USERS AND ENTITIES ON THE WEB

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

DOKTORIN DER NATURWISSENSCHAFTEN

Dr. rer. nat.

genehmigte Dissertation
von

Dipl.-Ing. Tereza Iofciu

geboren am 31. März 1982, in Bukarest, Rumänien

2012

Referent: Prof. Dr. Wolfgang Nejdl
Korreferent: Prof. Dr. Heribert Vollmer
Tag der Promotion: 6. December 2012

ABSTRACT

With the ever increasing amount of information people make available on the Web, there is an obvious need for better understanding both the user and this information. Recent progress in the research areas such as Information Extraction and Information Retrieval enables the development of systems providing better experiences to Web users. Social Web applications have become very popular, nowadays people spend most of their online time interacting and publishing information via such systems. The main user activity in Social Bookmarking systems is sharing information, such as pictures, videos and other Web resources, and annotating these objects with tags. The tagging activity can then be interpreted bidirectionally, to understand the users' interests and to further describe the tagged objects.

In this thesis we first focus on ways to extract and then use entity profiles. A good example of entities are persons, thus also Web users, and the best way to build a user profile is to aggregate the information created and posted by the respective user on the Web. In this part of the work we extract tag based user profiles, either from authored publications, or from resources tagged by the users on the Web. We then present approaches for using the user profiles for recommending resources and tags to the users. We also present an approach for identifying users across Social Networks based on their implicit and explicit profiles, which falls under the problem of entity identification.

In the second part of the thesis we focus on the problem of entity search on the Web. For example, for an entity query like "cities in Germany" users expect systems to retrieve entities, such as "Hamburg","Berlin",etc. instead of just documents about Germany. Creating meaningful entity profiles based on tags as presented in the first chapter can help better describe the entities. In this chapter we show how semi-structured information about entities can improve search. For this we used the link and category information about entities in Wikipedia as an additional step to entity textual search. Another important part of Web data which can be used for search, besides the documents themselves, are the actions of the searching users. Thus, we also exploit Click-through and Session data, gathered from the Bing! search engine, in order to facilitate entity search on the Web.

Keywords: Social Networks, User Identification, Entities

ZUSAMMENFASSUNG

Mit der stetig zunehmenden Menge an Information, die Benutzer im Web verfügbar machen, besteht die Notwendigkeit sowohl die Informationen als auch die Benutzer besser zu verstehen. Fortschritte in Bereichen wie Information Extraction oder Information Retrieval ermöglichen die Entwicklung von Systemen, die das Nutzererlebnis verbessern. Social Web Anwendungen sind zunehmend populärer geworden, sodass heutzutage viele Menschen einen Großteil ihrer Zeit damit verbringen, mit diesen Anwendungen zu interagieren und dort Information zu veröffentlichen. Der Hauptzweck von sogenannten Social Bookmarking Systemen ist etwa das Teilen von Informationen wie Bildern, Videos und anderen Web Ressourcen sowie das Annotieren dieser Inhalte mit Schlagwörtern (tagging). Tagging Aktionen können bidirektional interpretiert werden, um einerseits die Interessen der Benutzer zu verstehen und andererseits weitere Erkenntnisse bezüglich der annotierten Inhalte zu erlangen.

In dieser Doktorarbeit konzentrieren wir uns zunächst auf Ansätze für die Extraktion und Verwendung von Profilen von Entitäten und im speziellen von Benutzerprofilen. Die besten Strategien für die Erstellung von Benutzerprofilen bestehen darin, Information zu aggregieren, die der jeweilige Benutzer im Web erstellt und veröffentlicht hat. Wir diskutieren hierbei Strategien, die tag-basierte Benutzerprofile entweder von verfassten Publikationen oder von Inhalten extrahieren, die die Benutzer im Web annotiert haben. Zudem präsentieren wir Ansätze, die die Benutzerprofile verwenden um den jeweiligen Benutzern Inhalte und Tags zu empfehlen. Ferner erforschen und vergleichen wir Methoden, die basierend auf impliziten und expliziten Profildaten das Identifizieren von Benutzern über die Grenzen von Social Web Systemen hinweg ermöglichen, und leisten somit einen wichtigen Beitrag dazu Entitäten im Web zu identifizieren.

Im zweiten Teil der Doktorarbeit beschäftigen wir uns mir dem Problem Entitäten im Web zu suchen (entity search). Eine Anfrage wie "Städte in Deutschland" zielt zum Beispiel darauf ab anstatt einer Liste von Dokumenten, die Informationen zu Deutschland enthalten, direkt eine Liste von Entitäten wie "Hamburg", "Berlin", etc. zu erhalten. Die Erstellung von aussagekräftigen Profilen für Entitäten basierend auf Tags entsprechend kann hierbei erneut hilfreich sein. In diesem Teil zeigen wir zudem wie teilstrukturierte Informationen über Entitäten die Suche verbessern können. Hierzu verwenden wir Informationen über Links und Kategorien von Entitäten in Wikipedia als zusätzliches Wissen für die textbasierte Suche nach Entitäten. Ein weiterer wichtiger Teil der Daten im Web, die neben den Webinhalten für die Suche genutzt werden können, sind Verwendungsdaten, die durch die Suchaktionen der Benutzer generiert werden. In einem weiteren Experiment im Rahmen der Suchmaschine Bing! untersuchen wir daher den positiven Einfluss von Verwendungsdaten wie Click und Web Session Daten auf die Suche nach Entitäten.

