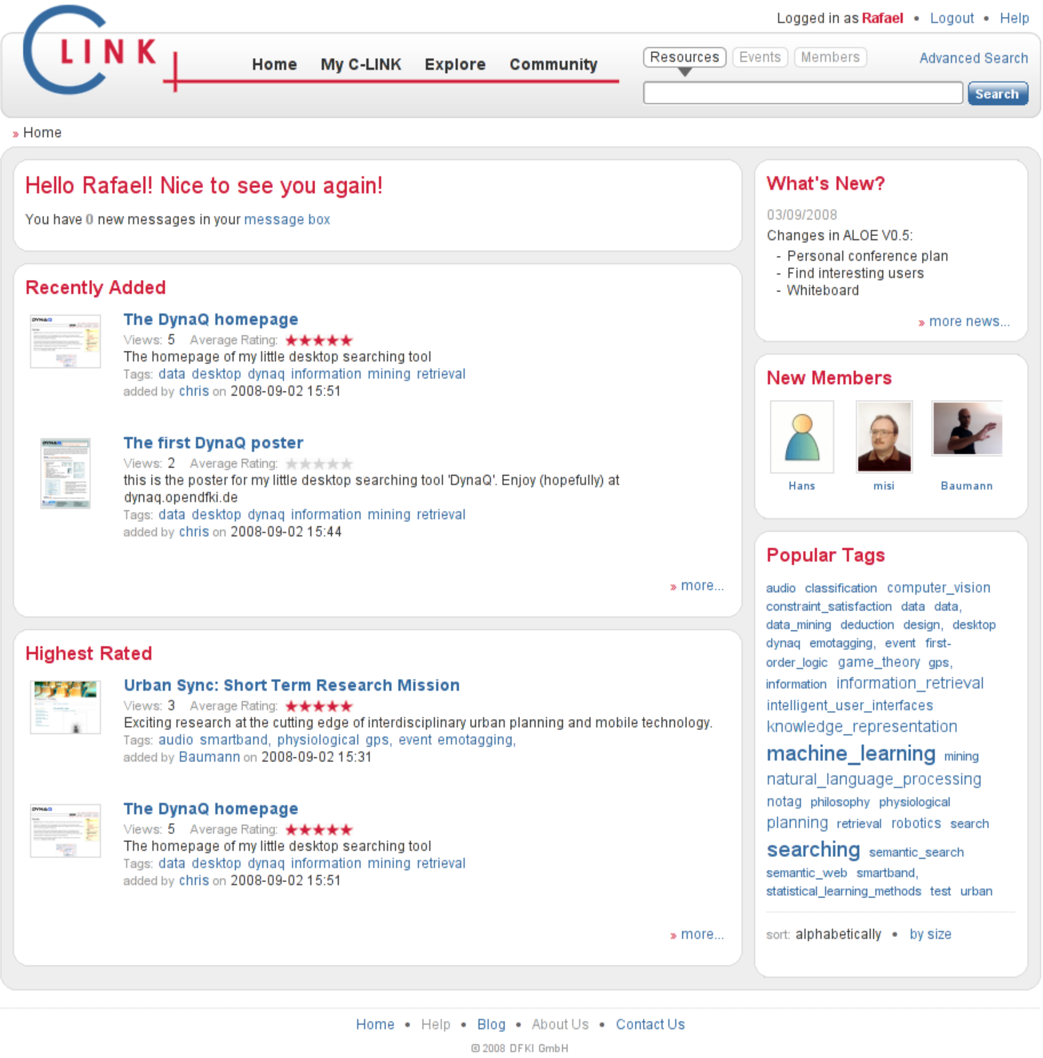
Table 4.1: ALOE resource metadata.

Web 2.0 conference organization system that has been built on top of the ALOE platform. It is a social sharing tool allowing conference participants to exchange, for instance, material related to their talks. C-LINK also provides social networking facilities such as finding users, e. g., according to their affiliation, exchanging messages, a chat room, and a whiteboard. A content-based recommender system has been integrated into the platform allowing for event recommendations as well as recommendations of potentially interesting users based on a user's research topics. Figure 4.2 shows the welcome page of the C-LINK system.

4.4.1 Recommendations in C-LINK

Content-based recommendations in C-LINK have been realized by integrating three different tools developed at DFKI:

- The *ALOE* platform is used as the underlying system for resource sharing and collection of social metadata.
- *DynaQ* [Agne et al., 2006] is a desktop search engine for document based



Hello Rafael! Nice to see you again!

You have 0 new messages in your message box

Recently Added

The DynaQ homepage
Views: 5 Average Rating: ★★★★★
The homepage of my little desktop searching tool
Tags: data desktop dynaq information mining retrieval
added by chris on 2008-09-02 15:51

The first DynaQ poster
Views: 2 Average Rating: ☆☆☆☆☆
this is the poster for my little desktop searching tool 'DynaQ'. Enjoy (hopefully) at dynaq.opendfki.de
Tags: data desktop dynaq information mining retrieval
added by chris on 2008-09-02 15:44

» more...

Highest Rated

Urban Sync: Short Term Research Mission
Views: 3 Average Rating: ★★★★★
Exciting research at the cutting edge of interdisciplinary urban planning and mobile technology.
Tags: audio smartband, physiological gps, event emotagging,
added by Baumann on 2008-09-02 15:31

The DynaQ homepage
Views: 5 Average Rating: ★★★★★
The homepage of my little desktop searching tool
Tags: data desktop dynaq information mining retrieval
added by chris on 2008-09-02 15:51

» more...

What's New?

03/09/2008
Changes in ALOE V0.5:
- Personal conference plan
- Find interesting users
- Whiteboard

» more news...

New Members

Hans misi Baumann

Popular Tags

audio classification computer_vision constraint_satisfaction data data, data_mining deduction design, desktop dynaq emotagging, event first-order_logic game_theory gps, information information_retrieval intelligent_user_interfaces knowledge_representation machine_learning mining natural_language_processing notag philosophy physiological planning retrieval robotics search searching semantic_search semantic_web smartband, statistical_learning_methods test urban

sort: alphabetically • by size

Figure 4.2: Welcome page of the C-LINK system.

Resources:	creator, description, title, full text
Events:	research topics (manually annotated), resource metadata profile of associated conference paper
User:	research topics (from user profile), annotated tags, resource metadata profiles of portfolio resources

Table 4.2: Composition of metadata profiles of resources, events, and users in the C-LINK system.

personal information spaces. It has a Lucene[1] backend thus enabling high-performance, full-featured text search. In C-LINK, DynaQ is used for matching metadata profiles of users and events.

- *MyCBR* [Stahl and Roth-Berghofer, 2008] is an integrated Case-Based Reasoning tool that extends the Protégé ontology editor.[2] In the C-LINK system, MyCBR is used to model the similarities between different research topics.

There are three different kinds of items in the C-LINK system that are relevant for recommendations: resources (i. e., user-contributed content), users, and events. For each of these items metadata profiles are composed which consist of user-contributed metadata, the full texts of the associated resources (where available) as well as manually annotated research topics. The detailed constitution of the metadata profiles is shown in Table 4.2.

Whenever a user requests event recommendations her current metadata profile is determined in the DynaQ backend. The user's research interests are extended by similar research topics as defined in MyCBR. The resulting query is matched against the profiles of the conference events. Finding similar users is performed analogously by extending the current user's metadata profile with related research topics and then matching it against the profiles of the other users in the system.

4.4.2 Review of the C-LINK Approach

Intuitively we found that using manually annotated interest topics leads to good recommendation results. However there are two drawbacks of such an approach:

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

[2] `http://protege.stanford.edu/`

First, a domain taxonomy of topics might not be available for every resource sharing platform. In our ALOE system, the users share resources according to their research interests, about software development, but also about topics in which they are interested privately. Setting up a domain taxonomy for such an open world scenario might not always be feasible. Second, it is widely recognized that the success of resource sharing platforms is among others based on their ease of use. Requiring the users to annotate resources with concepts from a taxonomy aggravates the contribution process and might hinder the usage of the system. For these reasons we aim at an approach that captures the interest topics of the users unobtrusively as a side effect of the normal usage of the system.

Part II

Approach

Chapter 5

Providing Topic-based Resource Recommendations

The approach described in this chapter aims at providing topic-based recommendations in Enterprise 2.0 resource sharing platforms taking into account a knowledge worker's different topics of interest. Usually in enterprise platforms the amount of resources is much larger than the amount of users thus aggravating the sparsity problems of traditional collaborative filtering systems (see Chapter 2.3). Collaborative filtering is in general not well suited to provide access to the long tail of resources. However it can provide high quality recommendations of popular items ([Celma, 2008]). In [Herlocker et al., 2004] it was argued that this can increase a user's trust in a recommender system which is particularly important for new users. According to our conviction only content-based recommender systems can provide access to the long tail of resources in scenarios where sparsity of the user-item rating matrix is an issue. Hence, we propose a switching hybrid recommender system that generates traditional item-based collaborative filtering recommendations for new users and provides content-based recommendations with a high degree of inter-topic diversity as soon as enough information about the user's preferences is available.

The remainder of this chapter is structured as follows: In Section 5.1 we present ways to elicit preferences for items and show how it is done in the ALOE system. Next, in Section 5.2 we introduce the concept of contextualized user interest profiles facilitating topic-based recommendations. The overall process for recommendation generation is depicted in detail in Section 5.3. Finally, we present related work in the fields of topic extraction and user interests identification in Section 5.4.

5.1 Eliciting Preferences for Items

Recommender systems need to learn about the users' preferences in order to provide them with useful recommendations. There are two possibilities to capture these preferences: First, users can be asked to rate items explicitly on a predefined scale. Second, user preferences can be inferred implicitly by observing the user's interaction with the system ([Nichols, 1997]). As explicit ratings impose a cognitive cost, often users are reluctant to vote. For that purpose many systems try to infer rating values implicitly by observing the users. Certain actions on a platform are considered as positive vote for a resource, e. g., adding a resource to ones portfolio, repeatedly visiting a resource, or printing a textual resource. The advantage of explicit ratings is that they are more accurate than the implicit votes. However recommender systems usually require a large amount of ratings from each user which is difficult to obtain with explicit ratings only. For that purpose hybrid approaches exist that exploit both kinds of available information.

The question which user actions on a resource can be interpreted as an expression of preference has to be answered for every application individually. For our recommender system based on the ALOE platform, we consider resource contributions, adding a resource to one's portfolio, and looking at the detailed metadata of a resource as implicit positive ratings. Explicit ratings are also considered by our recommender system and are taken over as provided by the users. All ratings are on a five point rating scale with five as the best and one as the worst rating that can be given. Further every action that is associated with a preference expression has a priority value assigned (one of low, middle, or high). In case two such actions from one user are associated with a resource, the rating value of the action with the higher priority is used. The rating and priority values of each preference relevant user action are depicted in Table 5.1.

5.2 Modeling Contextualized User Profiles

As stated by [Schwarz, 2006], the term *context* is used in different disciplines (e. g., linguistics and psychology) and understood in many different ways. Therefore when talking about context it is necessary to talk about its application as well as the scenario in which it is used. In our system, we assume that a knowledge worker has

Action	Rating Value	Priority Value
View detailed metadata	3	low
Add resource to portfolio	4	middle
Contribute Resource	4	middle
Rate resource	user's rating value	high

Table 5.1: Rating and priority values that are associated with user actions. The rating values are on a five point rating scale, where five is the best and one the worst value that can be given.

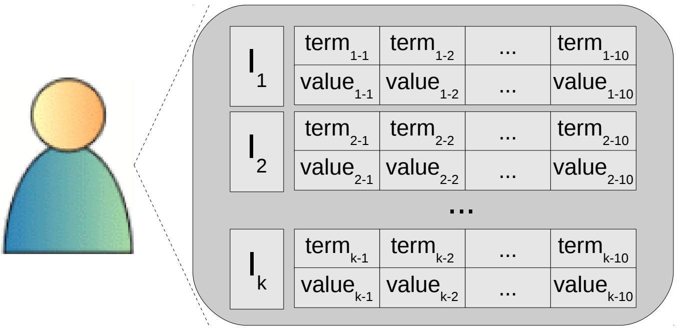

I_1
$term_{1-1}$ $term_{1-2}$ ... $term_{1-10}$
$value_{1-1}$ $value_{1-2}$ ... $value_{1-10}$
I_2
$term_{2-1}$ $term_{2-2}$ ... $term_{2-10}$
$value_{2-1}$ $value_{2-2}$ ... $value_{2-10}$
...
I_k
$term_{k-1}$ $term_{k-2}$ ... $term_{k-10}$
$value_{k-1}$ $value_{k-2}$ ... $value_{k-10}$

Figure 5.1: Schematic representation of a contextualized user interest profile.

different topics of interest. For instance, a software engineer might be interested in the Java programming language, the Linux operating system, and in punk rock music. When talking about the user's current context we refer to the interest topic that is currently relevant for her. Hence when modeling contextualized user interest profiles we require that these profiles are capable of representing the user's different interest topics in a way that allows for efficient retrieval of items for the respective topics.

Our approach applies textual data mining techniques on the metadata profiles of each user's preferred resources thus finding thematic groups that represent the users' interest topics. For every identified topic a weighted term vector consisting of at most ten terms is calculated. The weights are in accordance with the relevance of the associated term for the respective topic. We call the user-specific aggregation of these vectors a contextualized user profile. The expression multifaceted user profile is used synonymously in this thesis. A schematic representation of such a user profile is depicted in Figure 5.1.

Modeling user interest profiles that way allows us to generate resource recommendations with a high degree of diversity. For that purpose we can formulate data base queries where each query consists of the terms of one topic vector. When storing the metadata profiles of the resources, e. g., in a Lucene index, also the term weights can easily be exploited. The final recommendation list can be composed by selecting resources matching different interest topics of a user.

5.3 Providing Recommendations

The goal of our approach is the provision of resource recommendations according to a knowledge worker's different topics of interest. To achieve this target we need to gather a critical amount of information about the users first. Currently, we require that a user has explicitly or implicitly expressed preferences for at least 20 resources. However we want to generate useful recommendations also for users that are new to the system. For that purpose we propose a *switching hybrid recommender system* that generates traditional item-based collaborative filtering recommendations for users for which no interest topics have been identified yet. For users that have interacted with the system over a longer period of time and expressed preferences for a sufficient amount of items we determine the user's topics of interest and use them for content-based recommendations. The current section describes the calculations that are performed offline as well as the online recommendation generation process according to our approach firstly presented in [Schirru et al., 2011b].

5.3.1 Offline Analysis

In order to provide recommendations in real time for many users some time-consuming calculations are performed offline. The resulting data is stored in a data base or an index, and can be retrieved quickly when recommendations are requested. Subsequently we describe how similarities between items are calculated for the item-based collaborative filtering recommender. We then go on to depict the process of identifying a user's different topics of interest. Finally, we describe how item profiles are stored in a Lucene index thus enabling a fast lookup of on-topic resources when recommendations are requested.

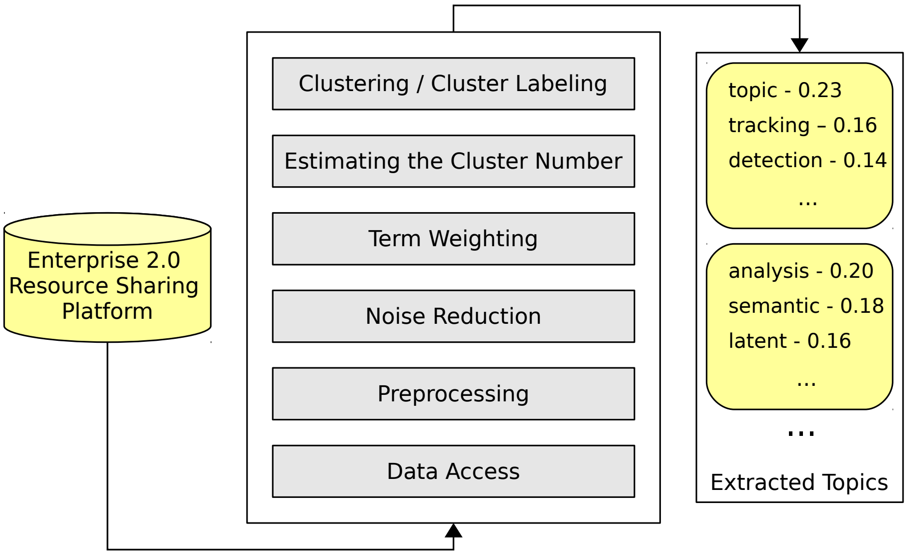


Figure 5.2: Topic extraction process steps.

Item Similarities

Once every day, we calculate the similarities between all public items in the ALOE data base, for which at least three common users have provided explicit or implicit ratings. The similarity between items is calculated as described in Chapter 2.2.2. To keep the recommendation generation process simple and efficient, we currently calculate the similarities between public items only. However, it would also be possible to include resources with group visibility. When providing recommendations it would then be necessary to restrict the result set to those items which are visible for the active user.

Identification of User Interests

The users' current topics of interest are at present determined once every week. The process is performed for those users that have expressed a preference for at least 20 items since the last time their interest topics have been identified. We apply textual data mining techniques on the profiles of a user's preferred resources. For these resources, metadata profiles are composed which are worked up and then fed to a clustering algorithm. The process steps of our topic extraction algorithm are depicted in Figure 5.2 and will be described in detail subsequently ([Schirru et al., 2010a]):

Data Access We determine all resources for which a user has expressed a preference since the last time her interest topics have been extracted. For each of these resources a metadata profile consisting of the annotated titles and tags is composed. Per resource potentially many titles are available as every user that adds the resource to her portfolio has to provide a title (see Chapter 4.3). Experiments have been conducted that also included the descriptions of the resources. However we observed that the best clustering results were achieved when only the titles and the tags of the resources were used.

Preprocessing We convert the terms contained in the metadata profiles to lower case characters, remove punctuation characters and stop words. Further stemming is applied to bring the terms to a normalized form. We use the Snowball stemmer[1] for this purpose. The normalized profiles of the resources are represented according to the "bag-of-words" model, i. e., they are represented as an unordered collection of words. These item profiles are then mapped to a vector space where each dimension corresponds to a term in the corpus (i. e., the set of the active user's preferred resources) and the dimension values are the counts of the words in the respective metadata profiles.

Noise Reduction Very rare and very frequent terms are not considered helpful to characterize resources. As a consequence dimensions representing these terms are removed. To reduce the noise that is inherent in social metadata we experimented with dimensionality reduction based on Latent Semantic Analysis ([Deerwester et al., 1990]). However, in our future work the positive impact of the application of this technique still has to be examined in greater depth.

Term Weighting Terms that appear frequently in the metadata profile of one resource but rarely in the whole corpus are likely to be good discriminators and should therefore obtain a higher weight. We use the TF-IDF measure ([Sparck Jones, 1972], see Chapter 2.1) which is widely applied in information retrieval systems in order to achieve this goal.

Clustering and Cluster Labeling To be able to cluster the set of a user's preferred resources we need to find a reasonable number of clusters in our data first.

[1] http://snowball.tartarus.org/

For this purpose we follow an approach which is based on the residual sum of squares (RSS) of a clustering result. The approach is depicted in detail in Appendix C.2.2.

For document clustering and cluster label extraction we apply non-negative matrix factorization (NMF, [Xu et al., 2003]). The output of the NMF algorithm is (i) a soft resource clustering and (ii) for each term and topic a weight indicating the relevance of the term for the respective topic. That way we can directly build the contextualized user profiles as described in Section 5.2 by composing a label for each topic consisting of its most relevant terms and aggregating the labels associated with a user's interest topics into her profile. The details of the NMF algorithm are described in Appendix C.3.2.

For text clustering several co-clustering techniques such as latent semantic analysis (LSA, cf. Appendix C.3.1), probabilistic latent semantic analysis (PLSA, [Hofmann, 1999]), non-negative matrix factorization, and latent dirichlet allocation (LDA) have been applied successfully in the past. We selected NMF for our approach as it is known to lead to good co-clustering results and its output is easy to interpret. Compared to LSA it has three advantages:

- LSA requires the axes of the semantic space to be orthogonal which makes the identification of latent semantic directions difficult for overlapping clusters.
- With NMF a document is an additive combination of the base latent semantics which is more reasonable in the text domain.
- The cluster membership of a document can directly be inferred from the NMF whereas with LSA further data clustering methods such as K-Means have to be applied in the eigenvector space.

We expect that PLSA and LDA lead to results that are comparable to NMF in the application domain. For our future work it would be interesting to experiment with these co-clustering algorithms.

Item Profiles Index

In order to enable a fast lookup of items matching a user's topics of interest we store the metadata profiles of all public resources of the ALOE system in a Lucene index. The profiles consist of the titles, descriptions, and tags that have been annotated for each resource. The index is updated once every day. As with the item similarities

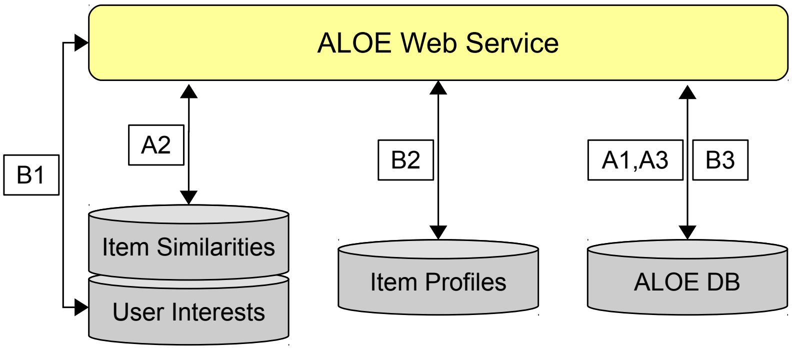


Figure 5.3: High level overview of the proposed recommender system.

it would also be possible to include resources with group visibility in the index. At the point in time when recommendations are generated, the resources that are not visible for the active user would then have to be filtered out.

5.3.2 Online Recommendation Generation

The components which are involved in the recommendation generation process are depicted in Figure 5.3. Whenever a user requests recommendations, two use cases have to be distinguished that will be described subsequently:

Recommendations without Interest Topics Available

The user has expressed a preference for at least one resource for which similar items could be determined, however no user interest topics are available yet. This may be accounted to the following reasons: First, it might be that the user has not expressed preferences for the minimum number of required items. Second, it might be that the user interests identification algorithm has not been run for the current user yet. In both cases recommendations are calculated according to the item-based collaborative filtering method:

- The list of the user's preferred items is loaded from the ALOE data base (A1).
- For each preferred item the similar items (see Section 5.3.1) are loaded (A2).
- From the list of similar items those items are removed that are already known by the user.

- For the remaining items a rating prediction is calculated using the item-based collaborative filtering algorithm as described in Section 2.2.2.
- Metadata of the top ten items is loaded from the ALOE data base and returned to the user (A3).

Recommendations with Interest Topics Available

In the case that a user's topics of interest have been identified, the recommendations are generated as follows:

- The user's interest topics are loaded from the data base (B1).
- From the latest interest topics we select randomly a predefined number. Currently, we select at most five interest topics. However if less topics are available, all of them will be used.
- The index containing the item profiles is queried according to these interest topics (B2). For each topic, we compose a query that contains the relevant topic terms. A discussion on how to choose the term relevance threshold is provided in Section 7.3.2. The term weights are used as boosting factors in our query and the terms are connected by an "OR" semantic.
- From the retrieved resources those items which are already known by the user are removed.
- A recommendation list is composed that should contain at most ten items and we aim at delivering an equal number of items for each selected user interest topic.
- Metadata of the recommended items is loaded from the ALOE data base and the list is returned to the user (B3).

When relevant resources for the selected interest topics can be found in the data base the proposed approach ensures a high degree of inter-topic diversity in the recommendation lists. In Chapter 7.3.2 we evaluate the specificity of our extracted cluster labels, i.e., we test how well the topic labels can separate a user's resources that belong to a topic from the rest of her resources. The specificity of the cluster labels is assessed by making use of measures from the field of information retrieval such as precision, recall, and the F-measure. Subsequently we will present related work in the field of topic extraction that has been applied in different scenarios comprising the detection of topics in accumulating document collections, search results, as well as social resource sharing platforms.

5.4 Background on Topic Extraction Algorithms

To detect the topics in the metadata profiles of the users' preferred resources our approach uses algorithms from the domain of topic detection and tracking (TDT). TDT is concerned with finding and following new events in a stream of documents. In [Allan et al., 1998] the following TDT tasks have been identified: First is the segmentation task, i. e., segmenting a continuous stream of text into its several stories. Second, there is the detection task which comprises the retrospective analysis of a corpus to identify the discussed events and the identification of new events based on online streams of stories. Third is the tracking task where incoming stories are associated with events known in the system. In this work we focus on the detection of topics in the profiles of the users' preferred resources.

[Schult and Spiliopoulou, 2006] consider the problem of finding emerging and persistent themes in accumulating document collections which are organized in rigid categorization schemes such as taxonomies. They propose Theme-Monitor, an algorithm for monitoring evolving themes from accumulating document collections. The algorithm works as follows: In the first period, it clusters all documents in the collection. In the following periods, it clusters the new documents with the old feature space and compares the new clusters to the ones found in the previous period. If the clusters of two adjacent periods are similar with regard to their themes and if the quality of the clustering is not declining significantly, then the original feature space is kept. Otherwise a new feature space is build for the documents of the latest period and the next comparison. Thematic clusters are represented by a label, consisting of a set of terms that have a minimal support in the associated cluster. Thematic clusters that survived over several periods, despite re-clustering and changes of the feature space, will become part of the classification scheme. Theme-Monitor identifies persistent topics in accumulating document collections. In contrast our approach does not consider a user's preferred resources as an accumulating set of items. Instead, each time the user's current topics of interest are determined only those items are considered for which the user has expressed a preference since the last time her interest topics have been identified. Also our method does not aim at keeping the feature space between different runs of the algorithm. Each time the interests identification algorithm is run we build a new feature space from the metadata profiles of the items of the current period that way targeting at topic labels that best describe a user's current topics of interest. In our future work we also intend to find persistent interest topics of users. For that purpose we will either compare the topic labels of different time periods or apply sophisticated methods such as online non-negative matrix factorization ([Cao et al., 2007]) which automatically track the evolution of topics while the data evolves.

[Osinski et al., 2004] present Lingo, an algorithm for search results clustering based on singular value decomposition (SVD). Lingo is implemented in Carrot[2], an open source search results clustering engine.[2] The goal of Lingo is to provide an overview of the topics that are covered in a search result by providing readable and unambiguous descriptions of the thematic groups that way facilitating access to the specific group of documents a user is looking for. The authors state that most open text clustering algorithms first perform document clustering and then derive descriptions (cluster labels) from these clusters. Lingo follows a reverse "description-comes-first" approach. It first ensures that human-understandable cluster labels can be created and then assigns documents to them. The algorithm works as follows: first the algorithm extracts frequent phrases from the input documents, hoping that they are the most informative sources of human-understandable topic descriptions. Next the document-term matrix which represents the search result is reduced by applying SVD, that way trying to discover the existing latent structure of diverse topics. In the last step group descriptions are matched with the extracted topics and the relevant documents are assigned to the groups. Similar to our approach Lingo uses a matrix factorization technique to find groups of topically related documents in a corpus. However, in the first place the algorithm aims at deriving human understandable cluster labels. For that purpose it generates cluster descriptions consisting of frequent phrases or expressive single terms. Contrastingly the main target of our approach is the identification of cluster labels that optimize the retrieval of resources matching the associated topic. Therefore we chose a label representation consisting of at most ten weighted terms facilitating precise weighted queries.

[2] http://project.carrot2.org/

[Li et al., 2008] propose an approach to discover social interests shared by groups of users. For that purpose they analyze patterns of frequent co-occurring tags to capture and characterize topics of user interests. They developed the ISID (Internet Social Interest Discovery) system to identify common user interests and to cluster users as well as their saved URLs by different topics of interest. The architecture of the system consists of three components. First, there is the *topic discovery* component which for a given set of bookmarks finds all topics of interest. Every topic of interest is a set of tags whose number of co-occurrences exceeds a given threshold. The patterns are identified by an association rule algorithm. Second, there is the *clustering* component. This component finds for each topic of interest, all the URLs and users that have labeled the URLs with all the tags in the topic. That way for every topic a user cluster and a URL cluster are generated. Third, the *indexing* component imports the topics of interest as well as the respective user and URL clusters into an indexing system which allows for application queries. ISID is applied in a scenario similar to ours namely in a Web 2.0 resource sharing platform. It also uses social metadata (tags) to capture the content of items and to derive interest topics of users. However, like the Fab system proposed by [Balabanovic and Shoham, 1997] (see Chapter 8.1) ISID extracts interest topics that are shared by many users. We aim at topic descriptions that are stronger tailored to the individual users in order to recommend resources that better match a user's particular interests in a vast information space in which potentially many resources are available for many interest topics.

Chapter 6

Mood-based Music Recommendations

As already stated before, traditional content-based and collaborative recommender systems are known to suffer from overspecialization thus not taking into account a user's full range of interests. In this chapter we tackle the problem of extracting multifaceted user profiles that facilitate context-sensitive music recommendations based on the user's current mood. We hypothesize that users clearly have a multitude of different preferred music styles that should be considered when music recommendations are provided. For that purpose we propose an approach to extract contextualized user profiles that allow for personalized recommendations of e. g., internet radio stations or can be integrated into existing content-based recommender systems. The analyses are performed on the artist level. Our method annotates the artists a user prefers by making use of metadata from Semantic Web data sources. These artists are clustered and for each group a label is extracted that describes the music style associated with the respective group. We compose the multifaceted profile of a user by aggregating the labels describing her preferred music styles. The method is a second use case for the approach presented in Chapter 5. The major difference is that the metadata annotated for a user's preferred artists is not provided by the user herself or other users of the platform where the method is applied. Instead LOD is used to represent music items. Further, some algorithms have been exchanged for experimental purposes.

The current chapter is structured as follows: In Section 6.1 we introduce two psychological models that are used to represent mood in music. Further two mood classification schemes based on social tags are presented of which the second relies on these psychological models. Next in Section 6.2 we depict the data sets used for our work and the method applied to extract a user's contextualized profile of preferred music styles. In Section 6.3 we propose different strategies on how to use these profiles for music recommendations. Finally, in Section 6.4 we summarize some of the sparse literature about recommendation approaches applying Semantic Web data sets and technologies.

6.1 Representations of Mood in Music

In the literature different models for the representation of mood (both music specific and general) have been proposed. Subsequently we will describe two different approaches: [Hevner, 1936] arranged 66 adjectives describing mood in music into eight neighboring groups. All adjectives in one group should be closely related and compatible with each other. Adjectives in two adjacent groups are intended to have at least some characteristics in common. The eight groups are arranged in a circle. Two groups at the extremities of any diameter should be as oppositional as possible. Hevner's mood groups are depicted in detail in Figure 6.1. Other approaches assume a highly systematic relation between affective states. [Russell, 1980] proposed the hypothesis that affective states are best represented as a circle in a two-dimensional, bipolar space with the first dimension representing pleasure and the second dimension representing arousal. Russell's assumption is based on evidence on how lay people conceptualize affective states and on multivariate analysis of self-reported states. His affect model is depicted in Figure 6.2.

When talking about mood-based music recommendations we need to first settle on the mood categories that have to be used and characterize each category. [Hu et al., 2007] proposed a classification consisting of three mood categories that have been developed to be used for the "Audio Music Mood Classification" task in the Music Information Retrieval Evaluation eXchange (MIREX). However using only three mood categories was considered by many as domain oversimplification. For this reason [Hu et al., 2009a] proposed a new classification schema consisting of 18 categories. They collected tags from Last.fm for an in-house collection of 21,000 audio tracks. For 12,066 songs at least one tag could be found. The tags that

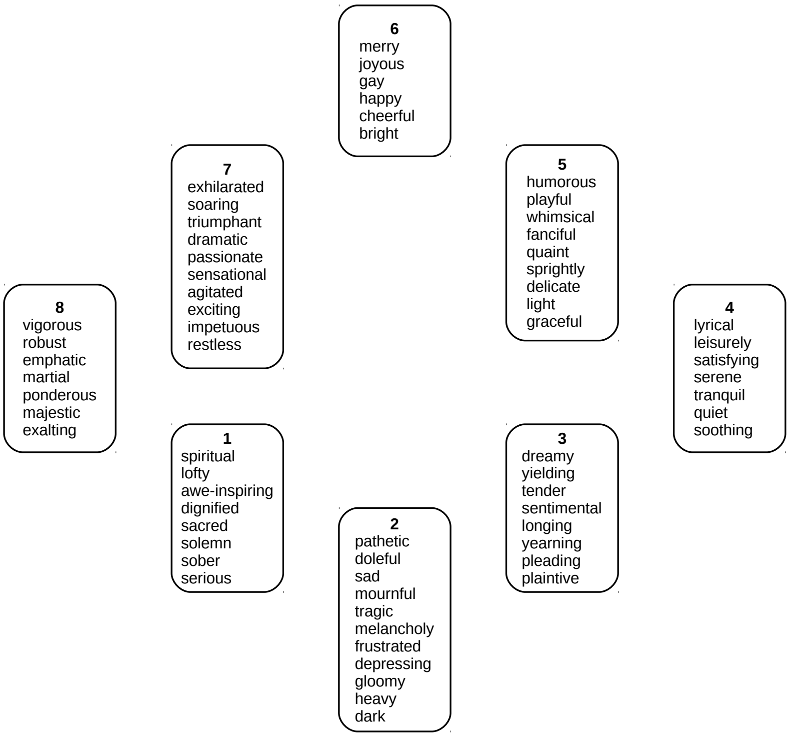


Figure 6.1: Mood adjectives arranged in eight related groups ([Hevner, 1936]).

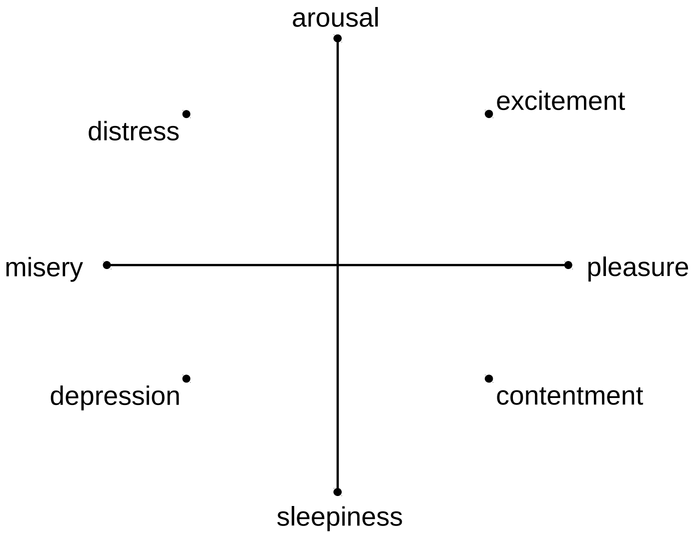


Figure 6.2: Eight affect concepts according to [Russell, 1980] in a circular order.

have been retrieved were filtered using the WordNet-Affect[1] lexicon. Such tags with no or little affective meaning were removed. The remaining 348 tags were further cleaned up by two human experts so that 186 tags remained in the data set. Using sets of synonyms defined in WordNet-Affect these tags were grouped into 49 categories. The experts further merged these groups leading to 34 mood categories. Those categories with less than 20 songs in them were dropped and finally 18 mood categories with 135 tags were used. A detailed overview of the mood categories and tags proposed by Hu et al. is provided in Appendix D.

[Laurier et al., 2009] presented a different representation of mood using social tags. The authors selected 120 words related to emotions from different sources (among others from psychological models, e. g., [Hevner, 1936] and [Russell, 1980] and from the MIREX task). Then they crawled 6,814,068 tag annotations from 575,149 songs from Last.fm where they found 107 of their original 120 terms. After that they discarded the terms appearing less then 100 times which resulted in a list of 80 words. Using 61,080 tracks where the same mood tags have been assigned by several users they map the songs and mood tags to a vector space where the columns are track vectors. As this space has a high dimensionality and is very sparse they apply Latent Semantic Analysis on the matrix reducing it to 100 dimensions. In the last step the authors apply clustering on the data obtained and find four clusters of mood tags. It was found that these clusters are in accordance with the arousal-valence plane from Russell as presented before. In Appendix D we depict the first 15 tags from each cluster. Unfortunately we were not able to obtain a list of all tags for each cluster from the authors.

6.2 Approach

The current section describes our algorithm to extract a user's contextualized profile of preferred music styles as presented in [Schirru et al., 2011a]. According to our research hypothesis H5 we expect each identified preferred music style to correlate with a particular user mood. We depict in Section 6.2.1 the data sets that have been examined according to their usefulness for our approach. In Section 6.2.2 we present the data mining stack applied for the user profile extraction.

[1] WordNet-Affect extends WordNet by assigning affect labels to concepts representing emotions, moods, and emotional responses.

6.2.1 Data Sources

We checked three Semantic Web data sources for metadata that are available to describe artists: Freebase, DBpedia, and MusicBrainz. Each of them will be described briefly subsequently.

Freebase

Freebase[2] is an online collection of structured data that has been harvested from many different sources. It also includes direct wiki-like contributions provided by the community of users that way forming a large collaborative knowledge base. The aim of Freebase is the creation of a global resource allowing people and machines to access common information in a convenient way. The Freebase data is available under the Creative Commons Attribution 2.5 Generic license.[3] It can be accessed via a dedicated API, an RDF endpoint, as well as data base dumps. Freebase is developed by the American software company Metaweb and has been publicly available since March 2007. The available metadata to describe artists comprise:

- *origin*: The place (city or country) where an artist or group started their career.
- *instrument*: The instrument(s) an artist plays. For these instruments also the associated *instrument families* can be requested from Freebase.
- *genre*: The musical genre of the artist or group.
- *artist collaboration*: Collection of artists and/or groups that worked together.
- *record release year*: Year in which an artist or group has released a record.

The full list of metadata describing items in the music domain can be retrieved at `http://wiki.freebase.com/wiki/Music_commons`.

DBpedia

The DBpedia[4] project is a community effort that aims at extracting structured information from Wikipedia and making the information available on the Web. That

[2] `http://www.freebase.com/`

[3] `http://creativecommons.org/licenses/by/2.5/`

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

way it covers many domains, represents community agreement, and it automatically evolves as Wikipedia changes. Access to the DBpedia data set is granted online via a SPARQL query endpoint and as Linked Data. Further the data can be downloaded as text files either in N-Triples or in N-Quads format. DBpedia is licensed under the Creative Commons Attribution-Share Alike 3.0 license[5] and the GNU Free Documentation License.[6]

Interesting metadata in DBpedia describing artists comprise the category labels. These are the labels that are shown on the bottom of most of the Wikipedia pages. For example, for Madonna we have categories “1958 births", “1980s singers", “1990s singers", “American female singers", etc. A complete overview of the metadata available in DBpedia can be retrieved at `http://mappings.dbpedia.org/server/ontology/classes/`.

MusicBrainz

MusicBrainz[7] is an open user-maintained community that collects music metadata and makes them available to the public. It was initiated by Robert Kaye as a response to Gracenote taking over CDDB and charging money for the access to the previously free data. MusicBrainz collects a large amount of data comprising metadata about artists, release groups (e. g., albums and singles), release dates, track data, and label data. They are placed partly into the public domain[8] and some parts are covered by a Creative Commons Attribution-NonCommercial-ShareAlike 2.0 license.[9] The data are made available as a dump for PostgreSQL data bases. To describe artists the data base comprises among others the following information:

- Name of the artist or group
- Common aliases and misspellings
- Type (one of person or group)
- Begin date (birth date or formation date, depending on the type)

[5] `http://creativecommons.org/licenses/by-sa/3.0/`

[6] `http://en.wikipedia.org/wiki/Wikipedia:Text_of_the_GNU_Free_Documentation_License/`

[7] `http://musicbrainz.org/`

[8] `http://creativecommons.org/licenses/publicdomain/`

[9] `http://creativecommons.org/licenses/by-nc-sa/2.0/`

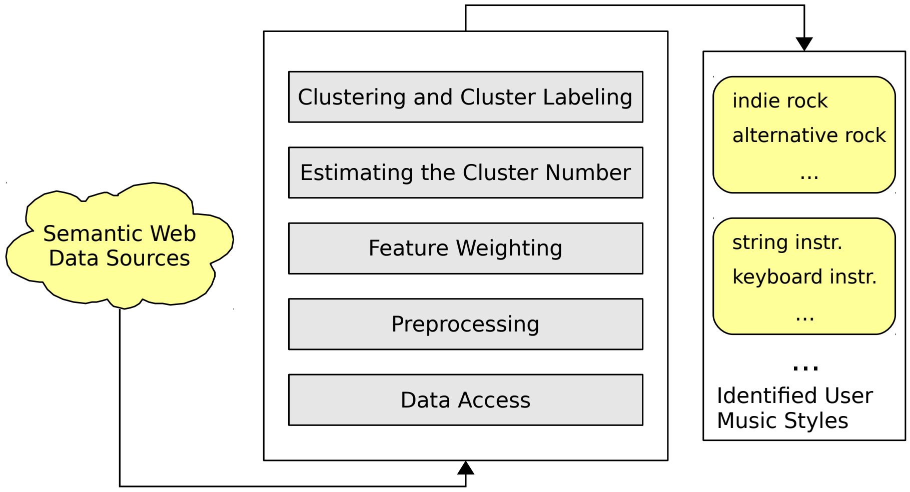


Figure 6.3: Music styles identification process steps.

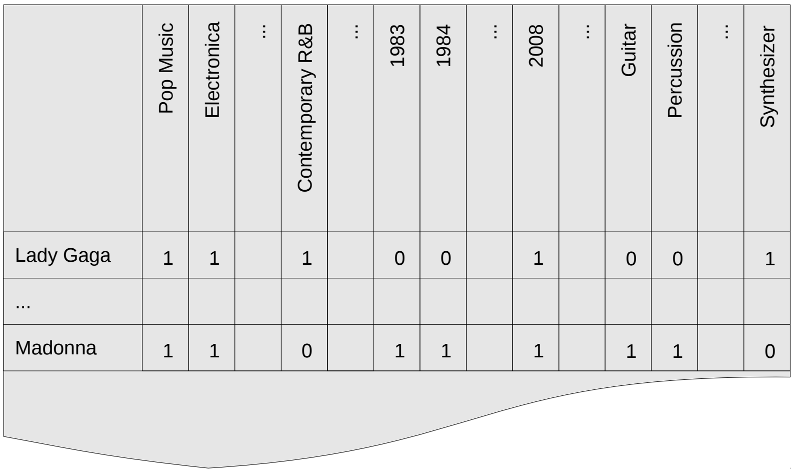
- End date (death date or dissolution date, depending on the type)

The complete list of metadata in the MusicBrainz data base can be retrieved at `http://musicbrainz.org/doc/MusicBrainz_Database`.

6.2.2 Algorithm

In order to determine the styles of music a user listens to we extract the played artists from her playlist (e. g., from Last.fm). For these artists metadata profiles are built describing the music styles an artist is associated with. Our approach groups artists serving similar music styles and extracts a label for each group. Figure 6.3 shows the single steps of our method. Each step will be described in detail subsequently.

Data Access For every user we extract the artists played from her playlist. We require a user to have at least 20 distinct artists in her playlist in order to enable the identification of different preferred music styles. For every artist we build a metadata profile consisting of features extracted from Freebase. Currently we use the genres as well as the instruments and instrument families an artist is associated with. These artist profiles are then mapped to a vector space. In the vector space we have a dimension for each available artist feature. Its value in the artist property vector is either set to one if the feature applies to an artist or to zero, otherwise. Figure 6.4 shows an example of such an artist property matrix.



	Pop Music	Electronica	...	Contemporary R&B	...	1983	1984	...	2008	...	Guitar	Percussion	...	Synthesizer
Lady Gaga	1	1		1		0	0		1		0	0		1
...														
Madonna	1	1		0		1	1		1		1	1		0

Figure 6.4: Exemplary representation of an artist-property matrix.

Preprocessing In order to cope with noise features and features with low information content we apply frequency-based feature selection. We remove very rare features appearing with less than 3% of a user's preferred artists and we remove very frequent features that are annotated for more than 60% of the artists as these features are not considered to be discriminative.

Feature Weighting Artist properties that appear rather rarely are more discriminative than such properties that appear very often. For that reason such features should obtain a higher weight. Analogously to our method extracting a user's preferred topics of interest (cf. Chapter 5.3.1) we apply the TF-IDF measure in order to weight the features of the artist vectors.

Estimating the Cluster Number To estimate a reasonable cluster number we apply once again our approach based on the residual sum of squares of clustering results with different cluster numbers. The method is described in detail in Appendix C.2.2.

Clustering and Cluster Label Extraction We performed K-Means clustering on our data using the RapidMiner[10] Java library. Cluster labeling has been implemented based on the chi-square method as described in [Manning et al., 2009], pp. 396. The details of K-Means clustering are depicted in detail in Appendix C.2.1.

[10] `http://rapid-i.com/`

We explain cluster labeling using the chi-square test in Appendix C.4.2. Only such features are included in the label of a cluster whose weight is at least $r\%$ of the weight of the most relevant feature of the currently considered cluster. In Section 7.4.2 we evaluate the best value for parameter r.

As a flat clustering algorithm K-Means has some drawbacks compared to hierarchical clustering algorithms, these comprise (cf. Appendix C.2):

- The results are nondeterministic.
- For hierarchical clustering algorithms there exist good heuristics to determine a reasonable number of clusters.
- The results miss a structure that can be visualized easily like the dendrograms that are obtained with hierarchical clustering.

However, a major advantage of flat clustering strategies is their efficiency. While the most common hierarchical algorithms have at least quadratic complexity in the number of items, the runtime of K-Means is linear in all relevant factors: the number of iterations, clusters, vectors and dimensionality of the space. As our proposed algorithm has to be applicable in an online platform with a large amount of users we chose to use this efficient algorithm for our purposes.

6.3 Providing Recommendations

In this section we want to sketch possibilities on how the contextualized user profiles representing a user's preferred music styles can be used for music recommendations:

Integration with Previous Approach In [Baumann et al., 2010] we proposed an approach to determine similar artists based on metadata from the Semantic Web. In our online demonstration system HORST[11] a user has to enter a preferred artist and the system identifies similar artists with a high degree of novelty. Figure 6.5 shows HORST's similar artists interface. The approach presented in this chapter integrates with HORST seamlessly. The identified clusters representing a user's preferred music styles can, e. g., be used to find other artists matching a specific music style but are presumably unknown to the active user. For that purpose it

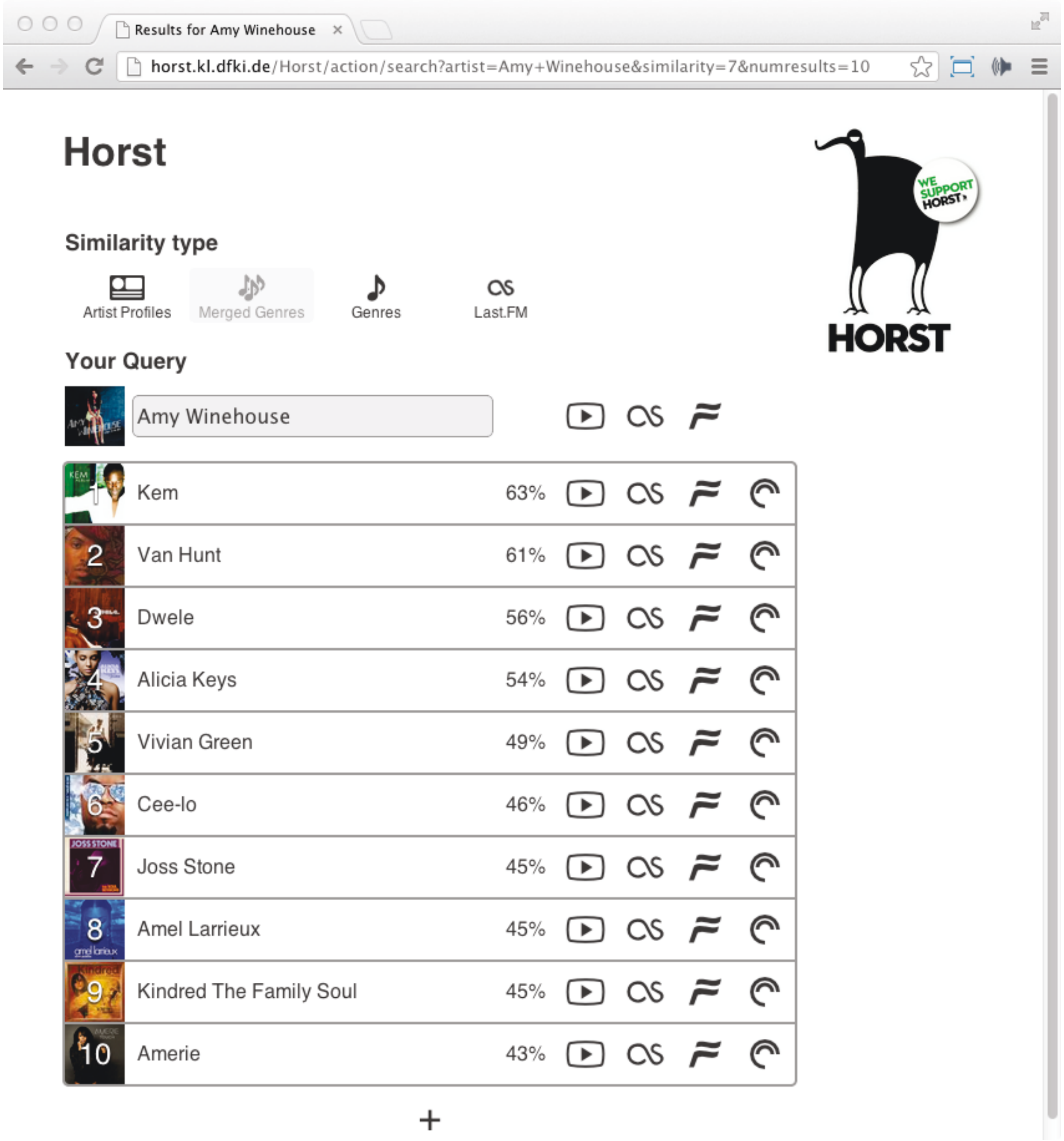
[11] `http://horst.kl.dfki.de/Horst/action/index`

would simply be necessary to pick a representative artist from the respective cluster and to use it as a query for HORST in order to find similar artists that are not in the active user's listening history.

Recommendation of Radio Stations The labels representing preferred music styles mostly consist of genre descriptions that apply to the artists in the respective clusters (cf. Table 7.5). They can be used for personalized recommendations of genre-based internet radio stations taking into account a user's full range of music preferences.

6.4 Background on Semantics-based Recommender Systems

A recommender system exploiting the global graph of Linked Data (see Chapter 3.2) is presented by [Passant, 2010]. Passant introduces distance measures on Linked Data to determine the relatedness between resources. These measures can be used to generate self-explanatory resource recommendations. For music recommendations Passant uses a distance measure that takes into account direct links and indirect links between resources (i. e., incoming links from two different resources or outgoing links to two different resources). For indirect links it is important that the resources are connected via the same link types. Further link types are weighted so that frequent link types obtain a lower weight. For recommendations in the music domain the author uses the DBpedia data set. The algorithm takes a seed URI as input and computes the distance between this URI and all other resources from the data set. To provide relevant recommendations the result is limited to instances of `dbpedia-owl:MusicArtist` and `dbpedia-owl:Band`. The algorithm remembers the links that have been used to explain the recommendations in a user-friendly way. Figure 6.6 visualizes the explanation interface of the system. It can be seen that Mutya Buena is recommended for people preferring Amy Winehouse, as Amy Winehouse is associated musical artist, act, and band of Mutya Buena. Passant determines the similarity between artists by applying a link-based distance measure on the Linking Open Data graph. In contrast our approach selects metadata that we consider appropriate to describe the music styles of artists and performs similarity calculations in the vector space. We focus on rich metadata profiles to unveil



Results for Amy Winehouse

horst.kl.dfki.de/Horst/action/search?artist=Amy+Winehouse&similarity=7&numresults=10

Horst

Similarity type

Artist Profiles | Merged Genres | Genres | Last.FM

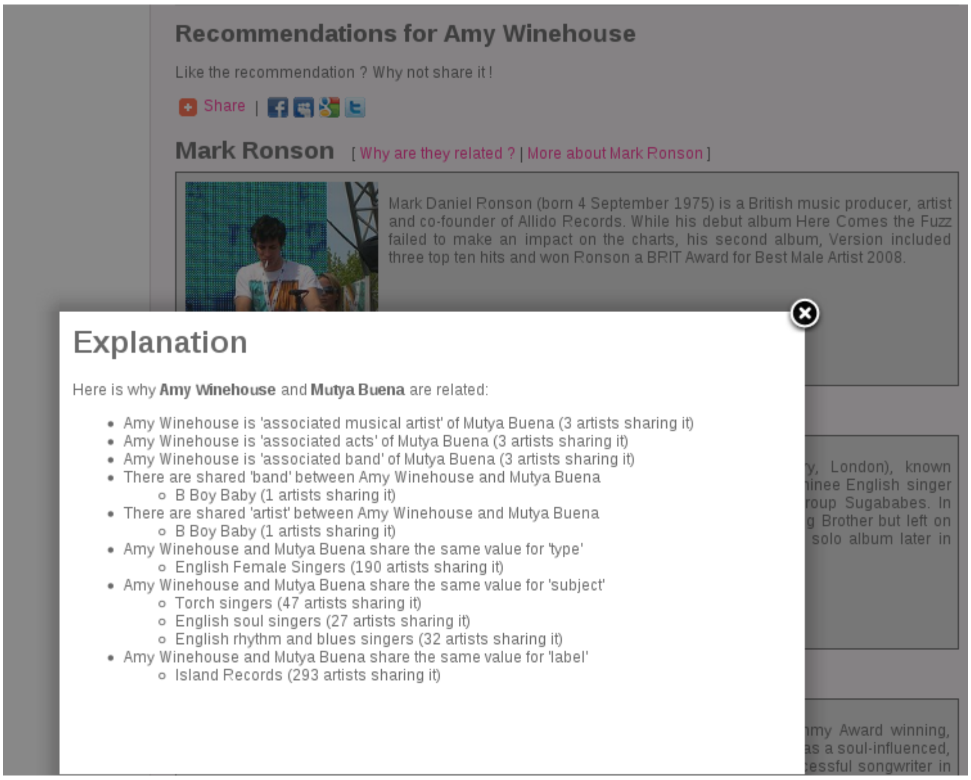
Your Query

Amy Winehouse

1	Kem	63%
2	Van Hunt	61%
3	Dwele	56%
4	Alicia Keys	54%
5	Vivian Green	49%
6	Cee-lo	46%
7	Joss Stone	45%
8	Amel Larrieux	45%
9	Kindred The Family Soul	45%
10	Amerie	43%

+

Figure 6.5: Screenshot of the HORST system showing similar artists for Amy Winehouse.



Recommendations for Amy Winehouse

Like the recommendation ? Why not share it !

Mark Ronson [Why are they related ? | More about Mark Ronson]

Mark Daniel Ronson (born 4 September 1975) is a British music producer, artist and co-founder of Allido Records. While his debut album Here Comes the Fuzz failed to make an impact on the charts, his second album, Version included three top ten hits and won Ronson a BRIT Award for Best Male Artist 2008.

Explanation

Here is why **Amy Winehouse** and **Mutya Buena** are related:

- Amy Winehouse is 'associated musical artist' of Mutya Buena (3 artists sharing it)
- Amy Winehouse is 'associated acts' of Mutya Buena (3 artists sharing it)
- Amy Winehouse is 'associated band' of Mutya Buena (3 artists sharing it)
- There are shared 'band' between Amy Winehouse and Mutya Buena
  - B Boy Baby (1 artists sharing it)
- There are shared 'artist' between Amy Winehouse and Mutya Buena
  - B Boy Baby (1 artists sharing it)
- Amy Winehouse and Mutya Buena share the same value for 'type'
  - English Female Singers (190 artists sharing it)
- Amy Winehouse and Mutya Buena share the same value for 'subject'
  - Torch singers (47 artists sharing it)
  - English soul singers (27 artists sharing it)
  - English rhythm and blues singers (32 artists sharing it)
- Amy Winehouse and Mutya Buena share the same value for 'label'
  - Island Records (293 artists sharing it)

y, London), known
inee English singer
roup Sugababes. In
g Brother but left on
solo album later in

mmy Award winning,
as a soul-influenced,
essful songwriter in

Figure 6.6: Explanations in the dbrec music recommender system.

unobvious similarities between artists hence enabling recommendations with a high degree of novelty.

A semantic-enhanced collaborative filtering method is presented by [Wang and Kong, 2007]. Besides similarity calculations based on the user-item rating matrix they exploit category information of items and demographic information of users. Their algorithm works as follows: In the first step they build a domain ontology of categories and assign items to these categories. Each item can belong to many categories. The similarity between items is calculated on the basis of the ratio of their shared categories. In the second step Wang and Kong perform user clustering. For that purpose they first calculate the similarity between users as a weighted score based on

- the Pearson correlation of their item ratings,
- their demographic similarity,
- and the similarity of their interests and preferences based on the semantic similarity of their rated items.

For clustering the K-Means algorithm is used. Recommendations are then generated based on the ratings of the users in the active user's cluster. Further the items are restricted to those matching the active user's predominant interest category. Wang and Kong show that their approach outperforms the precision of traditional collaborative filtering systems while at the same time not suffering from performance issues on large scale data sets. In contrast to Wang and Kong we use metadata from the Semantic Web to enhance content-based recommendations. We consider content-based methods more appropriate to provide access to the long tail of seldom rated resources as they do not require an item to have a certain number of ratings to be recommendable. Also people with extraordinary tastes can profit from content-based methods as they do not require to find similar users from which recommendations can be generated. However the quality of recommendations can be a concern in content-based systems and should therefore be considered carefully.

In [Baumann et al., 2010] we proposed an approach to identify similar artists based on Semantic Web metadata. For each artist a metadata profile is built consisting of combined genres and record release years that are associated with the respective artist. The profiles are mapped to a vector space. To reduce the noise in the data and to cope with features with low information content we remove very rare and very frequent features. Then the features are weighted using the TF-IDF measure. To determine the similarity between artists we calculate the pairwise cosine similarity between their weighted feature vectors. The evaluation of the approach has shown that this approach leads to more high quality novel artist recommendations than well-known systems such as Last.fm or Echo Nest.[12] However the overall recommendation accuracy leaves room for further improvement. The multifaceted user profiles proposed in this thesis can directly be combined with our method to determine similar artists. E. g., we can retrieve representative artists for a preferred music style of a user and find other artists matching this music style that are presumably unknown to the user.

[12] http://the.echonest.com/

Chapter 7

Evaluation

When developing a new recommendation algorithm a solid evaluation of the approach constitutes an important element in the process. However there is constant disagreement in the community on how recommendation algorithms should be evaluated. In recent years recommender systems have mostly been evaluated by using a data set of user-item preferences and predicting certain withheld values. The results were then analyzed by using one of the metrics discussed in Appendix E.4 ([Herlocker et al., 2004]). However it has often been claimed that such an offline analysis is not enough (e. g., [McNee et al., 2006b]) as this approach only predicts what is already known in the data set hence not taking into account the novelty and serendipity aspects of recommendations. In a panel discussion at the Recommender Systems 2010 conference Professor Joseph Konstan proposed the integration of new recommendation algorithms into existing platforms and asking the users about the perceived quality of the recommendations that way directly obtaining feedback from the users about the performance of the recommendation algorithm. However an integration into an existing platform is not always feasible.

In this thesis we proposed an algorithm extracting contextualized user profiles taking into account multifaceted user preferences. Each preference is represented by a label that should allow for a retrieval of resources that are relevant for the respective topic of interest or preferred music style. Hence we evaluate how specific the labels are for the identified preference by making use of measures known from the field of information retrieval. We argue that specific labels will allow for a future retrieval of relevant resources that a user will enjoy.

The current chapter is structured as follows: in Section 7.1 we depict evaluation goals proposed in the recommender systems literature and discuss on which of these goals we focus with our experiments. Next, in Section 7.2 we report on the challenges we faced evaluating our proposed approach and explain the reasoning behind the method applied. In Section 7.3 we present the results of the evaluation experiments for our topic-based resource recommendation approach. Our analyses here were twofold: We performed a first subjective evaluation study with a group of eight staff members of DFKI. It was a small sample biased towards researchers and software engineers. For that purpose, in the second phase we made a larger objective analysis using the data base of the BibSonomy system. Finally, in Section 7.4 we present the results for the mood-based music recommendations. The evaluation was carried out in an offline experiment using a data set obtained from Last.fm. We first analyzed whether users listen to different styles of music and checked in a second step whether the identified preferred music styles correlate with specific mood categories.

7.1 Evaluation Based on User Goals

For a good evaluation of a recommender system it is important to consider the goals the users have with the system (in contrast to, e. g., marketer goals). We will describe different such goals according to [Herlocker et al., 2004].

Annotation in Context Early recommender systems applied filtering on structured discussion postings to help the user decide which posts are worth reading and which are not (e. g., the Tapestry system [Goldberg et al., 1992] or the GroupLens system [Resnick et al., 1994]). These systems require to keep the structure and context of the messages. They directly annotate postings with rating predictions. The usefulness of these recommenders depends on how well the system can distinguish between desired and undesired items. Further, coverage is of major importance, i. e., the system has to be able to generate rating predictions for all items the users view.

Find Good Items Shortly after Tapestry and GroupLens had been introduced, first recommender systems emerged that suggested specific items to their users (e. g., the Ringo system [Shardanand and Maes, 1995] for music albums and artists) by providing a ranked list of items together with prediction values that indicated how much a user would like the respective items. The “find good items” task is often seen as the core recommendation task and applied in many commercial systems. However in these systems the rating prediction is often not shown to the user.

The tasks "annotation in context" and "find good items" are most commonly evaluated. Subsequently user goals will be presented that are not (so often) addressed in the literature but are still considered useful to be evaluated.

Find All Good Items In general recommender systems deal with the problem of information overload. For that purpose for most of the systems it is enough to recommend some good items to users. However there are domains where it is crucial not to miss any relevant items. Examples comprise systems in the field of legal data bases but also for researchers it might be important to keep track of as many publications in their fields of interest as possible. In these scenarios the false negative rate needs to be sufficiently low and as in the case of "annotation in context" coverage of the approach is a major concern.

Recommender Sequence The task of recommending sequences of items can be important, e. g., when streaming music. Here the goal is to compose a playlist that is pleasing as a whole. However this task can also be important in E-Learning scenarios where sequences of learning objects have to be recommended (e. g., "First read this introduction, second that survey, ...").

Just Browse When talking to users of recommender systems Herlocker et al. found out that many of them used the systems without the intension to actually purchase anything. They just found it pleasant to browse through the items in the system. For those users the accuracy of the recommender might be less important than the graphical user interface, its ease of use, as well as the level and nature of the information presented.

Find Credible Recommender Another finding from discussions with users was that users do not automatically trust recommender systems. They play with the system in order to check whether the recommender matches their tastes. A system that is optimized for utility (and therefore, e. g., does not recommend items the user already knows) may in these scenarios fail to appear trustworthy as it does not recommend items the user is sure to enjoy.

To generate useful recommendations it is crucial that users express their preferences by contributing a large amount of ratings. Evaluating if and why users provide ratings is therefore another important evaluation task on which we will focus subsequently.

Improve Profile With this goal the assumption is that users provide ratings in order to improve their profiles and hence improve the quality of the recommendations they obtain.

Express Self For some users it is important to have a forum where they can express their opinions. They do not provide ratings in the first place to obtain better recommendations. For these users anonymity has to be considered. Some of them may want to disclose their identity to the other users of the systems while others may not. Further the ease of making the contribution is mentioned as an important aspect here. By encouraging this kind of self-expression the recommender platform may obtain more data that can be used to improve the quality of the recommendations.

Help Others Some users provide ratings supposing that the community benefits from them. In many cases these users also contribute ratings to express themselves (see previous goal) however this is not always necessarily the case.

Influence Others Sometimes users have the undesired explicit intension to influence others to view or purchase particular items. For many platforms it might be interesting to evaluate how good the recommender system can prevent this task.

Our system targets the tasks *find good items* and *recommender sequence.* To achieve the first goal, we evaluate how likely the labels extracted by our approach will enable the retrieval of relevant on topic resources. The second goal aims at judging sequences of recommended items as a whole, instead of evaluating each recommended item separately. For that purpose we test the specificity of our extracted topics. If both goals are achieved, we can generate recommendation lists that match different interest topics of the users and thus improve the diversity of the recommended items.

We do not aim at being able to calculate rating predictions for items as needed for the *annotation in context* task and we assume that in most platforms many relevant items for a user are available so that it will be sufficient to recommend only some of them. As we propose a switching hybrid recommender system that generates item-based collaborative filtering recommendations for new users we are confident that we can establish trust with new users as this algorithm is known to provide popular, high quality recommendations. However the task *find credible recommender* will also not be evaluated in our experiments.

Evaluating why users provide ratings is an important task for every deployed recommender system. The motivations for users to vote depend on numerous factors such as the recommendation algorithm, the system interface, privacy settings, etc. In this work we propose a generic recommendation strategy that can be used for platforms where users have multifaceted preferences. The reasons that motivate users to provide ratings have to be evaluated for every system individually for that reason such evaluations are not in the scope of this work. Subsequently we will report on the challenges we faced evaluating the recommendation approaches proposed in this thesis and explain the reasoning behind the method that has finally been applied.

7.2 Evaluation of Our Research Hypotheses

For the topic-based resource recommendations we have to evaluate the following hypotheses:

H1 Knowledge workers have different topics of interest.

H2 By applying topic detection algorithms on the users' preferred resources we can detect these topics.

H3 The detected topics can be used to generate recommendation lists with a high degree of diversity.

Concerning the mood-based music recommendations the following hypotheses have to be evaluated:

H4 Many people listen to different styles of music.

H5 An active user's preferred style of music depends on her mood.

The best way to validate whether these research hypotheses apply or not would be to integrate our proposed approach into an online platform and to perform an evaluation as live user experiment (cf. Appendix E). In this experiment the users would have to be asked about their actual topics of interest and preferred music styles, how well the labels extracted by our approach describe these interest topics and music styles, and of course about the perceived diversity and quality of the generated recommendations. To validate H5 the users could be asked whether they associate their identified preferred music styles with certain moods. Unfortunately, in the course of this thesis we did not have the chance to integrate our approach into an online platform with enough users.

H1 and H2 were first checked in a small user experiment using the ALOE system deployed in the Knowledge Management group of DFKI. However the participant group was rather small and biased. So we performed a second larger offline analysis on the data base of the BibSonomy system. The goal of this analysis was to check the specificity of the topic labels extracted by our method. The assumption is that a label is specific for a certain topic when a query with the relevant terms of its label will pick many or all resources within the topic cluster and few or none resources outside the cluster. We expect that specific labels will enable the retrieval of relevant, previously unknown on-topic resources for a user. However with this approach we cannot evaluate the quality of the recommendations which are provided on this basis. H4 has been evaluated in a similar offline experiment using listening histories from the Last.fm platform. Further we checked whether the users' preferred music styles we identified overlap with certain moods. For that purpose we crawled mood tags associated with artists from the Last.fm platform and checked the dominance of moods within the extracted clusters. We will report about the detailed results of our evaluation experiments in the following sections.

7.3 Evaluation of Topic-based Recommendations

As mentioned before, the evaluation of the topic-based resource recommendation approach has been performed in a two-staged process. In the first stage we conducted a user study in which we checked whether the users of the ALOE system could associate the topic labels that have been identified by our approach with their actual topics of interest. We report on the results of this study in Section 7.3.1. In the second stage we tested on a larger data set whether the terms of the extracted cluster labels are likely to find resources matching the associated topics. The results of this evaluation experiment are depicted in Section 7.3.2.

7.3.1 Subjective Evaluation Study

The subjective user evaluation study has been performed with a small participant group consisting of eight staff members of the Knowledge Management department of DFKI ([Schirru et al., 2010a]). Seven of the participants were researchers (junior to senior), one participant was a software engineer. Every participant had expressed preferences for at least twenty resources in the system. A questionnaire was sent to these users showing the terms which represented their identified topics of interest. For each of these topics the users had to answer three questions:

Q1: Has the topic of interest correctly been detected?

Q2: How would you describe the topic in your own words?

Q3: Would you like to get recommendations for the topic?

Table 7.1 shows for each participant of the evaluation experiment, how many topics were identified by the system, how many of them were classified as correctly identified by the user and for how many of these topics the user would appreciate recommendations.

Altogether 39 user interest topics have been identified, on average 4.9 per user. 32 of the topics were classified as correct by the users, i. e., on average four topics per user. For 27 topics the users said that they would appreciate resource recommendations, 3.4 on average per user. Each user on average classified 84.17% of her identified topics as correct. User C is an outlier, only one of three identified topics has been classified as correctly identified.

With the results of this first evaluation experiment we were confident that our approach could identify interest topics covering a broad range of the participants' preferences. In most cases the extracted labels where expressive enough so that the users could assign them to their corresponding actual interest topics. Also mostly recommendations for the identified topics were desired by the users. However the participant group was small and biased so we wanted to verify our findings in a

User	detected topics	correct topics	recommendations desired
A	3	3	3
B	2	2	2
C	3	1	1
D	10	9	7
E	3	3	3
F	9	6	6
G	6	5	2
H	3	3	3
Sum	39	32	27

Table 7.1: Results of the subjective user interests identification evaluation study. For each user the number of detected topics is juxtaposed to the number of correctly detected topics and the number of topics for which the user would appreciate recommendations.

larger experiment. For that purpose we performed an offline evaluation based on the data set of the BibSonomy platform. We report on the results subsequently.

7.3.2 Objective Evaluation Study

In a second larger evaluation study we examined whether the derived topic labels could separate a user's resources that are associated with a topic from the rest of the user's resources. That way we aim at verifying that users clearly have different specific interest topics and further test the likeliness of our extracted labels to find relevant resources matching the associated topics. A similar approach to evaluate cluster labels for topic-based recommendations has been applied by [Au Yeung et al., 2008]. Subsequently we present the data set that we used for our objective evaluation. Afterwards we report on the results and discuss the consequences that we derived for our approach.

Data Set

Currently there are not enough users that regularly use ALOE, thus making an expressive evaluation based on the data in the ALOE system impossible. So we decided to use the BibSonomy data set[1] which is freely available for research purposes. BibSonomy is a social sharing platform for bookmarks (URLs) and publication references [Hotho et al., 2006a]. The provided data set consists of four tables:

- Table *bibtex* stores the bibtex entries that are associated with a shared publication. It has 566,939 instances.
- Table *bookmark* contains the URLs and metadata of bookmark contributions. The data set comprises 275,410 instances.
- In table *tas* the tag assignments of bibtex entries and bookmarks are stored. There are 2,622,423 tag assignments by 6,569 users for 837,757 resources available. Figure 7.1 shows the distribution of the number of tags. Figure 7.2 presents the distribution of the number of tagged bookmarks.
- Further in table *relation* sub and super relations of tags are stored. The table has 11,292 instances.

From the figures we can observe a typical phenomenon of resource sharing platforms, i. e., the largest amount of users contributes only small or medium large amounts of content and metadata. For instance 5302 of the 6569 users ($\approx 80.71\%$) annotated 100 or less tags and 5833 users ($\approx 88.8\%$) contributed 100 or less tagged resources.

In ALOE the users can share files and bookmarks. We decided to focus on the BibSonomy bookmarks for our evaluation as we consider them to best represent the data and metadata in ALOE. As described in Chapter 5.3.1 we use titles and tags to construct metadata profiles of the resources from which the topics are determined. This metadata is also available for resources in the BibSonomy system. A minor difference is that in BibSonomy each resource only has one title annotated whereas in ALOE there is one title per user that has the resource in her portfolio.

When deploying a topic-based recommender system the current user interest topics should be extracted on a regular basis, e. g., once every week. We selected users that have contributed between 20 and 500 resources as we consider this amount as representative to be contributed in such a time interval. From 503 users matching

[1] Knowledge and Data Engineering Group, University of Kassel: Benchmark Folksonomy Data from BibSonomy, version of July 1st, 2010

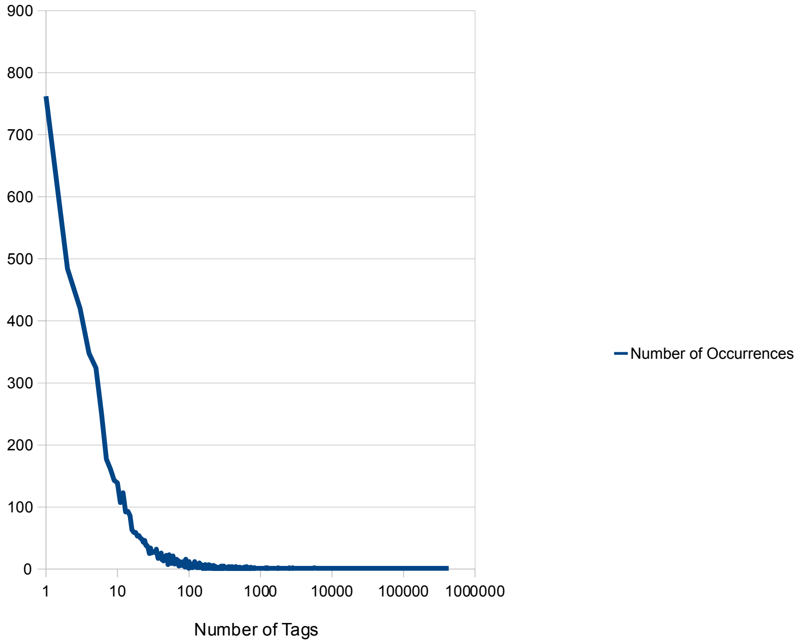


Figure 7.1: Distribution of the number of tags. The x-axis shows the number of tags in a logarithmic scale, the y-axis shows the number of users having the respective number of tags.

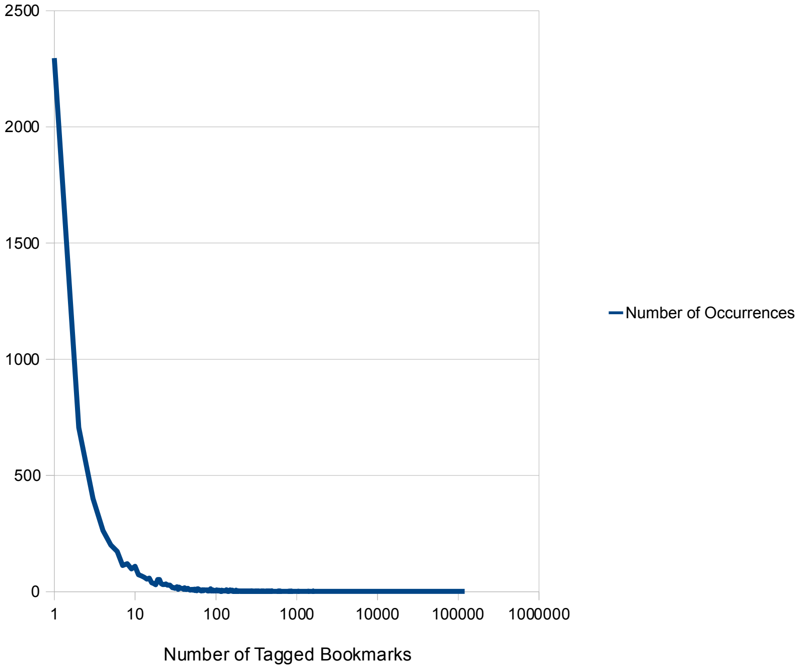


Figure 7.2: Distribution of the number of tagged bookmarks. The x-axis shows the number of bookmarks in a logarithmic scale, the y-axis shows the number of users having tagged the respective number of bookmarks.

these criteria we skipped the first 200 early adopters of the system. From the remaining 303 users our approach could detect topics for 296 of them. We analyzed the contributions of the remaining seven users for which no topics could be extracted. It was found that these users either did not provide the required metadata (four users just contributed ISBN numbers) or they contributed spam (e. g., online pharmacy). In our preprocessing steps the metadata profiles of these resources were deleted as the contained terms appeared either only once (e. g., in the case of the ISBN numbers) or they appeared in all resource profiles of the users' contributions.

Results

To evaluate how good the cluster labels can separate on-topic from off-topic resources we took for each user and each interest topic all relevant terms from the cluster label and queried the data base with them. We used an exact match query with an "AND" semantic. The selected terms of the topic labels had to appear in the tags, the title, or the description of a retrieved resource. We compared the retrieved resources to the set of resources the clustering algorithm had found for the respective topic. A term was considered as relevant, when its relevance value was at least $r\%$ of the relevance value of the most relevant term in the cluster. We controlled parameter r to see which term relevance threshold would provide the best results.

The measures precision, recall, and the F-measure have been used to evaluate our approach. These measures are commonly applied to evaluate systems in the field of information retrieval. The precision can be seen as a measure for the exactness of a search result. It is defined as follows:

$$precision = \frac{\mid \{relevant\, documents\} \cap \{retrieved\, documents\} \mid}{\mid \{retrieved\, documents\} \mid} \tag{7.1}$$

In contrast recall measures the completeness of a search result, i. e., how many of the relevant resources have been retrieved. It is defined as follows:

$$recall = \frac{\mid \{relevant\, documents\} \cap \{retrieved\, documents\} \mid}{\mid \{relevant\, documents\} \mid} \tag{7.2}$$

Often, an inverse relationship between precision and recall can be observed, where increasing the one at the cost of the other is possible. The F-measure considers both the precision and the recall of a search result and is therefore intended to remedy this issue. In particular we use the F_1 score for our evaluation, in which precision and recall are evenly weighted:

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{7.3}$$

With our approach we could identify 1403 topics for the 296 users, i. e., on average 4.74 topics per user. Figure 7.3 plots the precision, recall, and F-measure values of our experiments for different term relevance thresholds. The best results were achieved when only the most relevant terms were used for the query. With a term relevance threshold r of 100% the average precision per user was 0.85, the average recall was 0.42, and the average F-measure was 0.49. When examining the results in greater detail, we found that the accuracy of the clustering algorithm was less than perfect thus leading to off-topic resources in the clusters. Such resources are not intended to be found by our algorithm that way leading to decreased recall values. However with an average precision of 0.85 we are confident that our algorithm can recommend resources matching the users' topics of interest.

After having examined the average evaluation measures we wanted to check for how many users our approach could extract labels that are specific enough to enable the retrieval of on-topic resources. In Figure 7.4 we plot the rounded, average precision values at different term relevance thresholds against the cumulated percentage of users. We find that even with a term relevance threshold of 60% we can still achieve a rounded, average precision of 60% or better for 78.72% of the users. The best results are again obtained with a term relevance threshold of 100%. In this case a rounded, average precision of 80% or higher for 76.69% of the users can be achieved.

The objective evaluation study confirms the results of the smaller subjective study, namely that knowledge workers have different interest topics that can be identified by our approach and for which specific labels can be extracted. For that reason we accept H1 and H2 as valid hypotheses. Using the topic labels to retrieve resources for different interest topics of a user (cf. Chapter 5.3.2) we can generate recommendation lists with a high degree of diversity which leads us to the conjecture that H3 is also a valid hypothesis.

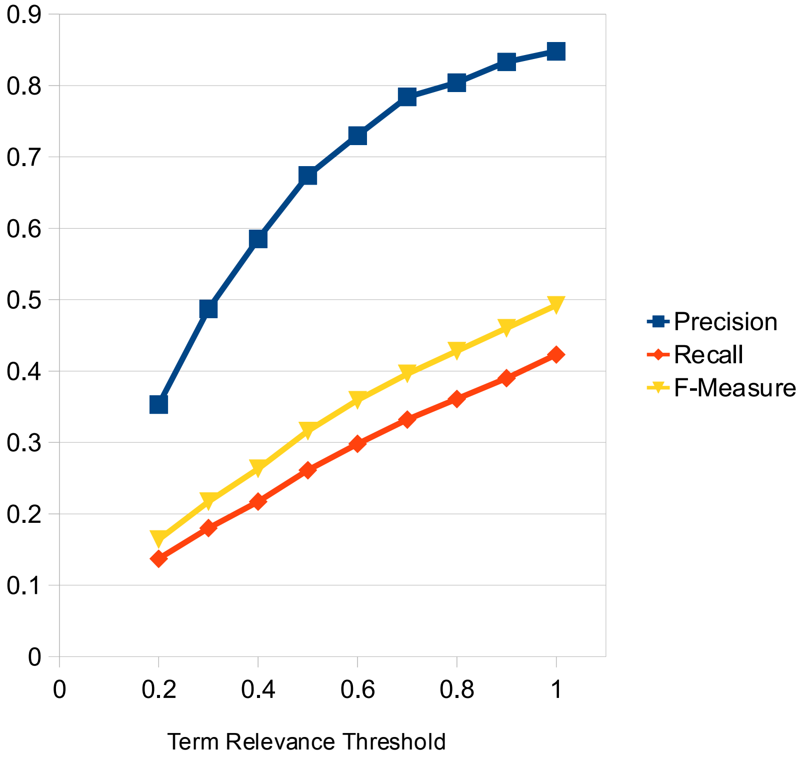


Figure 7.3: Evaluation of the topic labels by making use of precision, recall, and F-measure (y-axis). The x-axis shows the term relevance threshold that has been used as control parameter.

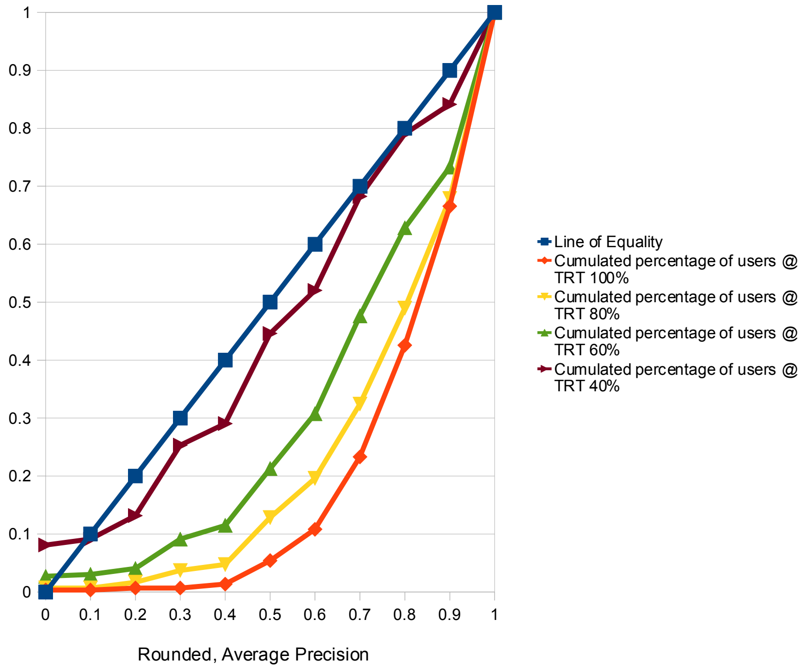


Figure 7.4: Lorenz curve plotting the rounded, average precision at different term relevance thresholds on the x-axis against the cumulated percentage of users on the y-axis.

Discussion

The goal of the objective evaluation study was to show that the cluster labels derived by our approach are capable of separating resources belonging to a topic from the rest of the resource. In our experiments we found that by including only the most relevant terms of the label the best results could be achieved. However, the subjective evaluation study we conducted in the first evaluation step indicated that users were able to associate cluster labels with their interest topics also when the term relevance threshold was set to 25% only. We therefore propose to relax the condition of using queries with only the most relevant topic terms thus obtaining more terms in the labels. Instead, our system uses a query with an “OR” semantic and boosts the terms according to their relevance values that way ranking those items best matching a topic label first. In other scenarios in which our topic extraction approach has been applied, we found that a term relevance threshold of 50% leads to reasonable results ([Schirru et al., 2010b]).

In Table 7.2 we show exemplarily the extracted user interest topics for five users that achieved high scores and one user with low scores on our evaluation measures at a term relevance threshold of 70%. For the users with high scores we can make three observations that we consider as typical for our approach:

1. We can identify different interest topics for the users either within a domain such as for user D in the domain of natural language processing or across domains such as for user B who is interested in literature (tolkien), JavaScript related programming technologies (jquery), video games (nintendo), etc.

2. Some topics are too broad to contribute to useful recommendations (e. g., *customer* and *research* for user A). These topics should be filtered out.

3. In platforms where resources in different languages are shared, multilinguality should be considered in the preprocessing steps, i. e., after a language identification step, stop word removal and stemming should be performed in a language specific way in order to avoid problems such as those appearing for user E (*einrichtung, zur, abkommen, für*).

In some cases our approach does not lead to good results. We observe the following conditions that lead to low scores on our selected evaluation measures:

1. The topic-based resource recommendations are not advantageous for users that share resources about random topics, e. g., all sorts of news articles instead of posting content that is related to specific interest topics.

2. In some cases the clustering results are suboptimal, for instance, when the number of clusters has not been detected correctly. If the cluster number is selected too small, some interest topics will not be recognized. Then again, if the cluster number is too big, different clusters will represent the same topic thus decreasing the diversity of the recommendation list.

3. Using a conjunctive query is sometimes too restrictive. For instance, an "OR" semantic in the query for user F and topic "*der, aol, welt*" would increase the recall value from 0.032 to 0.419 without decreasing the precision value. As stated before, we propose to relax the condition of using conjunctive queries and use an "OR" semantic in combination with weighting terms according to their identified relevance for the associated topic.

Table 7.2: Examples of extracted user interest topics at a term relevance threshold of 70%.

Topic Terms	**Precision**	**Recall**	**F-Measure**
User A			
web, social	1.000	0.750	0.857
innovation	0.300	1.000	0.462
eric, von, hippel	1.000	1.000	1.000
agile, development	1.000	0.333	0.500
networks, social	0.667	0.667	0.667
customer	1.000	1.000	1.000
research	0.500	1.000	0.667
User B			
infoscreen	1.000	0.600	0.750
zoomii.com, architektur, wohnen	1.000	0.333	0.500
nintendo	1.000	0.750	0.857
jugendbuch	1.000	0.632	0.774
tolkien	1.000	0.375	0.545
jquery	1.000	1.000	1.000
User C			
ebm	1.000	0.500	0.667
vaccine, hpv	1.000	0.400	0.571
genetic	0.500	1.000	0.667
nejm	1.000	1.000	1.000
jewish	0.500	1.000	0.667
medical	0.750	0.750	0.750
dermatology	1.000	1.000	1.000
hospitalist	1.000	1.000	1.000
User D			
tagger, pos	0.500	0.091	0.154
tools	1.000	0.667	0.800
gate	1.000	0.667	0.800
treetagger	1.000	1.000	1.000

Table 7.2: Examples of extracted user interest topics (continued).

Topic Terms	**Precision**	**Recall**	**F-Measure**
language, natural	1.000	1.000	1.000
User E			
wiener	0.667	0.667	0.667
locarno	0.667	1.000	0.800
patent	1.000	0.500	0.667
einrichtung, zur, abkommen, für	1.000	1.000	1.000
markenrecht	1.000	0.333	0.500
geschmacksmusterschutz	1.000	0.667	0.800
User F			
crm	1.000	0.333	0.500
vertrauen, media, trust	0.000	0.000	0.000
der, aol, welt	1.000	0.032	0.062

7.4 Evaluation of Mood-based Recommendations

The evaluation of the mood-based music recommendations was performed based on data from the Last.fm platform. The data set is described in Section 7.4.1. We performed the evaluation experiments in a two-staged process: First, we checked whether we could identify clearly different preferred music styles for the users. Our findings are presented in Section 7.4.2. Secondly, we tested whether the extracted preferred music styles correlate with mood categories found in the literature. The results of this experiment are depicted in Section 7.4.3. Finally, we conclude with a discussion of the results in Section 7.4.4.

7.4.1 Data Set

For the evaluation of our results we used number two of the "Music Recommendation Datasets for Research"[2] provided by Òscar Celma. The data set contains the full Last.fm listening histories (until 5th May 2009) of 992 users. The data was collected

[2] `http://www.dtic.upf.edu/~ocelma/MusicRecommendationDataset/index.html`

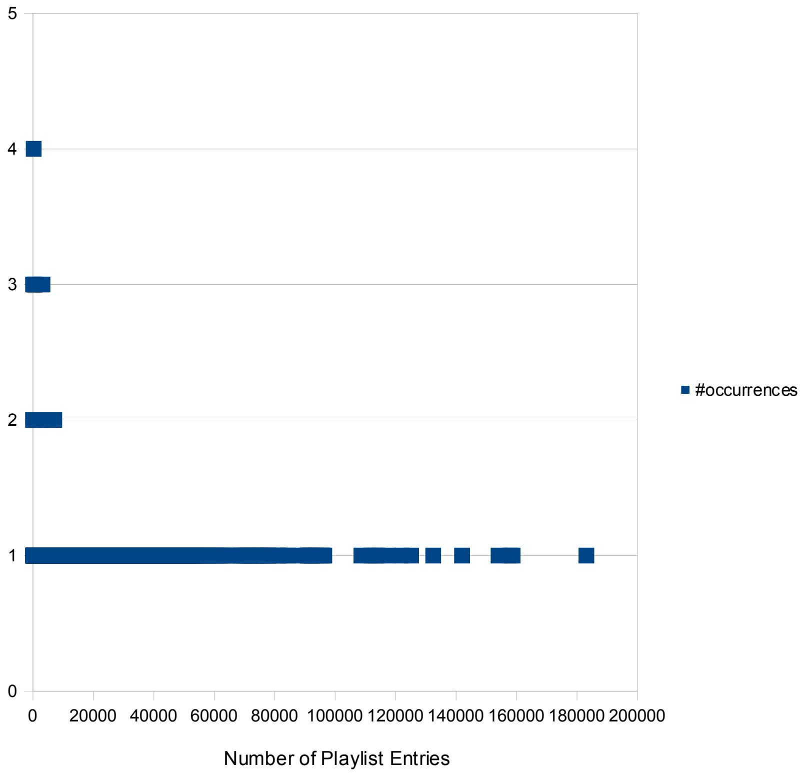


Figure 7.5: Distribution of the number of playlist entries. The x-axis shows the number of entries, the y-axis shows the number of users having the respective amount of entries in their playlist.

by using the *user.getRecentTracks()* method of the Last.fm API. Table 7.3 shows the metadata that is available in the data set. The data set comprises in total 19,150,819 playlist entries. The maximum number of playlist entries per user was 183,103. Figure 7.5 shows the distribution of the number of playlist entries per user. However, an interpretation of the graph is difficult as the range of playlist entries is much larger than the number of users. So we accumulated in Figure 7.6 the number of users for each number of playlist entries. For a point p on the x-axis we plot the number of users having p or less entries in their playlists. From this figure we can observe that the users having 50,000 or less items in their playlists account for more than 90% of all users.

The identification of a user's preferred music styles should be performed on a regular basis, e. g., once every week. We consider users that have listened to between 20 and 500 artists in such a time interval as typical users. Hence we include 442 users having their number of artists played in that range in our experiments.

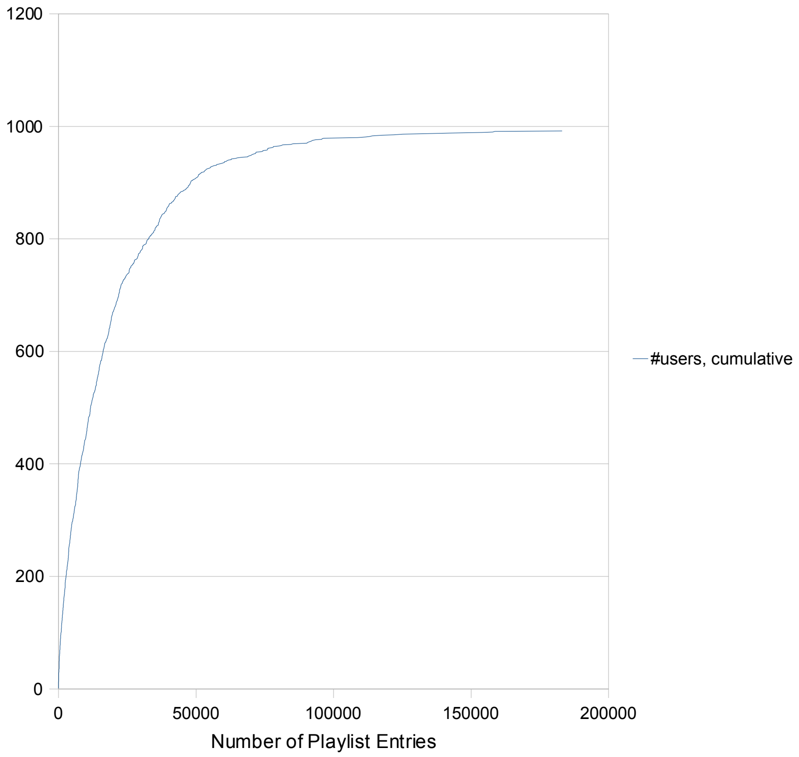


Figure 7.6: Cumulative distribution of the number of playlist entries. For each point p on the x-axis the number of users having p or less entries in their playlist are depicted on the y-axis.

Metadata Field	**Description**
user-id	Identifier of the user who owns the playlist entry.
timestamp	Point in time when the song was played.
artist-mbid	MusicBrainz identifier of the artist that is associated with the playlist entry.
artist-name	Name of the artist that is associated with the playlist entry.
song-mbid	MusicBrainz identifier of the song.
song-title	Name of the song.

Table 7.3: Available metadata for each playlist entry in the music recommendation data set.

7.4.2 Results: Identification of Preferred Music Styles

As with the evaluation of the cluster labels for the topic-based resource recommendations, the goal of the evaluation of the cluster labels for the mood-based music recommendations was to check how well the features of a label could distinguish the artists in a cluster from the rest of the artists. For that purpose we determined the support of a label in a cluster as well as its confidence.

According to [Witten and Frank, 2005] (page 69) the *support* of an association rule measures the number of instances to which the rule correctly applies. It may also be specified as the percentage of instances to which the rule correctly applies over the total number of instances instead. In the context of our cluster labels we define the support of a label in its associated cluster as follows:

$$\textit{support} = \frac{\#\textit{matching cluster items}}{\#\textit{cluster items}} \tag{7.4}$$

Please note that this definition of support corresponds to the definition of recall in information retrieval scenarios.

On the other hand *confidence* measures the number of instances to which a rule correctly applies, expressed as a proportion to the number of all instances to which it applies. For the evaluation of our cluster labels we define confidence as follows:

$$confidence = \frac{\#\textit{matching cluster items}}{\#\textit{matching items}} \tag{7.5}$$

Please note again that this definition of confidence corresponds to the definition of precision in information retrieval scenarios.

As the support can be increased at the cost of confidence and vice versa again we calculate their harmonic mean as our measure of choice to determine the best solution:

$$\textit{harmonic-mean} = 2 \cdot \frac{\textit{support} \cdot confidence}{\textit{support} + confidence} \tag{7.6}$$

In the evaluation experiment we wanted to show that our approach could generate labels that describe a user's preferred music styles precisely. When a label is specific for a music style it has a high support in its respective cluster (i. e., it matches many items in the cluster) and it has a high confidence (i. e., it does not match too many items outside the cluster).

In total our approach identified 3097 clusters for 442 users, i. e., on average 7.01 clusters per user. We tested two cluster labeling methods in order to identify a user's preferred music styles. The first was straight forward frequency-based cluster labeling. The second was the more sophisticated chi-square cluster labeling approach. Both methods are described in detail in Appendix C.4. As with the evaluation of the topic labels in Chapter 7.3.2 we controlled the feature relevance threshold, i. e., the parameter that indicates how relevant a feature has to be with respect to the most relevant cluster feature in order to be included in the label.

Our experiments show that the best results can be obtained by using the simple frequency-based cluster labeling approach when only the most relevant features are used. In this case an average support of 0.791 at an average confidence level of 0.65 with an average harmonic mean of 0.627 can be achieved. In Figure 7.7 we plot the detailed results for different term relevance thresholds. The measures show that many of the extracted music styles of a user are clearly distinct. Hence we conclude that H4 is a valid hypothesis. The results obtained using chi-square cluster labeling were slightly worse. Here the best results could also be obtained at a feature relevance threshold of 100% with an average support of 0.736 at an average confidence level of 0.544 and an average harmonic mean of 0.569. Figure 7.8 shows the detailed results for this approach.

Again we wanted to check for how many users our best approach could identify appropriate labels for preferred music styles. In Figure 7.9 we plot the Lorenz curve showing for specific support values the cumulated percentage of users at different feature relevance thresholds. For the cluster labels representing music styles we used the support values focusing on the recall of retrieved relevant artists. The assumption here is that users will perceive the recommendation of novel artists from their preferred music styles as particularly useful. As such artists are sometimes hard to find, in particular for niche genres, the support is preferred over the confidence at the cost of sometimes recommending artists that might not be useful for the active user. From Figure 7.9 we can observe that even at a feature relevance threshold of 60% our approach could identify cluster labels with a rounded, average support of 60% or better for 76.92% of the users. Using a feature relevance threshold of 100% we could detect labels with a support of 80% or better for 73.08% of all users.

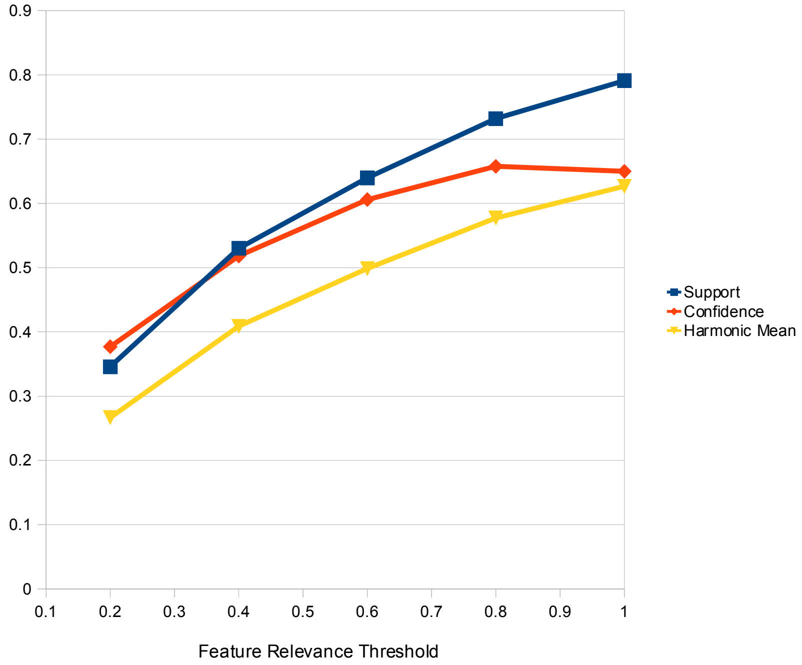


Figure 7.7: Evaluation of the labels describing preferred music styles obtained by using frequency-based cluster labeling. We plot the feature relevance threshold used on the x-axis against the achieved results in terms of support, confidence, and their harmonic mean on the y-axis.

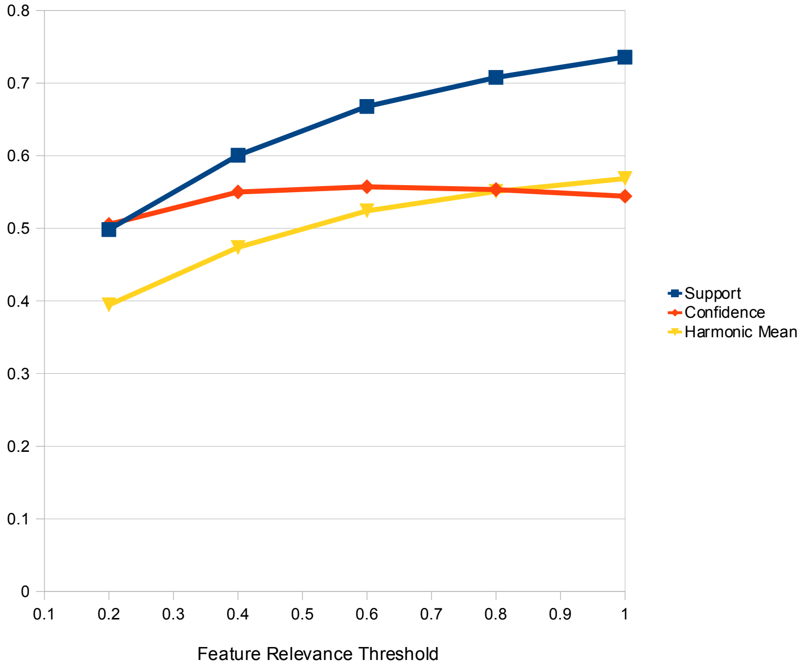


Figure 7.8: Evaluation of the labels describing preferred music styles obtained by using chi-square cluster labeling. We plot the feature relevance threshold used on the x-axis against the achieved results in terms of support, confidence, and their harmonic mean on the y-axis.

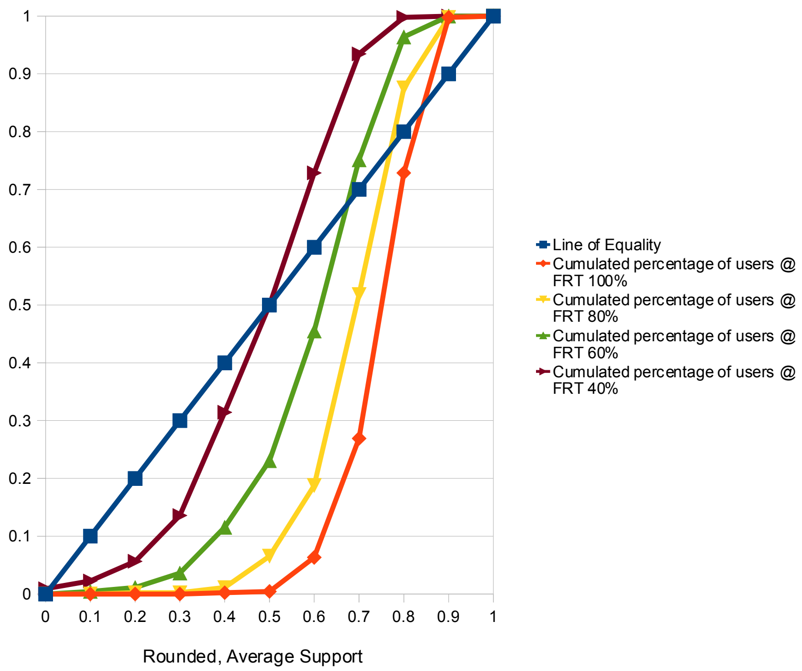


Figure 7.9: Lorenz curve plotting the rounded, average support at different feature relevance thresholds on the x-axis against the cumulated percentage of users on the y-axis.

7.4.3 Results: Mood Support in Clusters

In the last step of our evaluation we checked whether the clusters representing a user's preferred music styles correlate with specific moods. During our experiments we decided that the 18 mood categories proposed by [Hu et al., 2009a] are probably too many in order to lead to useful results hence we used the four mood categories identified by [Laurier et al., 2009] (cf. Appendix D). We crawled for each artist in our data set the tags from Last.fm and filtered out those tags that were not associated with one of the four mood categories. Then we determined for each category the mood support in each cluster and we stored the category with the maximum support in a cluster in our data base.

On average, we achieved a mood support of 0.358 in the clusters of a user. The value is too low in order to state that the clusters correlate with specific moods. In Table 7.4 we summarize in tenth part intervals the number of clusters with the respective mood supports. We find that some clusters have a high support with respect to a specific mood category (408 clusters with a support of 0.55 or higher) but most clusters are not specific for a certain mood (2689 with a support of less

Mood Support	**#Occurrences**
1	38
0.9	7
0.8	38
0.7	91
0.6	234
0.5	415
0.4	679
0.3	727
0.2	629
0.1	194
0.0	45

Table 7.4: Mood supports in tenth part intervals juxtaposed to the number of occurring clusters.

than 0.55) for that reason we have to reject hypothesis H5.

7.4.4 Discussion

Our evaluation shows clearly that many people have different preferred music styles. In Table 7.5 we depict the labels of the identified preferred music styles for five users using chi-square cluster labeling at a feature relevance threshold of 80%. From the examples we observe two points that we find typical for our approach:

1. Based on the choice of the feature space, the labels extracted consist mainly of genres hence being well suited for the recommendation of internet radio stations that a user might enjoy.
2. Sometimes the clustering seems to be too fine grained. For instance, user D has clusters such as “indie rock” and “indie rock, alternative rock”. It seems sensible to merge these clusters. This observation suggests that the heuristic on which reasonable cluster number to pick (cf. Table C.1) could be adapted in order to improve the results obtained.

In our experiments we could not find a correlation between the extracted clusters representing a user’s preferred music styles and specific moods according to the classification proposed by [Laurier et al., 2009]. In our approach we describe artists by their associated genres and instruments played. Most of the cluster labels consist of genre annotations that are associated with the artists in the respective clusters. However [Hu and Downie, 2007] found that genres and moods are independent from each other. Also they observed that many artists are associated with different moods.

Table 7.5: Examples of identified preferred music styles at a feature relevance threshold of 80% using chi-square cluster labeling.

Cluster Label	**Support**	**Confidence**	**Harmonic Mean**
User A			
salsa music	1.000	0.929	0.963
new age music	1.000	0.824	0.904
pop rock	0.846	0.688	0.759
trip hop, downtempo	0.548	0.944	0.693
trance music	0.464	0.929	0.619
ambient music	1.000	0.381	0.552
jazz	0.700	0.438	0.539
electronic music	1.000	0.229	0.373
alternative rock	1.000	0.033	0.064
electronic music	0.028	0.114	0.045
User B			
reggae	1.000	0.880	0.936
death metal	0.943	0.786	0.857
electronic music	0.818	0.871	0.844
hip hop	1.000	0.690	0.817
punk rock, hardcore punk	0.652	0.833	0.731
heavy metal, gothic metal	0.365	1.000	0.535
alternative rock, hard rock, rock music	0.051	1.000	0.097
User C			
reggae	1.000	0.952	0.975
folk rock	0.667	0.952	0.784

Table 7.5: Examples of identified preferred music styles (continued).

Cluster Label	**Support**	**Confidence**	**Harmonic Mean**
ambient music	0.714	0.833	0.769
rock music	1.000	0.519	0.683
post-rock	0.667	0.688	0.677
string instrument, keyboard instrument	0.529	0.500	0.514
post-rock	0.003	0.031	0.005
User D			
post-punk revival	1.000	0.889	0.941
heavy metal	0.944	0.850	0.895
indie pop	1.000	0.750	0.857
synthpop	0.917	0.733	0.815
soul music	1.000	0.542	0.703
punk rock	1.000	0.500	0.667
string instrument, keyboard instrument	0.683	0.636	0.659
hip hop	0.800	0.552	0.653
electronica	0.900	0.429	0.581
country	1.000	0.357	0.526
pop music	1.000	0.338	0.505
hip hop	1.000	0.310	0.473
house music	0.800	0.308	0.445
indie rock	1.000	0.260	0.413
alternative rock	0.696	0.182	0.289
indie rock	1.000	0.060	0.113
indie rock, alternative rock	0.019	0.094	0.032
User E			
southern rap	0.913	0.840	0.875
hard rock	0.868	0.805	0.835
post hardcore, screamo	0.650	1.000	0.788
indie rock	0.682	0.811	0.741
alternative hip hop	0.909	0.625	0.741

Table 7.5: Examples of identified preferred music styles (continued).

Cluster Label	**Support**	**Confidence**	**Harmonic Mean**
electronic music	0.941	0.571	0.711
electronic instruments	0.750	0.643	0.692
heavy metal	1.000	0.353	0.522
rhythm and blues, pop music, contemporary r&b	0.326	1.000	0.492
hardcore hip hop, east coast hip hop	0.320	1.000	0.485
string instrument	0.933	0.298	0.452
hip hop	1.000	0.162	0.279
contemporary r&b	1.000	0.059	0.111
hip hop	0.056	0.072	0.063

Chapter 8

Related Work

In this chapter we summarize and discuss related work in the fields of topic-based resource recommendations (Section 8.1) and mood-based music recommendations (Section 8.2) respectively.

8.1 Topic-based Resource Recommendations

An early recommender system that takes into account the user's different topics of interest was the Fab system, proposed by [Balabanovic and Shoham, 1997]. Fab is a distributed content-based, collaborative recommender system. The recommendation process is divided into two stages: First is the collection of items from the Web according to different interest topics. For each topic a collection agent exists that maintains a content-based profile thus allowing it to gather relevant pages for the respective topic. The topics are computer generated clusters of interests that track the changing preferences of the user population. Second is the selection stage. For each user, an individual selection agent maintains a content-based user profile, that way allowing the delivery of pages gathered from the collection agents according to the user's preferences. Figure 8.1 illustrates the two-staged process. In Fab the users provide feedback for resources by explicit ratings. The users' ratings are stored in each user's individual selection agent and are forwarded to the respective collection agents. The collection agents then adapt their profiles accordingly. Pages that have been rated highly by a user are recommended to other users with similar profiles. In Fab each collection agent retrieves items according to a topic of interest that is shared by many users. In contrast, our approach for topic-based resource recommendations

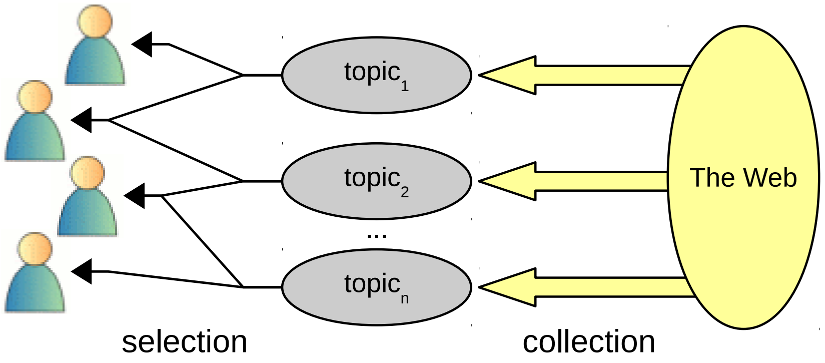


Figure 8.1: High level overview of the FAB recommender system ([Balabanovic and Shoham, 1997]).

determines each user's interest topics individually. That way we aim at extracting user profiles that are stronger tailored to a user's individual preferences. Differently from the Fab system, our approach does not consider similarities between users when providing topic-based recommendations hence not requiring to find peers for a user and allowing access to rarely rated resources. As described in Chapter 2.3, pure content-based recommender systems do not take into account the quality of items. Considering the ratings of similar users could be an interesting task for our future work as it provides opportunities to improve the quality of the recommendations.

A folksonomy-based approach for user interests identification in collaborative tagging systems has been proposed by [Au Yeung et al., 2008]. Assuming that the resources and tags posted to such systems highly depend on the user interests, the authors use the folksonomies in these systems to build topic-based user profiles. The authors propose a network analysis technique applied on the personomy of the users to identify their different topics of interest. A personomy is defined as the part of the folksonomy that is restricted to the tags, documents, and annotations of one particular user. It is represented as a bipartite graph with the tags and documents as vertices and associated annotations as edges. To enable clustering, this graph has to be converted into a one-mode network (e. g., the network of documents where the edges between the documents are weighted with the number of common tags). Then clustering is performed based on modularity optimization over the network using a greedy algorithm. For each cluster the authors extract a signature consisting of the tags that appear with more than a certain percentage of the documents in the cluster. Finally a user profile is returned as a set of these signatures. Yeung et al. identify a set of tags per user interest topic. This is similar to our approach extracting a weighted term vector for each interest topic of a user. However, a major difference between the two approaches is that Yeung et al. do not exploit the user-contributed metadata of other users for shared bookmarks. Hence, collective intelligence in not harnessed to obtain a proper resource description. From our perspective this has two drawbacks. First, a user's profile extracted with the proposed approach only reflects her personal vocabulary, thus ignoring how other users would describe bookmarks of certain topics. We assume that this might complicate the process of retrieving resources matching a specific topic. Second, for users that do not tag their resources interest topics cannot be extracted. However, the authors report that this applies to only 246 users of their test set of 9,431 users.

[Guo and Joshi, 2010] propose a topic-based recommendation framework integrating the tag annotations of individual users, user communities, and all users of a collaborative tagging system. They apply a modified Latent Dirichlet Allocation model (LDA, [Blei et al., 2003]) to cluster users and tags simultaneously, thus obtaining the implicit linking of tags and users. The generalized description of resources and users is expected to alleviate the noise in the tag data as well as the problem of data sparsity. The authors assume a fixed number of 100 topics. They calculate vectors that determine the degree of affiliation of each resource, user, user community, and query term to each topic. Recommendations are then provided by first combining the vectors of the query terms, the active user, and the community of the active user. Second, the top five topics are selected and all bookmarks with a high degree of affiliation for the selected topics are found in the data base. Resources whose feature vectors have a cosine similarity of more than 0.75 with the combined feature vector of the query, the active user, and the community of the active user are ordered and shown to the user. Offline analyses performed on data from the Delicious system show that the proposed recommendation method can alleviate data sparsity and provide more effective recommendations than previous methods. The approach proposed by Guo and Joshi represents resources and queries as vectors in which each dimension represents the degree of affiliation to a topic in the LDA model. In contrast, our approach uses the model built by the NMF algorithm only to derive the contextualized user interest profiles. Resources are represented by the metadata annotated by the community of users. When retrieving resources for a selected topic, we exclusively rely on the match between the topic terms and the metadata profiles of the resources. Further, Guo and Joshi assign each user to a topic community based on the user's interest value for the associated topic. Our method does not assume one predominant interest topic for a user. It retrieves resources for different interest topics each with equal weight.

Another user modeling approach that takes into account a user's different topics of interest is presented by [Middleton et al., 2001]. The authors describe the Quickstep recommender system which unobtrusively monitors the browsing behavior of its users. The target users of the system are scientists that need to be informed about new papers in their field of interest as well as older papers relating to their work. The system applies supervised machine learning coupled with an ontological representation of topics to elicit user preferences. It uses a multi-class behavioral model with classes representing paper topics, that way allowing domain knowledge to be used when the user profile is constructed. The system works as follows: User browsing behavior is monitored unobtrusively via a proxy server that logs every URL browsed during the user's working activity. Overnight, a machine learning algorithm classifies browsed URLs and saves the classified papers in a paper store. The interest profile is derived from explicit feedback and browsed topics. Recommendations are computed based on the user's current topics of interest and the classified paper topics. The generated recommendation lists contain items from the user's three most current topics of interest. Quickstep uses a research paper topic ontology that is based on the dmoz[1] taxonomy of computer science topics. In the ALOE system deployed at the Knowledge Management department of DFKI, employees share resources about their research interests, software engineering, and about topics in which they are privately interested. Designing a topic ontology for such an open world scenario does not seem to be feasible. Hence, our approach uses algorithms from the domain of topic detection and tracking in order to identify user interest topics automatically.

[Ziegler et al., 2005] aim at improving topic diversification by balancing top-N recommendation lists according to the users' full ranges of interests. In their recommender system each item is associated with features from a domain taxonomy like, e. g., author, genre, and audience in the domain of books. The proposed algorithm takes a top-N recommendation list and selects a (much) smaller subset of items with a low degree of intra-list similarity. The final recommendation list is built by

[1] `http://www.dmoz.org/`

gradually adding items that keep intra-list similarity low and are recommendable according to traditional collaborative filtering algorithms (see Chapter 2.2). The approach presented by Ziegler et al. assumes that features from a domain taxonomy are annotated for each item. In Enterprise 2.0 platforms such features are not always available as resources are contributed by the community of users and many systems do not want to place the burden of annotating content with concepts from a formal taxonomy on the users. Instead they rely on lightweight approaches such as tagging in order to classify content.

[Zhang and Hurley, 2009] propose an approach to improve the diversity of recommendations in collaborative information filters. To capture a user's full range of interests, her preferred resources are partitioned into clusters of similar items. Recommendations are then composed of items that match with the clusters instead of matching the whole user profile. The algorithm favors those clusters whose items are least similar to the average user taste, that way also improving the novelty aspect of the recommendations. To evaluate the effectiveness of their approach, the authors plot concentration curves of item novelty against recommendation accuracy. Their experiments show that the proposed approach reduces the bias of traditional collaborative information filters towards items that are similar to the average user taste at a small cost to overall accuracy. The underlying idea proposed by Zhang and Hurley is similar to our approach. Both methods partition a user's preferred items to obtain groups of similar items and then generate recommendations matching these different groups, thus improving recommendation diversity. The difference is that Zhang and Hurley apply a collaborative filtering algorithm while we implemented a content-based approach. In Enterprise 2.0 platforms where the number of users is small compared to the number of items, collaborative filtering suffers from severe sparsity issues. Hence a content-based approach matching the metadata profiles of items seemed to be an appropriate choice for our setting.

In Table 8.1 we summarize the main differences between the related topic-based recommender systems and our approach.

8.2 Mood-based Music Recommendations

[Lee and Lee, 2006] propose M^3 (Music for My Mood) a music recommendation system based on the user's intention and mood. The application layer of the system

Related Work	**Our Approach**
Topics shared by many users [Balabanovic and Shoham, 1997]	Topics tailored individually to each user
Requires peers to ensure quality [Balabanovic and Shoham, 1997]	Does not require peers but quality cannot be guaranteed
Topic extraction is based on own metadata annotations [Au Yeung et al., 2008]	Topic extraction is based on metadata annotations of potentially many users
Requires a domain ontology/taxonomy [Middleton et al., 2001, Ziegler et al., 2005]	Learns interest topics automatically
Collaborative filtering recommendation approach [Zhang and Hurley, 2009]	Main focus on content-based filtering

Table 8.1: Comparison with related topic-based recommender systems.

consists of three components: First, the Intention Module determines whether a user wants to listen to music. Therefore it uses the user's past listening history and environmental context data such as weather data and calendric information. Second, the Mood Module identifies the style of music that is suitable for the user's current context. M^3 distinguishes between three mood categories: slow music (ballads and R&B), any music (no preferred genre), and fast music (rock/metal and dance). This component also utilizes the user's listening history as well as the environmental context data. Both Intention Module and Mood Module are implemented by making use of CBR techniques. Third, the Recommendation Module generates music recommendations that are suitable for the user's detected mood. It recommends the top 15 songs from the corresponding genres most often played by the user in the last week. The system presented by [Lee and Lee, 2006] implements a static mapping between mood and preferred music style that is common to all users. In contrast, our approach assumes that when Alice is in a good mood she might listen to different music styles than Bob when he is in a good mood. Hence we target at more personalized context-sensitive recommendations. Further the approach presented by Lee and Lee only recommends songs a user has previously listened to. Our method aims at recommending novel artists that are presumably unknown to the active user, that way providing access to the long tail of artists that are not so well-known.

A mood-based music recommender system based on collaborative filtering was proposed by [Mortensen et al., 2008]. The authors conducted a mood-based recommendation experiment that was split into three phases. The first phase comprised a period of 19 days in which content-based recommendations were generated. In this phase recommendations were provided according to the users' listening histories ("more like I heard already") and the genres the users liked. Every song played was rated by the users on a binary scale (like/dislike) and annotated with the user's perceived mood (one of angry, happy, relaxed, or sad). Based on the ratings gathered during the first 19 days, in the second phase recommendations where generated with a collaborative filtering algorithm. In the third phase recommendations were also provided based on collaborative filtering however those songs that did not match a user's current mood were filtered out. The users stated their current mood explicitly in the interface. Again the recommendations were only calculated based on the data collected during the 19 days of phase one. The evaluation experiments show that collaborative mood filtering outperforms the content-based approach and significantly outperforms the collaborative filtering approach that does not take into account the users' moods. The system proposed by [Mortensen et al., 2008] uses mood as additional filter for recommendations based on collaborative filtering. Hence it suffers from the same drawbacks as the system proposed by [Wang and Kong, 2007] (cf. Chapter 6.4), i. e., in particular recommendations of popular items. Mood-based recommendations are generated based on mood annotations provided by the users of the system. In our approach we tried to automatically find a correlation between a user's mood and her currently preferred music style. For that purpose, a mood model based on social tags has been applied. However, the evaluation of our approach could not confirm such a correlation.

[Rho et al., 2009] propose a context-based music recommendation system that employs support vector regression (SVR) to map musical feature vectors to moods. An ontology-based context model is used to determine the users' current situation and mood. The music mood classifier based on SVR consists of three parts. First, seven distinct musical features are extracted and analyzed. These features are: pitch, tempo, loudness, tonality, key, rhythm, and harmonics. Second, the training module determines the arousal and valence of each song by using SVR on the analyzed musical features. This process is performed according to the acoustic and emotion rules defined by [Juslin and Sloboda, 2001]. In the third step, the mood mapper maps the arousal and valence values to one of eleven moods. These moods are: angry, bored, calm, excited, happy, nervous, peaceful, pleased, relaxed, sad, and sleepy. Further the authors developed COMUS, an ontology consisting of 826 classes and instances and 61 property definitions modeling domains such as genres, persons, mood, and situations ([Song et al., 2009]). The system considers a user's situational context and her favorite mood obtained from her profile and listening history to generate context-based music recommendations. The approach presented by Rho et al. assumes that a user has set her musical preferences manually (e. g., genre and mood). In contrast, our method learns a user's preferred music styles. Our intention was to also learn a mapping between the user's mood and her currently preferred music style. However, the metadata used to describe artists (in particular the genres) do not correlate with specific moods. In our future work we will investigate whether we can find groups of music styles that allow for an automatic mapping of mood and preferred music style.

Part III

Conclusion

Chapter 9

Summary

In this thesis we have presented an approach extracting contextualized user profiles which enable recommendations taking into account a user's full range of interests. The method applies algorithms from the domain of topic detection and tracking to identify diverse user interests and to represent them with descriptive labels. The approach was tested in two scenarios: First, we implemented a content-based recommender system for our Enterprise 2.0 resource sharing platform ALOE where the contextualized user interest profiles were used to generate recommendations with a high degree of inter-topic diversity. The evaluation experiments have shown that our approach is likely to capture a multitude of interest topics per user. The labels extracted are specific for these topics and can be used to retrieve relevant on-topic resources. Second, a slightly adapted variation of the algorithm was used to target music recommendations based on the user's current mood. The evaluation experiments conducted show that users clearly have a multitude of different preferred music styles. However, a correlation between these music styles and music mood categories could not be observed. In Section 9.1 we discuss the research hypotheses posed at the beginning of our work. Next, in Section 9.2 we summarize the research contributions made in this thesis. Finally, we discuss in Section 9.3 the limitations of our approach and the evaluation experiments performed.

9.1 Discussion of Research Hypotheses

For topic-based resource recommendations we set the following hypotheses:

H1 Knowledge workers have different topics of interest.

H2 By applying topic detection algorithms on the users' preferred resources we can detect these topics.

H3 The detected topics can be used to provide recommendation lists with a high degree of diversity.

First, we want to tackle H1 and H2. A preliminary evaluation study was conducted with eight users of the ALOE system deployed at the Knowledge Management department of DFKI. In this study we found first evidence that knowledge workers have different topics of interest which are also reflected in our Enterprise 2.0 resource sharing platform. By making use of a topic extraction algorithm based on non-negative matrix factorization we identified labels characterizing the users' different topics of interest (cf. Chapter 5.3.1). On average 4.9 topics were identified per user of which on average four were classified as correctly identified by the users. In a second, larger evaluation study using the data base of the BibSonomy system we could further verify the validity of H1 and H2. For 296 users we could identify 1043 topics, i. e., on average 4.73 topics per user. To check how likely our topic labels can retrieve on-topic resources we followed an evaluation approach proposed by [Au Yeung et al., 2008] where we determined precision and recall for the recommendations generated by our approach (within a user's own resources) compared with the results obtained from the NMF algorithm. With the best configuration we could obtain a precision of 0.85 at a recall level of 0.42. Plotting the Lorenz curve showed that with this configuration we could achieve a rounded, average precision of 80% or better for 76.69% of the users. Our approach requires that users have different interest topics according to which they share and interact with resources in the platform. For those users that interact with items from random topics (e. g., users contributing all sorts of news articles from the field of computer science) the extraction of meaningful topic labels is difficult and the approach can thus not be expected to be advantageous compared to traditional content-based recommendation approaches. When examining some examples in detail we find that, despite good precision and recall values, some topic labels are too broad in order to generate useful recommendations. These labels should be filtered out. With the remaining labels we are confident that useful recommendations according to different interest topics of the users can be generated thus leading us to the conjecture that also H3 is a valid hypothesis.

For mood-based music recommendations we set the following hypotheses:

H4 Many people listen to different styles of music.

H5 An active user's preferred style of music depends on her mood.

We were able to confirm H4 with our approach to identify a user's different preferred music styles that has been presented in Chapter 6.2. Using the Last.fm playlists of 442 users we clustered for each user the artists she had listened to and extracted labels for the identified groups. In total 3097 clusters were identified for the 442 users, i. e., on average 7.01 clusters per user. We found that the labels extracted were specific for the respective music styles achieving an average support of 0.761 at a confidence level of 0.604 and a harmonic mean of 0.586. Plotting the associated Lorenz curve showed that with the best configuration a rounded, average support of 80% or better could be achieved for 73.08% of the users. In contrast, H5 could not be confirmed. In our best configuration we achieved an average mood support of 0.358 in the clusters. We had 408 clusters with a mood support of 0.55 or better that were juxtaposed to 2689 clusters where the mood support was less than 0.55. Hence we cannot observe a correlation between our identified preferred music styles and associated moods. Currently the music styles detected by our approach are mainly described by the genres that are associated with the artists in the respective clusters. Our findings are in accordance with those of [Hu and Downie, 2007] who found that genres and moods are independent from each other. Further they observed that many artists are associated with different moods.

9.2 Research Contributions

In this thesis we have shown that users of social sharing platforms (and in particular knowledge workers) have different topics of interest that should be considered when resource recommendations are provided. We have proposed an approach that exploits user-generated metadata to describe resources. The metadata profiles of each user's preferred items are used to automatically identify her interest topics which are then represented in a multifaceted user profile. Further we have suggested a switching hybrid recommender system that generates item-based collaborative filtering recommendations for new users and content-based recommendations for those users for whom enough preference expressions are available. The content-based approach takes into account a user's different topics of interest and provides recommendations with a high degree of diversity.

A slightly adapted variation of our method extracting contextualized user interest profiles was successfully used to show that users have different preferred music styles. In this scenario a user's preferred music artists were described by making use of semantic data from the Linked Open Data cloud. The artist profiles were clustered and labels describing the music styles associated with the artists in the clusters were extracted. Even though a correlation between these identified music styles and specific mood groups could not be found, we are confident that the multifaceted user profiles obtained that way can be used for context-sensitive music recommendations when integrated with a general user model ontology such as GUMO ([Heckmann et al., 2005]) or a dedicated music ontology such as COMUS ([Song et al., 2009]).

9.3 Limitations

Finally, we want to discuss some of the limitations of this thesis. The focus of the work at hand is on the extraction of the user profiles. The perceived quality of real world recommendations based on the contextualized user profiles has not been evaluated so far. Traditionally, recommender systems (and in particular collaborative filtering systems) are evaluated in offline experiments (cf. Appendix E). For this method a data base of user-item ratings is used from which a certain percentage of ratings is removed. Based on the remaining data the user models are learned and rating predictions are calculated for the withheld items. Then the predictive accuracy is measured by comparing these predictions with the actual user ratings using error measures such as mean absolute error or root mean squared error (cf. Appendix B). The problem of this method is, that recommendations of novel items that the respective user has not rated yet, cannot be evaluated. However, besides diversity of recommendation lists, the novelty of the provided recommendations is a major goal of our approach. Further, novelty and diversity always have to be evaluated together with the perceived quality of the recommendations ([Celma and Herrera, 2008]) which can only be done in live user experiments. Finding an online platform where our method can be integrated and evaluated is still an important point for our future work.

The evaluation experiments conducted thus far are performed for each user independently based on her resource contributions. In some cases the extracted labels achieve good evaluation results in our experiments but are not useful to generate topic-based recommendations in a real world setting, for instance, when the terms in the cluster label are too broad as reported for user A in Table 7.2. Concerning the probability of our method to retrieve relevant on-topic resources in a real word setting, it should be noted that we implemented a similar approach for a social media mining tool where the topic extraction was applied on a set of domain-specific blog postings ([Schirru et al., 2010b]). In that system we also used a content-based recommendation approach based on the relevant terms of the topic labels. The evaluation experiments showed that the precision calculated over the top 10 recommended blog articles was 0.87 with the best configuration showing that the method can recommend relevant resources for a topic.

The offline evaluation experiments conducted in this thesis only compare different parameterizations (term or feature relevance thresholds) of the proposed approach. The use of different clustering algorithms has not been evaluated as from our point of view there is no single best clustering algorithm for our method. Instead the choice of the most appropriate clustering strategy should depend on the underlying data and has to be decided for every system individually. For the identification of a user's topics of interest in an Enterprise 2.0 platform we used non-negative matrix factorization. For topic modeling with text documents also algorithms such as latent dirichlet allocation [Blei et al., 2003] are known to lead to good results. In order to cluster artists that are described by metadata features obtained from LOD sources we experimented with the K-Means algorithm that we considered more appropriate for this problem domain. Comparisons with other approaches proposed in the literature are difficult as measures to judge the diversity of recommendation lists (e. g., the intra-list similarity metric proposed by [Ziegler et al., 2005]) have not been applied widely. Also the data sets on which the evaluation experiments have been conducted are often not published which makes a comparison with related methods difficult (e. g., [Au Yeung et al., 2008]).

Another limitation of this thesis that has often been discussed is the feature space used in the music recommendation use case. Currently our metadata profiles for artists consist of two different feature types only: first, the genres that are associated with the artist, and second, his/her instruments played together with the respective instrument families as found in the Freebase data set. We consider this selection of features as a starting point when describing artists using Linked Open Data. It should be noted that the genres in Freebase are very fine-grained. In our data set we had 2085 different genres describing the music style of artists precisely. For instance, for Lady Gaga we had the genres *dance music, electronic music, electronic dance music, eurodance, contemporary R&B, electronica, and pop music.* We consider an overlap in many of these specific genres as a good indication that artists are associated with a common music style. However it would be interesting for our future work to include more feature types such as record release years, associated artists, and events where the artists performed.

Chapter 10

Outlook

In our future work we will examine how the contextualized user interest profiles can be used for context-sensitive recommendations. [Anand et al., 2007] hypothesize that different visits of a user on a website may be associated with different information needs. E. g., a user might seek items according to her profession in one visit and items according to her hobbies in another. The authors argue that the context is reflected in the user's choice of items. For our approach it would be interesting to analyze how items visited in the user's current session match one particular interest topic of the user and how recommendations could be generated accordingly. When providing topic-based resource recommendations, it is not only interesting to recommend items that match a user's most recent topics of interest. Instead also long-term interest topics should be detected and respective recommendations should be provided. In [Schirru, 2010] we proposed a first approach in this direction. The method compared vectors representing a user's interest topics from different interests identification runs by calculating their cosine similarity. Such vectors whose similarity exceeded a predefined threshold were added to a persistent topic trace. For future work it would be interesting to follow this approach and to also experiment with more sophisticated methods, such as topic tracking using online non-negative matrix factorization ([Cao et al., 2007]). So far we have only evaluated the specificity of the cluster labels representing a user's different topics of interest. That way we could assess the likeliness of our system to generate useful on-topic recommendations. A similar approach was previously proposed by [Au Yeung et al., 2008]. However, it would be interesting to test the quality of the resulting recommendations in a live user experiment thus identifying further optimization possibilities for our approach.

Concerning music recommendations we need to integrate the contextualized user profiles with our approach finding similar artists that was presented in [Baumann et al., 2010]. Using artist profiles composed of structured Semantic Web metadata we were able to generate artist recommendations with a high degree of novelty. An integration with our multifaceted user profiles could provide novel artist recommendations taking into account different music styles a user prefers. In the long term it would be interesting to integrate the contextualized user profiles with user model ontologies, e. g., [Heckmann et al., 2005] or [Song et al., 2009], that way targeting context-sensitive music recommendations. For this purpose it would be necessary to learn relations between a user's current situation and her preferred music style.

Part IV

Appendix

Appendix A

Similarity Measures

In this chapter we present similarity and distance measures that are often used for recommender systems or clustering algorithms to determine the similarity between users, items, or between whole item sets. As soon as a representation of two objects in the vector space is found, there are many different ways to calculate the similarity between them. A similarity measure usually assumes values from 0 to 1 where a higher value indicates a higher similarity. In contrast, for distance measures a lower value indicates a higher similarity between the items.

Pearson Correlation Coefficient In collaborative filtering the Pearson correlation coefficient is often used to determine the nearest neighbors of a user [Adomavicius and Tuzhilin, 2005]. Let S_{xy} be the set of items which are co-rated by users x and y, and $r_{x,s}$ be the rating of user x for item s. Let further $\bar{r}_x$ be the average rating of user x. Then the Pearson correlation is determined as follows:

$$sim_{Pearson}\left(x,y\right) = \frac{\sum_{s\in S_{xy}}\left(r_{x,s}-\bar{r}_x\right)\left(r_{y,s}-\bar{r}_y\right)}{\sqrt{\sum_{s\in S_{xy}}\left(r_{x,s}-\bar{r}_x\right)^2\sum_{s\in S_{xy}}\left(r_{y,s}-\bar{r}_y\right)^2}} \tag{A.1}$$

Cosine Similarity The *cosine similarity* measures the similarity between two vectors of n dimensions by finding the angle between them. Given two vectors of attributes, $X = (x_1, x_2, ..., x_n)$ and $Y = (y_1, y_2, ..., y_n)$, the cosine similarity, θ, is represented by using a dot product and magnitude as

$$\cos\left(\theta\right) = \frac{XY}{\|X\|_2\|Y\|_2} = \frac{\sum_{i=1}^{n} x_i \cdot y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}} \tag{A.2}$$

Adjusted Cosine Similarity [Sarwar et al., 2001] introduce the adjusted cosine similarity as a measure to calculate the similarity between items in item-based collaborative filtering (cf. Chapter 2.2.2). Calculating the similarity between items using basic cosine similarity has the drawback that the users' different tendencies to vote are not taken into account, i. e., some users tend to give better ratings, others tend to rate items worse. The adjusted cosine similarity takes care of this drawback by subtracting a user's rating average from each pair of co-rated items. Let $u \in U$ be the set of users that rated both items i and j. Further $r_{u,i}$ is the rating of user u for item i and $\bar{r}_u$ is the average rating of user u. Then the adjusted cosine similarity is calculated as follows:

$$sim_{AdjustedCosine}\left(i,j\right) = \frac{\sum_{u \in U}\left(r_{u,i} - \bar{r}_u\right)\left(r_{u,j} - \bar{r}_u\right)}{\sqrt{\sum_{u \in U}\left(r_{u,i} - \bar{r}_u\right)^2}\sqrt{\sum_{u \in U}\left(r_{u,j} - \bar{r}_u\right)^2}} \tag{A.3}$$

Please note that when calculating the Pearson correlation between items, the average item ratings $\bar{r}_i$ and $\bar{r}_j$ would be subtracted from the user's actual item ratings instead:

$$sim_{PearsonCorrelation}\left(i,j\right) = \frac{\sum_{u \in U}\left(r_{u,i} - \bar{r}_i\right)\left(r_{u,j} - \bar{r}_j\right)}{\sqrt{\sum_{u \in U}\left(r_{u,i} - \bar{r}_i\right)^2}\sqrt{\sum_{u \in U}\left(r_{u,j} - \bar{r}_j\right)^2}} \tag{A.4}$$

Minkowski Metric Geometrical methods measure the distance between two points $X = (x_1, x_2, ..., x_n)$ and $Y = (y_1, y_2, ..., y_n)$ in a vector space. Usually, these metrics have the form of a Minkowski metric:

$$sim_{Minkowski}\left(X,Y\right) = \left[\sum_{i=1}^{n} \mid x_i - y_i \mid^r\right]^{\frac{1}{r}} \; where \; r \geq 1 \tag{A.5}$$

The three most common values for parameter r are:

1. $r = 1$; Manhattan or City-Block distance: $sim_{Manhattan}\left(X,Y\right) = \sum_{i=1}^{n} |x_i - y_i|$

2. $r = 2$; Euclidean distance: $sim_{Euclid}\left(X,Y\right) = \sqrt{\sum_{i=1}^{n}\left(x_i - y_i\right)^2}$

3. $r \to \infty$; Chebyshev distance: $sim_{Chebyshev}\left(X,Y\right) = \max\limits_{1 \leq i \leq n} \mid x_i - y_i \mid = \|x - y\|_\infty$

Association Coefficients The *Jaccard similarity coefficient* is a statistic used for comparing the similarity and diversity of sample sets X and Y:

$$Jaccard(X, Y) = \frac{\mid X \cap Y \mid}{\mid X \cup Y \mid} \tag{A.6}$$

The *Dice coefficient* is a similarity measure related to the Jaccard index. For sets X and Y, the coefficient may be defined as:

$$Dice(X, Y) = \frac{2 \mid X \cap Y \mid}{\mid X \mid + \mid Y \mid} \tag{A.7}$$

Appendix B

Error Measures

In order to evaluate the quality of a clustering result or the accuracy of a recommender system, statistical error measures are often used. In this chapter we present the residual sum of squares in Section B.1 as a popular means to measure the quality of clusterings. In Sections B.2 and B.3 the mean absolute error and the root mean squared error are presented respectively. Both measures are often used to evaluate the predictive accuracy of recommendation algorithms.

B.1 Residual Sum of Squares

The *residual sum of squares (RSS)* can be used to measure the quality of a clustering result. It determines how well the centroids represent the members of their clusters ([Manning et al., 2009], p. 360). It is calculated as the squared distance of each vector from its associated centroid summed over all vectors. For a clustering result with K clusters let $\vec{x} \in \omega_k$ be the feature vector of item x that has been assigned to cluster k. Let $\vec{\mu}\,(\omega_k)$ be the centroid of cluster k. Then the RSS of cluster k is determined as follows:

$$RSS_k = \Sigma_{\vec{x} \in \omega_k} \mid \vec{x} - \vec{\mu}\,(\omega_k) \mid^2 \tag{B.1}$$

The residual sum of squares of the complete clustering result is then determined as

$$RSS = \Sigma_{k=1}^{K} RSS_k \tag{B.2}$$

B.2 Mean Absolute Error

In statistics, the mean absolute error (MAE) measures how close predicted values are to the actual outcomes ([Wikipedia, 2010b]). For recommender systems it measures how close the rating predictions are to the rating values provided by the users ([Herlocker et al., 2004]). Let $p_1, p_2, ..., p_n$ be a set of rating predictions for items $i_1, i_2, ..., i_n$ and let $r_1, r_2, ..., r_n$ be the actual ratings for the items by the respective user. MAE is then calculated as follows:

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |p_i - r_i| \tag{B.3}$$

B.3 Root Mean Squared Error

Similar to MAE the root mean squared error (RMSE) or root mean square deviation (RMSD) measures the differences between predicted values and the values actually observed ([Wikipedia, 2010d]). While MAE stronger punishes a larger amount of errors RMSE penalizes larger errors more. For a set of rating predictions $p_1, p_2, ..., p_n$ for items $i_1, i_2, ..., i_n$ and actual user-item ratings $r_1, r_2, ..., r_n$ RMSE is calculated as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (p_i - r_i)^2}{n}} \tag{B.4}$$

Appendix C

Clustering Algorithms

The goal of clustering algorithms is to group a set of items into subsets that are internally coherent and clearly different from each other ([Manning et al., 2009], p. 349). That means, items in the same cluster should be as similar as possible while items in different clusters should be as dissimilar as possible. Clustering is a form of unsupervised learning. Compared to supervised learning (e. g., classification) the items do not carry any labels assigned by human experts. In information retrieval it is assumed that given an information need of a user the documents in a cluster behave similarly with respect to relevance. [Jardine and van Rijsbergen, 1971] formulated the *cluster hypothesis* as follows:

> *"It is intuitively plausible that the associations between documents convey information about the relevance of documents to requests."*

In our approach we assume that each cluster represents an interest topic of a user or one of her preferred music styles. According to the cluster hypothesis, the resources in a cluster should be relevant for the respective interest topic or music style.

Clustering algorithms can be distinguished into hard and soft clustering algorithms. Hard clustering algorithms assign each item to exactly one cluster. In contrast with soft clustering an item can have a fractional membership in many clusters. Both hard and soft clustering strategies will be depicted in the current chapter. According to the clustering strategy, hierarchical and flat clustering algorithms are distinguished. We describe both approaches following [Manning et al., 2009], pp. 349-401.

The remainder of the chapter is structured as follows: In Section C.1 hierarchical clustering algorithms are presented. As the most representative algorithm for flat clustering, K-Means is depicted in Section C.2 together with a heuristic on how to estimate a reasonable number of clusters. Next, in Section C.3 clustering strategies based on matrix factorization are described. Finally, we present strategies for cluster label extraction in Section C.4.

C.1 Hierarchical Clustering

We present hierarchical clustering algorithms first in this chapter as they generate a structure of clusters that can be easily illustrated in dendrograms and thus support the understanding of clustering algorithms.

In hierarchical clustering bottom-up and top-down approaches are distinguished. Agglomerative (bottom-up) approaches treat each item as a single cluster in the beginning. Then consecutively pairs of clusters are merged until all items are aggregated into one single cluster. Divisive (top-down) approaches need a method to split clusters. They start with one cluster containing all items and then recursively split the clusters until each item ends up in its own cluster. In this section we will focus on hierarchical agglomerative clustering as this approach is more frequently used in information retrieval.

A naïve hierarchical agglomerative clustering algorithm might work as follows: Given a corpus of N documents, compute the $N \times N$ similarity matrix C. In $N-1$ steps the algorithm merges the currently most similar clusters. After each merge update the affected rows and columns in C. The clustering result is stored as a list of merges. Two famous methods to determine the similarity between clusters are single-link and complete-link. With *single-link* the similarity between two clusters is defined as the similarity of the two most similar members in the clusters. In contrast with *complete-link* the similarity between two clusters is defined as the similarity of the two most dissimilar members of the clusters.

An advantage of hierarchical clustering algorithms is that their results can be visualized as dendrograms making the emergence of the clusters easy to comprehend for the user. Cluster merges are represented by horizontal lines. The locations of these lines along the y-coordinate show the similarity of the merged clusters. [Manning et al., 2009] call this similarity the *combination similarity* of two clusters.

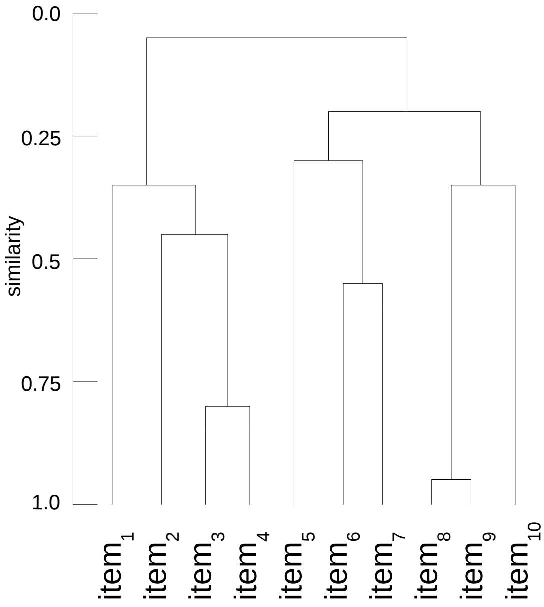


Figure C.1: Example of a dendrogram as output of a hierarchical clustering algorithm.

An example dendrogram is shown in Figure C.1.

With hierarchical clustering the user does not have to specify the number of clusters in advance. In this case a heuristic has to be applied where to cut the hierarchy. The following conditions could be used:

- Utilize a specified similarity threshold, i. e., clusters are merged as long as the combination similarity is higher than that threshold.

- Stop the merging process where the gap between two consecutive combination similarities is largest. Such gaps indicate a reasonable cluster number. The approach can be compared to the elbow criterion as described in Section C.2.2.

- The following formula can be applied: $K = \underset{K'}{\arg\min}\left[RSS\left(K'\right) + \lambda K'\right]$ with K' referring to the cut in the hierarchy resulting in K' clusters and λ is a penalty for additional clusters. RSS is the residual sum of squares as defined in Appendix B.1.

However, it is also possible to specify the desired number of clusters in advance and then stop the merging process when this number is reached.

Both single-link and complete-link clustering base each merging decision only on one pair of items (i. e., the most similar items with single-link and the most dissimilar items with complete-link clustering) hence not considering the distribution of items in the clusters. For that reason both algorithms often produce results that are undesirable. Single-link is known to suffer from the problem of chaining (i. e., producing long, straggly clusters). Complete-link on the other hand generates compact clusters with small diameters but it pays too much attention to outliers that way sometimes missing intuitive cluster structures. Methods have been developed that avoid the drawbacks of single-link and complete-link clustering. For instance group-average agglomerative clustering determines the cluster quality based on the similarity of all documents in the two clusters under consideration that way avoiding the issues of single-link and complete-link clustering mentioned before.

C.2 Flat Clustering Using K-Means

The current section first describes K-Means, the most famous flat clustering algorithm. As K-Means requires the number of clusters as input parameter we depict a heuristic to estimate a reasonable number of clusters subsequently.

C.2.1 The K-Means Algorithm

K-Means is often referred to as the most important flat clustering algorithm. Its goal is to minimize the residual sum of squares (RSS, cf. Appendix B.1) of items from the cluster centers. Given that each item $\vec{x}$ is represented as a length-normalized vector in a real valued space, the center (mean or centroid $\vec{\mu}$) of a cluster ω is defined as follows:

$$\vec{\mu}\,(\omega) = \frac{1}{\mid \omega \mid} \sum_{\vec{x} \in \omega} \vec{x} \tag{C.1}$$

The algorithm works as follows:

- Select k random documents as cluster centers (i. e., the seed).
- Repeat the following steps until a stopping criterion is met:
  - (Re)assign each document to the cluster with the closest centroid.
  - Recompute the centroids based on the current cluster members.

[Manning et al., 2009] list the following termination criteria:

1. A predefined number of iterations has been reached. This criterion limits the runtime of the algorithm, however it might lead to poor results when the number of iterations is insufficient.
2. The assignment of items to clusters remains stable between iterations. This condition may produce good clusterings except for cases where the algorithm is stuck in a bad local minimum. For this criterion the runtime might be unacceptable. An equivalent stopping condition is to check whether the centroids do not change between iterations.
3. Stop as soon as RSS is under a predefined threshold. This condition guarantees a clustering result of a certain quality. However, in practice it has to be combined with a limit on the number of iterations in order to make sure that the algorithm terminates.
4. Stop when the decrease in RSS falls below a certain threshold. For small thresholds this criterion indicates that the result is close to convergence but again this condition needs to be combined with a limit on the number of iterations in order to avoid long runtimes.

Compared to hierarchical clustering algorithms, flat clustering algorithms have some drawbacks. The results of flat clustering miss a structure that can be easily visualized like the dendrograms that are obtained with hierarchical clustering. Further the results of flat clustering are nondeterministic and the algorithms require a predefined number of clusters as input parameter. Most hierarchical clustering algorithms used in information retrieval are by contrast deterministic and there exist good heuristics to achieve a clustering result with a reasonable number of clusters without requiring the user to specify the cluster number in advance. A major advantage of flat clustering strategies however is their efficiency. While the most common hierarchical algorithms have at least quadratic complexity in the number of items, the runtime of K-Means is linear in all relevant factors: the number of iterations, clusters, vectors and dimensionality of the space.

C.2.2 Estimating the Cluster Number

We estimate the number of clusters in the data set as described in [Manning et al., 2009], p. 365. First we define a range in which we expect to find

#items/artists	**reasonable cluster number**
$i \leq 250$	1^{st}
$250 < i \leq 300$	2^{nd}
$300 < i \leq 350$	3^{rd}
$350 < i \leq 400$	4^{th}
$i > 400$	5^{th}

Table C.1: Reasonable cluster number used depending on the number of a user's preferred items/artists.

the number of interest topics or preferred music styles. We chose a range between 2 and 20 for our experiments, however the boarders are configurable in our algorithm. For each potential cluster size k $(2 \leq k \leq 20)$ we run K-Means i-times (we chose $i = 10$), each time with a different initialization. We compute for each clustering result the residual sum of squares and the minimum RSS over all i clusterings (denoted by $\widehat{RSS}_{min}(k)$). Then we take a look at the values $\widehat{RSS}_{min}(k)$ and search for the points where successive decreases in $\widehat{RSS}_{min}$ become significantly smaller.[1] When plotting the quality of the clustering results (in our case the values $\widehat{RSS}_{min}(k)$) against the number of clusters in a graph, this process is often referred to as searching the elbows in the curve. The first five such values $k-1$ are stored as reasonable cluster sizes. We store five values in order to enable clusterings according to different granularities. If broad clustering granularity is desired we take the first reasonable number of clusters, for middle granularity the second, and so on. Table C.1 shows the heuristic used to determine which reasonable cluster number should be used. The underlying assumption is that users with more preferred items/artists have more interest topics/preferred music styles hence the clustering granularity is increased with the number of preferred items/artists.

C.3 Clustering via Matrix Factorization

Aside from traditional hierarchical and flat clustering techniques document clustering based on matrix factorization has become popular in recent years. It proves to be particularly useful for co-clustering (i. e., simultaneous clustering of the rows

[1] $RSS_{min}(k)$ is a monotonically decreasing function in k with minimum 0 for $k = N$ with N being the number of documents.

and columns of a matrix) where documents have to be clustered and descriptive labels need to be extracted for each cluster. We depict in Section C.3.1 a clustering approach in the latent semantic space derived by singular value decomposition on the term-document matrix. Document clustering based on non-negative matrix factorization is described in Section C.3.2.

C.3.1 LSA-based Clustering

[Song and Park, 2007] propose a document model based on latent semantic analysis (LSA) for text clustering. LSA applies singular value decomposition on the term-document matrix representing the document corpus, thus obtaining k orthogonal factors. The reduced space is intended to better capture the relations between documents. In this semantic structure two documents can be related even if they do not share any common words. The authors state that clustering performed in the latent semantic space leads to better results than clustering performed using the original vector space model.

Given a term-document matrix A representing a document corpus, the singular value decomposition of A is given by:

$$A = U \Sigma V^T \tag{C.2}$$

with U being the matrix of the left singular vectors (matrix of term vectors) and V being the matrix of the right singular vectors (matrix of document vectors). Σ is the diagonal matrix consisting of singular values. With LSA A is approximated with a rank-k matrix:

$$A_k = U_k \Sigma_k V_k^T \tag{C.3}$$

with U_k being comprised of the first k columns of U, V_k^T comprising the first k rows of V^T and $\Sigma_k = diag\left(\sigma_1, \sigma_2, ..., \sigma_k\right)$ being the first k factors.

In order to represent a document d in the latent semantic space each document is firstly initialized as $m \times 1$ matrix with m being the total number of terms in the corpus. As mentioned before U represents the matrix of term vectors in all documents and U_k spans the basis vectors of U. Now matrices d^T and U_k are multiplied to represent the document vector hence obtaining for each document a

$1 \times k$ matrix.

$$\hat{d} = d^T U_k \tag{C.4}$$

The corpus is then organized as

$$D' = DU_k \tag{C.5}$$

with D being the $n \times m$ document-term matrix. To use the LSA-based document representation, each document d is constructed as a row vector of D'. For clustering the K-Means algorithm is used. To initialize the algorithm the dimensions of D' are reduced from n to k ($n < k$) and the documents are constructed in the reduced latent semantic space. Then K-Means is run as described in Section C.2.1.

In their evaluation experiments the authors show that document representations based on LSA clearly outperform traditional vector space model representations in terms of clustering quality.

C.3.2 NMF-based Clustering

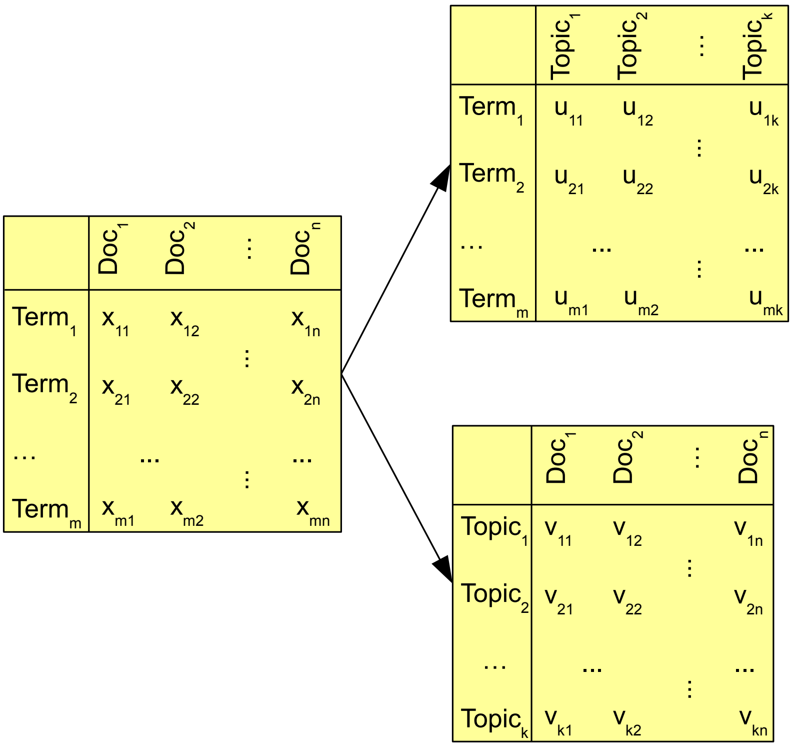
Using non-negative matrix factorization (NMF) for (soft) document clustering was first introduced in [Xu et al., 2003]. The authors show that NMF-based document clustering is able to surpass latent semantic indexing and spectral clustering based approaches.

NMF finds the positive factorization of a given positive matrix. It is applied to the term-document matrix representation of a document corpus. In the latent semantic space which is derived by applying NMF, each axis represents the base topic of a document cluster. Every document is represented as an additive combination of these base topics. Associating a document with a cluster is done by choosing the base topic (axis) that has the highest projection value with the document. We apply NMF as follows:

Let $W = \{f_1, f_2, ..., f_m\}$ be the set of terms in the document corpus after our preprocessing steps. The weighted term vector X_d of document d is defined as

$$X_d = [x_{1d}, x_{2d}, ..., x_{md}]^T \tag{C.6}$$

with x_{id} being the TF-IDF weight of the term f_i in document d.



	Doc$_1$	Doc$_2$	⋮	Doc$_n$
Term$_1$	x_{11}	x_{12}		x_{1n}
			⋮	
Term$_2$	x_{21}	x_{22}		x_{2n}
…	…			…
			⋮	
Term$_m$	x_{m1}	x_{m2}		x_{mn}

	Topic$_1$	Topic$_2$	⋮	Topic$_k$
Term$_1$	u_{11}	u_{12}		u_{1k}
			⋮	
Term$_2$	u_{21}	u_{22}		u_{2k}
…		…		…
			⋮	
Term$_m$	u_{m1}	u_{m2}		u_{mk}

	Doc$_1$	Doc$_2$	⋮	Doc$_n$
Topic$_1$	v_{11}	v_{12}		v_{1n}
			⋮	
Topic$_2$	v_{21}	v_{22}		v_{2n}
…		…		…
			⋮	
Topic$_k$	v_{k1}	v_{k2}		v_{kn}

Figure C.2: Factorization of the term-document matrix by the NMF algorithm.

We assume that our document corpus consists of k clusters. The goal of NMF is to factorize X into non-negative matrices U $(m \times k)$ and V^T $(k \times n)$ which minimize the following objective function:

$$J = \frac{1}{2} \parallel X - UV^T \parallel \tag{C.7}$$

$\parallel \cdot \parallel$ denotes the squared sum of all the elements in the matrix.

Each element u_{ij} of matrix U determines the degree to which the associated term f_i belongs to cluster j. For cluster labeling we simply choose for each cluster the ten terms with the highest degree of affiliation. Analogously each element v_{ij} of matrix V represents the degree to which document i is associated with cluster j. To cluster the documents, again we assign every document to the cluster with the highest degree of affiliation. If a document d clearly belongs to one cluster x then v_{dx} will have a high value compared to the rest of the values in the d'th row vector of V. The matrix factorization is depicted in Figure C.2.

For those clustering approaches that do not extract cluster labels at the same time, labels can be obtained by applying cluster labeling methods. Two well-known approaches will be presented in the next section.

C.4 Cluster Labeling Methods

When data has been clustered it is often desirable to have a description of the clusters obtained (e. g., when users interact with the clusters). Our proposed recommendation approaches use cluster labels consisting of representative terms or features in order to enable the retrieval of resources according to a user's respective topics of interest or preferred music styles. Cluster labels can be obtained by applying cluster labeling methods. [Manning et al., 2009], pp. 396, distinguish two approaches: First, there are *cluster-internal* methods that extract a label solely based on the cluster under consideration. Frequency-based cluster labeling is one such method that will be described in Section C.4.1. Second, with *differential cluster labeling* the labels are extracted by comparing the distribution of features in one cluster with the distribution of the features in the other clusters. We will present the chi-square method as one example of such an approach in Section C.4.2. For our topic-based resource recommendations approach we have used the NMF algorithm for clustering. As NMF is a co-clustering approach that extracts clusters and cluster labels at the same time, no separate cluster labeling method had to be applied with this algorithm. In Section C.3.2 it has been described how the matrices obtained by the NMF algorithm were used for clustering and cluster labeling respectively.

C.4.1 Frequency-based Cluster Labeling

Frequency-based cluster labeling selects the features that are most common in a cluster. Usually document frequency (the number of documents in the cluster that have the feature) and collection frequency (how often does the feature appear in the whole collection) are distinguished. For our purposes we use a slightly different approach:

Let I be our set of items and let $c = \{i_1, i_2, ..., i_j\}$ be the items in cluster c. Each item $i = (w_{i_1}, w_{i_2}, ..., w_{i_n})$ is represented by n weighted features. The weight of feature f in cluster c is then determined as follows:

$$FB\left(c, f\right) = \sum_{i \in c} w_{i_f} \tag{C.8}$$

To obtain a label for cluster c the features are ranked by their weight and the top k features are selected.

C.4.2 Chi-square Cluster Labeling

In statistics the chi-square test is used to check whether two events are independent. Two events A and B are defined to be independent if $P(AB) = P(A)P(B)$. An equivalent definition is: $P(A \mid B) = P(A)$ and $P(B \mid A) = P(B)$.

Let $e_f \in \{0,1\}$ be a variable that indicates whether a feature is present ($e_f = 1$) or absent ($e_f = 0$) in the profile of an item and $e_c \in \{0,1\}$ a variable that indicates whether an item is in the currently considered cluster ($e_c = 1$) or not ($e_c = 0$). Let further N be the observed frequency in our item set I and E be the expected frequency, e. g., E_{11} is the expected frequency that a feature and a cluster occur together given that feature and cluster are independent. The chi-square weight of a feature f for a cluster c is then calculated as follows:

$$\chi^2(I, f, c) = \sum_{e_f \in \{0,1\}} \sum_{e_c \in \{0,1\}} \frac{\left(N_{e_f e_c} - E_{e_f e_c}\right)^2}{E_{e_f e_c}} \tag{C.9}$$

To obtain a label for cluster c the features are ranked by their weight and the top k features are selected.

Appendix D

Music Mood Categories

[Laurier et al., 2009] derived four mood categories from the Last.fm folksonomy. Table D.1 shows the first 15 tags for each category.

Cluster 1	**Cluster 2**	**Cluster 3**	**Cluster 4**
angry	sad	tender	happy
aggressive	bittersweet	soothing	joyous
visceral	sentimental	sleepy	bright
rousing	tragic	tranquil	cheerful
intense	depressing	good natured	happiness
confident	sadness	quiet	humorous
anger	spooky	calm	gay
exciting	gloomy	serene	amiable
martial	sweet	relax	merry
tense	mysterious	dreamy	rollicking
anxious	mournful	delicate	campy
passionate	poignant	longing	light
quirky	lyrical	spiritual	silly
wry	miserable	wistful	boisterous
fiery	yearning	relaxed	fun

Table D.1: Music mood clusters proposed by [Laurier et al., 2009].

[Hu et al., 2009a] proposed the following list of 18 music mood categories consisting of social tags from Last.fm ([Hu et al., 2009b]):

- calm, comfort, quiet, serene, mellow, chill out, calm down, calming, chillout, comforting, content, cool down, mellow music, mellow rock, peace of mind, quietness, relaxation, serenity, solace, soothe, soothing, still, tranquil, tranquility, tranquillity

- sad, sadness, unhappy, melancholic, melancholy, feeling sad, mood: sad - slightly, sad song
- happy, happiness, happy songs, happy music, glad, mood: happy
- romantic, romantic music
- upbeat, gleeful, high spirits, zest, enthusiastic, buoyancy, elation, mood: upbeat
- depressed, blue, dark, depressive, dreary, gloom, darkness, depress, depression, depressing, gloomy
- anger, angry, choleric, fury, outraged, rage, angry music
- grief, heartbreak, mournful, sorrow, sorry, doleful, heartache, heartbreaking, heartsick, lachrymose, mourning, plaintive, regret, sorrowful
- dreamy
- cheerful, cheer up, festive, jolly, jovial, merry, cheer, cheering, cheery, get happy, rejoice, songs that are cheerful, sunny
- brooding, contemplative, meditative, reflective, broody, pensive, pondering, wistful
- aggression, aggressive
- confident, encouraging, encouragement, optimism, optimistic
- angst, anxiety, anxious, jumpy, nervous, angsty
- earnest, heartfelt
- desire, hope, hopeful, mood: hopeful
- pessimism, cynical, pessimistic, weltschmerz, cynical/sarcastic
- excitement, exciting, exhilarating, thrill, ardor, stimulating, thrilling, titillating

Appendix E

Evaluation

The current chapter of the appendix summarizes important aspects for the evaluation of recommender systems and in particular for the evaluation of collaborative filtering systems as discussed by [Herlocker et al., 2004]. First, live user experiments are compared to offline analyses in Section E.1. Next, in Section E.2 the use of synthesized in contrast to natural data sets is discussed. Properties of recommender system data sets are depicted in Section E.3. The chapter concludes with an overview of different measures for the evaluation of recommender systems in Section E.4.

E.1 Kind of Evaluation

Recommender systems can be evaluated by conducting *live user experiments*, *offline analyses*, or both. The approach taken by most researchers in the past was the offline analysis of the predictive accuracy. Offline analyses can be performed quickly and economically for different algorithms and even on different data sets. However, there are two major weaknesses of offline analyses: First, recommendations of items for which no ratings of the active user are available cannot be evaluated. Due to the sparsity of most rating data sets this point is a major issue. Second, with offline analysis it cannot be evaluated whether users preferred a system because of its predictions or because of other criteria like, e. g., the aesthetics of the user interface. Live user experiments on the other hand allow the evaluation of measures such as user performance, satisfaction, and participation. Dimensions of user evaluations are depicted at the end of Section E.5.

E.2 Kind of Data Set

When selecting a data set on which to evaluate a recommender algorithm researchers have to decide whether to use a *natural data set* (i. e., using an existing data base) or a *synthesized data set* (i. e., an artificially constructed data set). Using a natural data set may have the drawback that the data matches the target domain and task only imperfectly. A synthesized data set on the other hand may better fit these properties but is therefore also often criticized to be unfair as it might match the tested approach to well. Using synthesized data sets to test obvious flaws in an early stage of the recommender development is often reasonable however comparing algorithms based on such data is risky as the data might fit one algorithm better than the other.

E.3 Properties of Data Sets

Subsequently we will briefly present properties of recommender system data sets as proposed by [Herlocker et al., 2004] in tabular form (see Table E.1). The properties are divided into *domain features* (reflecting the nature of the content that is recommended), *inherent features* (reflecting the nature of the respective recommender system), and *sample features* summarizing the distribution of the attributes in the data set. When selecting a foreign data set for the evaluation of the own recommendation approach it is particularly important to carefully compare these properties in order to find out which one is suited best.

E.4 Evaluation Measures

Subsequently we will present four different kinds of accuracy metrics that can be used for the offline evaluation of recommender systems.

E.4.1 Predictive Accuracy Metrics

Predictive accuracy metrics are widely used to evaluate the performance of recommender systems. They determine how close a rating prediction is to the true user rating. Usually predictive accuracy is measured by taking a data base of user-item ratings, withholding a certain percentage of the ratings, and calculating a rating

Domain Features	- the topic of the content and the context in which the recommendations are provided - the users' tasks that are supported by the recommender - the need for novel and high quality recommendations - cost/benefit ratio of false/true positives/negatives - the granularity of the true user preferences (opposed to the granularity in which the user preferences may be expressed in the platform)
Inherent Features	- kind of ratings (explicit, implicit, or both) - the rating scale - the rating dimensions - availability of timestamps for ratings - tracking of recommendations (yes/no) - availability of demographic information about users or item content information - the biases involved in collecting the data
Sample Features	- the density of the ratings set - the number or density of the ratings from those users for whom recommendations are being made - general size and distribution properties of the data set

Table E.1: Properties of data sets ([Herlocker et al., 2004]).

	Chosen	**Not Chosen**	**Total**
Relevant	N_{rc}	N_{rn}	N_r
Irrelevant	N_{ic}	N_{in}	N_i
Total	N_c	N_n	

Table E.2: Confusion matrix showing recommended (chosen) items in contrast to the user's information need.

prediction for the withheld items. The deviation between the predicted and the actual user ratings can than be calculated by determining, e. g., the mean absolute error (cf. Appendix B.2) or the root mean squared error (cf. Appendix B.3).

Predictive accuracy metrics have often been criticized as being insufficient ([McNee et al., 2006b]) as they only evaluate what is already known in the system. Recommendations of items that have not been rated by the user cannot be assessed that way. Predictive accuracy metrics seem to be well suited for tasks such as *annotation in context*. They might be less appropriate for tasks like *find good items* where only the top items of a ranked list are presented to the user.

E.4.2 Classification Accuracy Metrics

Classification accuracy metrics measure the amount of correct and incorrect recommendations with respect to a user's information need. The results are often illustrated as a confusion matrix (see Table E.2). These metrics are well suited for tasks such as *find good items* where items in a recommendation list are either relevant to the user or not. Classification accuracy metrics suffer the same sparsity problems as predictive accuracy measures. Assessing recommended items that are not rated by the user is not possible. One approach to tackle this problem is to remove unrated items from the recommendations. However evaluating recommendation lists that are actually shown to the user is not possible that way.

Precision and recall are the most widely used metrics to evaluate information retrieval systems and they have also been used as classification accuracy metrics for the evaluation of recommender systems. In this work these measures have been introduced together with the F-measure in Chapter 7.3.2. They were used to evaluate how well our cluster labels enabled the retrieval of relevant on-topic resources in the use case of topic-based resource recommendations.

E.4.3 Rank Accuracy Metrics

Rank accuracy metrics determine how good a predicted ordering of items matches the user's ordering of the items. These measures are well suited in domains where a ranked list of items is presented to the user and the true user preferences are non binary. The rank accuracy metrics may be too sensitive in domains where the user just seeks items that are "good enough". However when good alternatives are not sufficient and the best items are required these measures can be helpful.

E.4.4 Prediction-Rating Correlation

We consider two variables as correlated when the variance in one variable can be explained by the variance in the other variable. Frequently used correlation measures are the Pearson correlation coefficient and Spearman's ρ. Pearson correlation is determined as depicted in Equation A.1. Spearman's ρ is a rank correlation measure. It is calculated in the same manner as the Pearson correlation coefficient however instead of ratings the ranks of the items are used. It has to be noted that the Spearman correlation metric doesn't handle partial orderings well. If the user's ranking of items given by the rating values is a partial ordering (e. g., many items rated with four stars with no further distinction) and the system uses a complete ordering, the measure will penalize every pair of items rated equivalently by the user and ranked differently by the system. Measures such as the normalized distance-based performance measure (NDPM, [Yao, 1995]) are supposed to remedy this problem. Despite of their simplicity, correlation based metrics have not been used very often to evaluate recommender or information retrieval systems.

E.5 Beyond Accuracy

As already mentioned before there is an increasing awareness that judging recommender systems based on accuracy alone does not necessarily lead to effective and satisfying user experiences with a system. Recommending very popular items can lead to high accuracy but may still not provide much utility to the user. E. g., a recommender system for a supermarket suggesting bread, eggs, and bananas will almost always be highly accurate but still doesn't aid the user. In the current section we will present measures for the evaluation of recommender systems that go beyond accuracy targeting at user utility ([Herlocker et al., 2004]).

Coverage is a measure of the domain of items that can be recommended by the system. A straightforward definition is coverage as the *percentage of items for which rating predictions can be generated* (prediction coverage). Alternatively one could define coverage as the *percentage of catalog items that is actually recommended to users* (catalog coverage). Coverage is particularly important for the *find all good items* task as well as for the *annotation in context* task. Much in the same way that precision should always be evaluated together with recall, coverage should always be assessed together with accuracy. Tuning a system for coverage at the cost of providing false rating predictions/recommendations is obviously not desirable.

The Learning Rate measures the amount of ratings needed for a collaborative filtering system to provide useful or "acceptable" recommendations and to improve the recommendation quality. Three types of learning rates are distinguished: (i) the *overall learning rate* determines the quality of the recommendations as a function of the total number of ratings (or users) in the system. (ii) The *per-item learning rate* indicates the quality of the predictions for an item as a function of the number of available ratings for the item. (iii) Analogously the *per-user learning rate* determines the quality of the recommendations for a user as a function of her provided ratings.

Novelty and Serendipity Recommender systems tuned for accuracy often suffer the problem of recommending too obvious items. This has two major drawbacks: First, it is very likely that the user already knows about these items. In this case she either has already consumed/purchased the item or has made a conscious decision not to consume/purchase it. Second, store managers already know which items are popular and make arrangements to present these products in an advantageous way. Hence new dimensions such as novelty and serendipity are often requested as measures to evaluate recommender systems. Novelty refers to recommendations of items that are new to the user but may potentially be similar to previously seen items (e. g., films by the same director in the domain of movies). Serendipitous recommendations on the other hand are both attractive and surprising to the user (i. e., not similar to previously seen items). However it should be noted that obvious recommendations can aid new users to build up trust with a recommender system and might therefore be helpful when a user starts to use a system.

Confidence is a measure of how sure the recommender system is about the accuracy of its recommendation. It is often measured based on the amount of data that is available to recommend an item. Confidence should clearly be distinguished from the strength of a recommendation which measures how much the system thinks the user will like a recommended item (e. g., a predicted rating of four on a five point rating scale). An item with a rating prediction of five stars will not necessarily be preferred by a user over an item with a rating prediction of four stars when the confidence for the five star prediction is low (i. e., the prediction is calculated based on few data points).

User Evaluation constitutes an important instrument that allows to directly measure the users' reactions to a recommender system. Subsequently we will present four dimensions as proposed by [Herlocker et al., 2004] for user evaluations:

Explicit vs. Implicit: In explicit evaluations users are directly asked about their reactions to a recommender system (e. g., via surveys and interviews). Implicit user evaluations typically log user behavior and then perform analyses on the log data.

Laboratory Studies vs. Field Studies: Laboratory studies are particularly suited to test well-defined hypotheses under controlled conditions. Field studies on the other hand allow to monitor users in their own actual contexts that way unveiling usage patterns, issues, and user needs that are unmet.

Outcome vs. Process: Measuring the outcome of a task such as *find good items* is important for the evaluation of recommender systems. However, it is crucial to also take into account the process, i. e., considering how efficiently (in terms of time and effort spent) the task could be solved by using a particular system.

Short-term vs. Long-term: Some issues users may have with a system may not be discovered in short-term studies. E. g., a short-term study may show that users still find useful items even if the overall quality of a recommender system is rather low. However in the long-term users might get dissatisfied with such a system and decide to stop using it.

Bibliography

[Adomavicius and Tuzhilin, 2005] Adomavicius, G. and Tuzhilin, A. (2005). 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.

[Agne et al., 2006] Agne, S., Reuschling, C., and Dengel, A. (2006). DynaQ - dynamic queries for electronic document management. In *Proceedings IEEE-EDM.*, pages 56–59. IEEE International Workshop on the Electronic Document Management in an Enterprise Computing Environment. Hong Kong, China.

[Allan et al., 1998] Allan, J., Carbonell, J., Doddington, G., Yamron, J., and Yang, Y. (1998). Topic detection and tracking pilot study: Final report. In *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, pages 194–218.

[Amatriain, 2009] Amatriain, X. (2009). The netflix prize: Lessons learned. `http://technocalifornia.blogspot.com/2009/09/netflix-prize-lessons-learned.html`. [Online; accessed 19-November-2010].

[Anand et al., 2007] Anand, S. S., Kearney, P., and Shapcott, M. (2007). Generating semantically enriched user profiles for web personalization. *ACM Trans. Internet Technol.*, 7(4):22.

[Ankolekar et al., 2007] Ankolekar, A., Krötzsch, M., Tran, T., and Vrandecic, D. (2007). The two cultures: mashing up web 2.0 and the semantic web. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 825–834, New York, NY, USA. ACM.

[Au Yeung et al., 2008] Au Yeung, C. M., Gibbins, N., and Shadbolt, N. (2008). A study of user profile generation from folksonomies. In *Proceedings of the Workshop on Social Web and Knowledge Management (SWKM2008), co-located with WWW2008, Beijing, China, 21-25 April, 2007*, pages 1–8.

[Balabanovic and Shoham, 1997] Balabanovic, M. and Shoham, Y. (1997). Fab: Content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72.

[Baumann et al., 2010] Baumann, S., Schirru, R., and Streit, B. (2010). Towards a storytelling approach for novel artist recommendations. In Detyniecki, M., Knees, P., Nürnberger, A., Schedl, M., and Stober, S., editors, *Adaptive Multimedia Retrieval*, volume 6817 of *Lecture Notes in Computer Science*, pages 1–15. Springer.

[Begelman et al., 2006] Begelman, G., Keller, P., and Smadja, F. (2006). Automated tag clustering: Improving search and exploration in the tag space. In *Collaborative Web Tagging Workshop at WWW2006, Edinburgh, Scotland.*

[Belkin and Croft, 1992] Belkin, N. J. and Croft, W. B. (1992). Information filtering and information retrieval: two sides of the same coin? *Communications of the ACM*, 35(12):29–38.

[Berners-Lee, 2009] Berners-Lee, T. (2009). Linked data. `http://www.w3.org/DesignIssues/LinkedData`. [Online; accessed 23-September-2009].

[Berners-Lee and Fischetti, 2000] Berners-Lee, T. and Fischetti, M. (2000). *Weaving the Web : The Original Design and Ultimate Destiny of the World Wide Web*. Harper Paperbacks.

[Berners-Lee et al., 2001] Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. *Scientific American*, May Issue:34–43.

[Bizer et al., 2009] Bizer, C., Heath, T., and Berners-Lee, T. (2009). Linked data - the story so far. *International Journal on Semantic Web and Information Systems*, 5(3):1–22.

[Blei et al., 2003] Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

[Bradley and Smyth, 2001] Bradley, K. and Smyth, B. (2001). Improving recommendation diversity. In O'Donoghue, D., editor, *Proceedings of the Twelfth National Conference in Articial Intelligence and Cognitive Science (AICS-01)*, pages 75–84.

[Burke, 2000] Burke, R. (2000). Knowledge-based recommender systems. *Encyclopedia of Library and Information Science*, 69(32).

[Burke, 2002] Burke, R. D. (2002). Hybrid recommender systems: Survey and experiments. *User Model. User-Adapt. Interact.*, 12(4):331–370.

[Cao et al., 2007] Cao, B., Shen, D., Sun, J.-T., Wang, X., Yang, Q., and Chen, Z. (2007). Detect and track latent factors with online nonnegative matrix factorization. In *IJCAI'07: Proceedings of the 20th international joint conference on Artificial intelligence*, pages 2689–2694, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Celma, 2008] Celma, Ó. (2008). *Music Recommendation and Discovery in the Long Tail*. PhD thesis, University Pompeu Fabra, Barcelona, Spain.

[Celma and Herrera, 2008] Celma, O. and Herrera, P. (2008). A new approach to evaluating novel recommendations. In *RecSys '08: Proceedings of the 2008 ACM conference on Recommender systems*, pages 179–186, New York, NY, USA. ACM.

[Cyganiak and Jentzsch, 2011] Cyganiak, R. and Jentzsch, A. (2011). The linking open data cloud diagram. `http://richard.cyganiak.de/2007/10/lod/`. [Online; accessed 16-May-2012].

[Deerwester et al., 1990] Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., and Harshman, R. A. (1990). Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407.

[Downes, 2003] Downes, S. (2003). Resource profiles. `http://www.downes.ca/files/resource_profiles.htm`. [Online; accessed 24-June-2010].

[Duan et al., 2009] Duan, M., Ulges, A., Breuel, T. M., and Wu, X.-q. (2009). Style modeling for tagging personal photo collections. In *CIVR '09: Proceeding of the ACM International Conference on Image and Video Retrieval*, pages 1–8, New York, NY, USA. ACM.

[Goldberg et al., 1992] Goldberg, D., Nichols, D., Oki, B. M., and Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 35(12):61–70.

[Golder and Huberman, 2006] Golder, S. and Huberman, B. A. (2006). Usage patterns of collaborative tagging systems. *Journal of Information Science*, 32(2):198–208.

[Greaves and Mika, 2008] Greaves, M. and Mika, P. (2008). Editorial: Semantic web and web 2.0. *Web Semant.*, 6:1–3.

[Gruber, 1993] Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220.

[Guo and Joshi, 2010] Guo, Y. and Joshi, J. B. (2010). Topic-based personalized recommendation for collaborative tagging system. In *HT '10: Proceedings of the 21st ACM conference on Hypertext and hypermedia*, pages 61–66, New York, NY, USA. ACM.

[Heckmann et al., 2005] Heckmann, D., Schwartz, T., Brandherm, B., Schmitz, M., and von Wilamowitz-Moellendorff, M. (2005). Gumo - the general user model ontology. In *Proceedings of the 10th International Conference on User Modeling*, pages 428–432, Edinburgh, UK. LNAI 3538: Springer, Berlin Heidelberg.

[Herlocker et al., 2004] Herlocker, J. L., Konstan, J. A., Terveen, L. G., and Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53.

[Hevner, 1936] Hevner, K. (1936). Experimental studies of the elements of expression in music. *The American Journal of Psychology*, 48(2):246–268.

[Hofmann, 1999] Hofmann, T. (1999). Probabilistic latent semantic analysis. In *Proc. of Uncertainty in Artificial Intelligence, UAI'99*, Stockholm.

[Hotho et al., 2006a] Hotho, A., Jäschke, R., Schmitz, C., and Stumme, G. (2006a). Bibsonomy: A social bookmark and publication sharing system. In *Proceedings of the Conceptual Structures Tool Interoperability Workshop at the 14th International Conference on Conceptual Structures*, pages 87–102. Aalborg University Press.

[Hotho et al., 2006b] Hotho, A., Jäschke, R., Schmitz, C., and Stumme, G. (2006b). Information retrieval in folksonomies: Search and ranking. In *The Semantic Web: Research and Applications*, volume 4011 of *Lecture Notes in Computer Science*, pages 411–426, Heidelberg. Springer.

[Hu and Pu, 2009] Hu, R. and Pu, P. (2009). A comparative user study on rating vs. personality quiz based preference elicitation methods. In *Proceedings of the 14th international conference on Intelligent user interfaces*, IUI '09, pages 367–372, New York, NY, USA. ACM.

[Hu et al., 2007] Hu, X., Bay, M., and Downie, J. S. (2007). Creating a simplified music mood classification ground-truth set. In *Proceedings of the 8th International Conference on Music Information Retrieval*, pages 309–310.

[Hu and Downie, 2007] Hu, X. and Downie, J. S. (2007). Exploring mood metadata: relationships with genre, artist and usage metadata. In *Proc. of ISMIR 2007*, pages 67–72.

[Hu et al., 2009a] Hu, X., Downie, J. S., and Ehmann, A. F. (2009a). Lyric text mining in music mood classification. In *Proceedings of the 10th International Conference on Music Information Retrieval*, pages 411–416.

[Hu et al., 2009b] Hu, X., Downie, J. S., and Ehmann, A. F. (2009b). Mood categories and song distributions. `http://music-ir.org/archive/figs/18moodcat.htm`. [Online; accessed 31-August-2010].

[Hu et al., 2008] Hu, X., Downie, J. S., Laurier, C., Bay, M., and Ehmann, A. F. (2008). The 2007 mirex audio mood classification task: Lessons learned. In Bello, J. P., Chew, E., and Turnbull, D., editors, *ISMIR*, pages 462–467.

[Jardine and van Rijsbergen, 1971] Jardine, N. and van Rijsbergen, C. J. (1971). The use of hierarchic clustering in information retrieval. *Information Storage and Retrieval*, 7(5):217–240.

[Juslin and Sloboda, 2001] Juslin, P. N. and Sloboda, J. A. (2001). *Music and Emotion: Theory and Research.* Oxford University Press, USA.

[Konstan et al., 1997] Konstan, J., Miller, B., Maltz, D., Herlocker, J., Gordon, L., and Riedl, J. (1997). Grouplens: Applying collaborative filtering to usenet news. *Communications of the ACM*, 40:77–87.

[Koren, 2009] Koren, Y. (2009). The bellkor solution to the netflix prize. `http://www.netflixprize.com/assets/GrandPrize2009_BPC_BellKor.pdf`. [Online; accessed 19-November-2010].

[Krötzsch et al., 2006] Krötzsch, M., Vrandečić, D., and Völkel, M. (2006). Semantic mediawiki. In Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., and Aroyo, L., editors, *The Semantic Web - ISWC 2006*, volume 4273 of *Lecture Notes in Computer Science*, pages 935–942. Springer Berlin / Heidelberg.

[Laurier et al., 2009] Laurier, C., Sordo, M., Serrà, J., and Herrera, P. (2009). Music mood representations from social tags. In *International Society for Music Information Retrieval (ISMIR) Conference*, pages 381–386.

[Lee and Lee, 2006] Lee, J. and Lee, J. (2006). Music for my mood: A music recommendation system based on context reasoning. In Havinga, P., Lijding, M., Meratnia, N., and Wegdam, M., editors, *Smart Sensing and Context*, volume 4272 of *Lecture Notes in Computer Science*, pages 190–203. Springer Berlin / Heidelberg.

[Li et al., 2008] Li, X., Guo, L., and Zhao, Y. E. (2008). Tag-based social interest discovery. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 675–684, New York, NY, USA. ACM.

[Linden et al., 2003] Linden, G., Smith, B., and York, J. (2003). Amazon.com recommendations: item-to-item collaborative filtering. *Internet Computing, IEEE*, 7(1):76–80.

[Liu and Huang, 2000] Liu, Z. and Huang, Q. (2000). Content-based indexing and retrieval-by-example in audio. In *IEEE International Conference on Multimedia and Expo (II)*, pages 877–880.

[Logan and Salomon, 2001] Logan, B. and Salomon, A. (2001). A music similarity function based on signal analysis. In *Multimedia and Expo, 2001. ICME 2001. IEEE International Conference on*, pages 745 – 748.

[Manning et al., 2009] Manning, C. D., Raghavan, P., and Schütze, H. (2009). *Introduction to Information Retrieval*. Cambridge University Press, online edition.

[Marlow et al., 2006] Marlow, C., Naaman, M., Boyd, D., and Davis, M. (2006). Ht06, tagging paper, taxonomy, flickr, academic article, to read. In *HYPERTEXT '06: Proceedings of the Seventeenth Conference on Hypertext and Hypermedia*, pages 31–40, New York, NY, USA. ACM.

[McAfee, 2006] McAfee, A. P. (2006). Enterprise 2.0: The dawn of emergent collaboration. *MIT Sloan Management Review*, 47(3):21–28.

[McNee et al., 2006a] McNee, S. M., Riedl, J., and Konstan, J. (2006a). Accurate is not always good: How accuracy metrics have hurt recommender systems. In *Extended Abstracts of the 2006 ACM Conference on Human Factors in Computing Systems (CHI 2006)*, pages 1–5.

[McNee et al., 2006b] McNee, S. M., Riedl, J., and Konstan, J. A. (2006b). Being accurate is not enough: how accuracy metrics have hurt recommender systems. In Olson, G. M. and Jeffries, R., editors, *CHI Extended Abstracts*, pages 1097–1101. ACM.

[Memmel et al., 2008] Memmel, M., Kockler, M., and Schirru, R. (2008). Providing multi source tag recommendations in a social resource sharing platform. In Maurer, H., Kappe, F., Haas, W., and Tochtermann, K., editors, *Proceedings of I-MEDIA '08*, pages 226–233. Know-Center, Graz, Journal of Universal Computer Science. ISSN 0948-695x.

[Memmel and Schirru, 2007] Memmel, M. and Schirru, R. (2007). ALOE - a socially aware learning resource and metadata hub. In Wolpers, M., Klamma, R., and Duval, E., editors, *Proceedings of the EC-TEL 2007 Poster Session*. CEUR workshop proceedings. ISSN 1613-0073.

[Middleton et al., 2001] Middleton, S. E., De Roure, D. C., and Shadbolt, N. R. (2001). Capturing knowledge of user preferences: ontologies in recommender systems. In *K-CAP '01: Proceedings of the 1st international conference on Knowledge capture*, pages 100–107, New York, NY, USA. ACM.

[Montaner et al., 2003] Montaner, M., López, B., and de la Rosa, J. L. (2003). A taxonomy of recommender agents on the internet. *Artif. Intell. Rev.*, 19(4):285–330.

[Mooney and Roy, 2000] Mooney, R. J. and Roy, L. (2000). Content-based book recommending using learning for text categorization. In *DL '00: Proceedings of the fifth ACM conference on Digital libraries*, pages 195–204, New York, NY, USA. ACM.

[Morris et al., 2005] Morris, M., Pohlmann, T., and Young, G. O. (2005). How do users feel about technology? Forrester Research.

[Mortensen et al., 2008] Mortensen, M., Gurrin, C., and Johansen, D. (2008). Real-world mood-based music recommendation. In Li, H., Liu, T., Ma, W.-Y., Sakai, T., Wong, K.-F., and Zhou, G., editors, *Information Retrieval Technology*, volume 4993 of *Lecture Notes in Computer Science*, pages 514–519. Springer Berlin / Heidelberg.

[Nichols, 1997] Nichols, D. M. (1997). Implicit rating and filtering. In *Proceedings of the Fifth DELOS Workshop on Filtering and Collaborative Filtering*, pages 31–36.

[O'Reilly, 2005a] O'Reilly, T. (2005a). Web 2.0: Compact definition? - o'reilly radar. `http://radar.oreilly.com/archives/2005/10/web-20-compact-definition.html`. [Online; accessed 02-November-2010].

[O'Reilly, 2005b] O'Reilly, T. (2005b). What is web 2.0 - design patterns and business models for the next generation of software. `http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html`. [Online; accessed 22-August-2008].

[Osinski et al., 2004] Osinski, S., Stefanowski, J., and Weiss, D. (2004). Lingo: Search results clustering algorithm based on singular value decomposition. In *Intelligent Information Systems*, pages 359–368.

[Passant, 2010] Passant, A. (2010). Measuring semantic distance on linking data and using it for resources recommendations. In *Proceedings of the AAAI Spring Symposium "Linked Data Meets Artificial Intelligence"*, pages 93–98.

[Pazzani, 1999] Pazzani, M. J. (1999). A framework for collaborative, content-based and demographic filtering. *Artif. Intell. Rev.*, 13(5-6):393–408.

[Pazzani and Billsus, 1997] Pazzani, M. J. and Billsus, D. (1997). Learning and revising user profiles: The identification of interesting web sites. *Machine Learning*, 27(3):313–331.

[Pazzani and Billsus, 2007] Pazzani, M. J. and Billsus, D. (2007). Content-based recommendation systems. In Brusilovsky, P., Kobsa, A., and Nejdl, W., editors, *The Adaptive Web: Methods and Strategies of Web Personalization*, volume 4321 of *Lecture Notes in Computer Science*, chapter 10, pages 325–341. Springer, Berlin.

[Resnick et al., 1994] Resnick, P., Iacovou, N., Sushak, M., Bergstrom, P., and Riedl, J. (1994). Grouplens: An open architecture for collaborative filtering of netnews. In *1994 ACM Conference on Computer Supported Collaborative Work Conference*, pages 175–186.

[Rho et al., 2009] Rho, S., Han, B., and Hwang, E. (2009). Svr-based music mood classification and context-based music recommendation. In Gao, W., Rui, Y., Hanjalic, A., Xu, C., Steinbach, E. G., El-Saddik, A., and Zhou, M. X., editors, *ACM Multimedia*, pages 713–716. ACM.

[Russell, 1980] Russell, J. A. (1980). A circumplex model of affect. *Journal of Personality and Social Psychology*, 39(06):1161–1178.

[Sarwar et al., 2001] Sarwar, B., Karypis, G., Konstan, J., and Reidl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *WWW '01: Proceedings of the 10th international conference on World Wide Web*, pages 285–295, New York, NY, USA. ACM.

[Sarwar et al., 2000] Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2000). Application of dimensionality reduction in recommender systems-a case study. In *WebKDD-2000 Workshop*, pages 309–324.

[Schirru, 2010] Schirru, R. (2010). Topic-based recommendations in enterprise social media sharing platforms. In *RecSys '10: Proceedings of the fourth ACM conference on Recommender systems*, pages 369–372, New York, NY, USA. ACM.

[Schirru et al., 2011a] Schirru, R., Baumann, S., Freye, C., and Dengel, A. (2011a). Towards context-sensitive music recommendations using multifaceted user profiles. In *The Proceedings of the AES 42nd International Conference*, pages 54–59.

[Schirru et al., 2010a] Schirru, R., Baumann, S., Memmel, M., and Dengel, A. (2010a). Extraction of contextualized user interest profiles in social sharing platforms. *Journal of Universal Computer Science*, 16(16):2196–2213.

[Schirru et al., 2011b] Schirru, R., Baumann, S., Memmel, M., and Dengel, A. (2011b). Topic-based recommendations for enterprise 2.0 resource sharing platforms. In König, A., Dengel, A., Hinkelmann, K., Kise, K., Howlett, R. J., and Jain, L. C., editors, *Knowledge-Based and Intelligent Information and Engineering Systems*, volume 6881 of *Lecture Notes in Computer Science*, pages 495–504. Springer.

[Schirru et al., 2010b] Schirru, R., Obradović, D., Baumann, S., and Wortmann, P. (2010b). Domain-specific identification of topics and trends in the blogosphere. In *Proceedings of the 10th industrial conference on Advances in data mining: applications and theoretical aspects*, ICDM'10, pages 490–504, Berlin, Heidelberg. Springer-Verlag.

[Schult and Spiliopoulou, 2006] Schult, R. and Spiliopoulou, M. (2006). Discovering emerging topics in unlabelled text collections. In Manolopoulos, Y., Pokorný, J., and Sellis, T. K., editors, *ADBIS*, volume 4152 of *Lecture Notes in Computer Science*, pages 353–366. Springer.

[Schwarz, 2006] Schwarz, S. (2006). A context model for personal knowledge management applications. In Roth-Berghofer, T., Schulz, S., and Leake, D. B., editors, *Modeling and Retrieval of Context, Second International Workshop, MRC 2005, Edinburgh, UK, July 31 - August 1, 2005, Revised Selected Papers*, volume 3946 of *Lecture Notes in Computer Science*, pages 18–33. Springer.

[Segaran, 2007] Segaran, T. (2007). *Programming collective intelligence*. O'Reilly.

[Sen et al., 2006] Sen, S., Lam, S. K., Rashid, A. M., Cosley, D., Frankowski, D., Osterhouse, J., Harper, F. M., and Riedl, J. (2006). tagging, communities, vocabulary, evolution. In *Proceedings of the 2006 20th anniversary conference on*

Computer supported cooperative work, CSCW '06, pages 181–190, New York, NY, USA. ACM.

[Shardanand and Maes, 1995] Shardanand, U. and Maes, P. (1995). Social information filtering: algorithms for automating "word of mouth". In *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 210–217, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.

[Sinha, 2005] Sinha, R. (2005). A cognitive analysis of tagging. `http://rashmisinha.com/2005/09/27/a-cognitive-analysis-of-tagging/`. [Online; accessed 17-September-2010].

[Song et al., 2009] Song, S., Kim, M., Rho, S., and Hwang, E. (2009). Music ontology for mood and situation reasoning to support music retrieval and recommendation. In *Proceedings of the 2009 Third International Conference on Digital Society*, pages 304–309, Washington, DC, USA. IEEE Computer Society.

[Song and Park, 2007] Song, W. and Park, S. C. (2007). A novel document clustering model based on latent semantic analysis. In *Semantics, Knowledge and Grid, Third International Conference on*, pages 539–542.

[Sparck Jones, 1972] Sparck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21.

[Stahl and Roth-Berghofer, 2008] Stahl, A. and Roth-Berghofer, T. R. (2008). Rapid prototyping of CBR applications with the open source tool myCBR. In Bergmann, R. and Althoff, K.-D., editors, *Advances in Case-Based Reasoning*, pages 615–629. Springer Verlag.

[Tran and Cohen, 2000] Tran, T. and Cohen, R. (2000). Hybrid recommender systems for electronic commerce. In *Knowledge-Based Electronic Markets, Papers from the AAAI Workshop, AAAI Technical Report WS-00-04*, pages 78–83. Menlo Park, CA: AAAI Press.

[Ulges et al., 2009] Ulges, A., Koch, M., Borth, D., and Breuel, T. (2009). TubeTagger – YouTube-based concept detection. In *Proc. Int. Workshop on Internet Multimedia Mining*, pages 190–195. IEEE Computer Society.

[van Elst et al., 2008] van Elst, L., Kiesel, M., Schwarz, S., Buscher, G., Lauer, A., and Dengel, A. (2008). Contextualized knowledge acquisition in a personal semantic wiki. In Gangemi, A. and Euzenat, J., editors, *EKAW*, volume 5268 of *Lecture Notes in Computer Science*, pages 172–187. Springer.

[Wahlster et al., 2006] Wahlster, W., Schwarzkopf, E., Sauermann, L., Roth-Berghofer, T., Pfalzgraf, A., Kiesel, M., Heckmann, D., Dengler, D., Dengel, A., and Sintek, M. (2006). Web 3.0: Convergence of web 2.0 and the semantic web. *Technology Radar*, pages 1–23.

[Wal, 2007] Wal, T. V. (2007). Folksonomy. `http://vanderwal.net/folksonomy.html`. [Online; accessed 22-August-2008].

[Wang and Kong, 2007] Wang, R.-Q. and Kong, F.-S. (2007). Semantic-enhanced personalized recommender system. In *International Conference on Machine Learning and Cybernetics*, volume 07, pages 4069–4074.

[Weller, 2010] Weller, K. (2010). *Knowledge Representation in the Social Semantic Web*. de Gruyter Saur.

[Wikipedia, 2008] Wikipedia (2008). Tag (metadata) — wikipedia, the free encyclopedia. `http://en.wikipedia.org/w/index.php?title=Tag_(metadata)&oldid=232998858`. [Online; accessed 27-August-2008].

[Wikipedia, 2009] Wikipedia (2009). Ontology — wikipedia, the free encyclopedia. `http://en.wikipedia.org/w/index.php?title=Ontology&oldid=311909039`. [Online; accessed 4-September-2009].

[Wikipedia, 2010a] Wikipedia (2010a). Linked data — wikipedia, the free encyclopedia. `http://en.wikipedia.org/wiki/Linked_Data`. [Online; accessed 23-June-2010].

[Wikipedia, 2010b] Wikipedia (2010b). Mean absolute error — wikipedia, the free encyclopedia. `http://en.wikipedia.org/w/index.php?title=Mean_absolute_error&oldid=392983876`. [Online; accessed 16-November-2010].

[Wikipedia, 2010c] Wikipedia (2010c). Netflix prize — wikipedia, the free encyclopedia. `http://en.wikipedia.org/w/index.php?title=Netflix_Prize&oldid=397065212`. [Online; accessed 19-November-2010].

[Wikipedia, 2010d] Wikipedia (2010d). Root mean square deviation — wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=Root_mean_square_deviation&oldid=392989116. [Online; accessed 16-November-2010].

[Wikipedia, 2010e] Wikipedia (2010e). Wiki — wikipedia, the free encyclopedia. http://en.wikipedia.org/w/index.php?title=Wiki&oldid=405173855. [Online; accessed 4-January-2011].

[Witten and Frank, 2005] Witten, I. H. and Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann.

[Xu et al., 2003] Xu, W., Liu, X., and Gong, Y. (2003). Document clustering based on non-negative matrix factorization. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 267–273, New York, NY, USA. ACM.

[Yao, 1995] Yao, Y. Y. (1995). Measuring retrieval effectiveness based on user preference of documents. *J. Am. Soc. Inf. Sci.*, 46:133–145.

[Yi and Deng, 2009] Yi, M. and Deng, W. (2009). A utility-based recommendation approach for e-commerce websites based on bayesian networks. *Business Intelligence and Financial Engineering, International Conference on*, 0:571–574.

[Zhang and Hurley, 2009] Zhang, M. and Hurley, N. (2009). Novel item recommendation by user profile partitioning. In *WI-IAT '09: Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, pages 508–515, Washington, DC, USA. IEEE Computer Society.

[Ziegler et al., 2005] Ziegler, C.-N., McNee, S. M., Konstan, J. A., and Lausen, G. (2005). Improving recommendation lists through topic diversification. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 22–32, New York, NY, USA. ACM.

Curriculum Vitae

Contact Information

Name	Rafael Schirru
Address	Müllerstraße 91
	13349 Berlin
Telephone	+49 - 160 - 44 07 50 4
eMail	rschirru@gmx.de

Personal Information

Date of Birth	removed in electronic version
Place of Birth	removed in electronic version
Citizenship	removed in electronic version
Family Status	removed in electronic version

Employment History

German Research Center for Artificial Intelligence (DFKI)

04/2007 - today	Researcher in the Knowledge Management department
05/11-today	Project Voice2Social (multimodal access to social networks on mobile devices)
11/08-10/10	Project Social Media Miner (identification of topics and trends in the blogosphere)

04/08-09/09	Project MACE (providing access to vast amounts of architectural content from diverse European repositories)
03/08-12/08	Project C-LINK (a conference organization system with an integrated recommender system for events and conference attendees)
04/07-03/08	Project CoMet (a brokerage service for all sorts of digital learning content and its associated metadata)

Education

University of Kaiserslautern

10/2001 - 03/2007	Applied Computer Science (Enterprise Information Systems) Diploma Thesis: *Conception and Implementation of an Open and Adaptable Environment for Multidimensional Learning Objects*

Privates Johannes Gymnasium

06/2001	Graduation diploma

Spoken Languages

German	Mother tongue
English	Fluent
French	Basic
Italian	Basic

Projects

The project *Voice2Social* aimed at combining multimodal interaction technologies on mobile devices (e. g., smartphones and tablets) with new and diverse means for users to interact with social media. This project was financed by the IBB Berlin and co-financed by the EFRE fonds of the European Union from 05/2011 to 04/2013.

The project *Social Media Miner* was concerned with the analysis of user-generated content in the Web 2.0. Text mining algorithms were implemented to support automatic topic and trend detection in social media such as the blogosphere and microblogging platforms. For identified topics, reading recommendations were provided based on metrics from the field of Social Network Analysis. The project was financed by the IBB Berlin and co-financed by the EFRE fonds of the European Union from 11/2008 to 10/2010.

MACE (Metadata for Architectural Contents in Europe) is a European initiative which aims at improving architectural education. For that purpose it has integrated and connected vast amounts of content from diverse repositories. DFKI was a partner in the consortium and has provided the social resource sharing platform ALOE to connect users and to allow them to share and organize contents and metadata. MACE was co-funded by the EU eContentPlus program from 09/2006 until 09/2009.

C-LINK (Conference Link) is a Web-based tool for individual conference organization. C-LINK has been introduced with the KI 2008 conference in Kaiserslautern. The system serves as a showcase to present DFKI technologies. Its functionalities comprise among others an individual conference planner, retrieval of users with similar interests, and Web 2.0 style resource sharing. C-LINK was an internal DFKI project funded from 03/2008 until 12/2008.

CoMet stands for Collaborative Sharing of Metadata and can be considered as a brokerage service for all sorts of digital learning content and its associated metadata. Functionalities of social software were intended to connect users, that way enabling interaction and collaboration. CoMet was sponsored by the Stiftung Rheinland-Pfalz für Innovation from 04/2007 until 03/2008.

Selected Publications

2011 Rafael Schirru, Stephan Baumann, Martin Memmel, and Andreas Dengel. Topic-Based Recommendations for Enterprise 2.0 Resource Sharing Platforms. In Andreas König et al., editors, *Knowledge-Based and Intelligent Information and Engineering Systems*, volume 6881 of *Lecture Notes in Computer Science*, pages 495-504. Springer.

Rafael Schirru, Stephan Baumann, Christian Freye, and Andreas Dengel. Towards Context-Sensitive Music Recommendations Using Multifaceted User Profiles. *The Proceedings of the AES 42nd International Conference*, 2011, 54-59.

2010 Stephan Baumann, Rafael Schirru, and Bernhard Streit. Towards a storytelling approach for novel artist recommendations. In Detyniecki, M., Knees, P., Nürnberger, A., Schedl, M., and Stober, S., editors, *Adaptive Multimedia Retrieval*, volume 6817 of *Lecture Notes in Computer Science*, pages 1-15. Springer.

Rafael Schirru, Stephan Baumann, Martin Memmel, and Andreas Dengel. Extraction of Contextualized User Interest Profiles in Social Sharing Platforms. *Journal of Universal Computer Science*, 16(16):2196-2213.

Fernanda Pimenta, Darko Obradovic, Rafael Schirru, Stephan Baumann, and Andreas Dengel. Automatic Sentiment Monitoring of Specific Topics in the Blogosphere. *Workshop on Dynamic Networks and Knowledge Discovery (DyNaK 2010)*, 2010.

Rafael Schirru. Topic-Based Recommendations in Enterprise Social Media Sharing Platforms. *RecSys '10: Proceedings of the fourth ACM conference on Recommender systems*, *ACM*, 2010, 369-372.

Rafael Schirru, Darko Obradovic, Stephan Baumann and Peter Wortmann. Domain-Specific Identification of Topics and Trends in the Blogosphere. In Petra Perner, editor, *Advances in Data Mining. Applications and Theoretical Aspects, 10th Industrial Conference, ICDM 2010, Berlin, Germany, July 12-14, 2010. Proceedings, Springer, 2010, 6171*, 490-504.

2009 Martin Wolpers, Martin Memmel, Hans-Christian Schmitz, Martin Friedrich, Marco Jahn, and Rafael Schirru. Usage metadata based support for learning activity reflection. In Klaus Tochtermann and Hermann Maurer, editors, *Proceedings of I-KNOW '09*, *Graz*, *Austria*, pages 354-359. Journal of Universal Computer Science, 2009.

Martin Memmel, Michael Kockler, and Rafael Schirru. Providing Multi Source Tag Recommendations in a Social Resource Sharing Platform. In *Journal of Universal Computer Science*, *vol. 15*, *no. 3*, pages 678-691. Graz, Austria 2009.

2008 Martin Memmel, Michael Kockler, and Rafael Schirru. Providing Multi Source Tag Recommendations in a Social Resource Sharing Platform. In Hermann Maurer, Frank Kappe, Werner Haas and Klaus Tochtermann, editors, *Proceedings of I-MEDIA '08*, *Graz*, *Austria*, pages 226-233. Journal of Universal Computer Science, 2008.

Martin Memmel, Rafael Schirru, Elia Tomadaki, and Martin Wolpers. Towards the Combined Use of Metadata to Improve the Learning Experience. In Paloma Díaz, Kinshuk, Ignacio Aedo, and Eduardo Mora, editors, *Proceedings of the 8th IEEE International Conference on Advanced Learning Technologies*, *Santander*, *2008*, pages 930-932.

2007 Martin Memmel and Rafael Schirru. ALOE - A Socially Aware Learning Resource and Metadata Hub. In Martin Wolpers, Ralf Klamma and Erik Duval, editors, *Proceedings of the EC-TEL 2007 Poster Session.* CEUR workshop proceedings, 2007.

Martin Memmel and Rafael Schirru. Sharing Digital Resources and Metadata for Open and Flexible Knowledge Management Systems. In Klaus Tochtermann and Hermann Maurer, editors, *Proceedings of the 7th International Conference on Knowledge Management (I-KNOW)*, *Graz*, *Austria*, pages 41-48. Journal of Universal Computer Science, 2007.