Schlagworte: Soziale Netzwerke, Identifikation von Benutzern, Entitäten

FOREWORD

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

In Chapter 3 we describe contributions included in:

- *Finding Communities of Practice from User Profiles Based on Folksonomies.* Jörg Diederich, Tereza Iofciu. In: Proceedings of the 1st International Workshop on Building Technology Enhanced Learning solutions for Communities of Practice (TEL-CoPs06), co-located with the First European Conference on Technology-Enhanced Learning, 2006. [DI06]
- *Time based Tag Recommendation using Direct and Extended Users Sets.* Tereza Iofciu, Gianluca Demartini. In: ECML PKDD Discovery Challenge 2009 (DC09), volume 497,pages 99-107, Bled, Slovenia, September 2009. [ID09]
- *Identifying Users Across Social Tagging Systems.* Tereza Iofciu, Peter Fankhauser, Fabian Abel, Kerstin Bischoff. In: Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media (ICSWM'11), Barcelona, Spain, July 2011. [IFAB11]

Chapter 4 presenting entity search on the Web is built upon the work published in:

- *Exploiting Click-Through Data for Entity Retrieval.* Bodo Billerbeck, Gianluca Demartini, Claudiu-S Firan, Tereza Iofciu, Ralf Krestel. In: Proceedings of the 33rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2010, Geneva, Switzerland, July 19-23, 2010. ACM. [BDF+10a]
- *Ranking Entities Using Web Search Query Logs.* Bodo Billerbeck, Gianluca Demartini, Claudiu-S Firan, Tereza Iofciu, Ralf Krestel. In: Research and Advanced Technology for Digital Libraries, 14th European Conference (ECDL 2010), September 6-10, 2010. Glasgow, UK.[BDF+10b]
- *L3S at INEX 2008: Retrieving Entities using Structured Information.* Nick Craswell, Gianluca Demartini, Julien Gaugaz, Tereza Iofciu. In: Advances in Focused Retrieval, 7th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2008, Dagstuhl Castle, Germany, December 15-18, 2008. [CDGI08]

- *ReFER: effective Relevance Feedback for Entity Ranking.* Tereza Iofciu, Gianluca Demartini, Nick Craswell, Arjen P. de Vries. In: Advances in Information Retrieval, 33th European Conference on IR Research, ECIR 2011, Dublin, Ireland, April 2011. [IDCdV11]

During my Ph.D. studies I also published a series of articles which complement the core topics of this thesis. These articles focus, for example, on entity search in enterprise systems or on the temporal evolution of entities. A complete list of the complementary articles follows:

- *Evaluating the Impact of Snippet Highlighting in Search.* Tereza Iofciu, Nick Craswell, Milad Shokouhi. In: Understanding the user - Logging and interpreting user interactions in information retrieval. Workshop in Conjunction with the ACM SIGIR Conference on Information Retrieval., 2009. [ISC09]

- *An Architecture for Finding Entities on the Web.* Gianluca Demartini, Claudiu S. Firan, Mihai Georgescu, Tereza Iofciu, Ralf Krestel Wolfgang Nejdl. In: 7th Latin American Web Congress LA-WEB/CLIHC, pages 230-237, November 2009. [DFG+09]

- *A Model for Ranking Entities and Its Application to Wikipedia.* Gianluca Demartini, Claudiu-S Firan, Tereza Iofciu, Ralf Krestel, Wolfgang Nejdl. In: 6th Latin American Web Congress (LA-WEB 2008), Vila Velha, Espirito Santo, Brasil, October, 2008.[DFI+08]

- *Semantically Enhanced Entity Ranking.* Tereza Iofciu, Gianluca Demartini, Claudiu-S Firan, Wolfgang Nejdl. In: Web Information Systems Engineering - WISE 2008, 9th International Conference, Auckland, New Zealand, September 1-3, 2008. Proceedings, pp. 176-188, 2008.[DFIN08]

- *Terminology Evolution in Web Archiving: Open Issues.* Nina Tahmasebi, Tereza Iofciu, Thomas Risse, Claudia Niedere, Wolf Siberski. In: Proc. of the 8th International Web Archiving Workshop in conjunction with ECDL 2008, Aarhus, Denmark[TIR+08]

- *Relation Retrieval for Entities and Experts.* Jianhan Zhu, Arjen P. de Vries, Gianluca Demartini, Tereza Iofciu. In: Future Challenges in Expertise Retrieval (fCHER 2008), SIGIR 2008 Workshop, Singapore, July, 2008.[ZdVDI08]

- *L3S at INEX 2007: Query Expansion for Entity Ranking Using a Highly Accurate Ontology.* Tereza Iofciu, Gianluca Demartini, Claudiu-S Firan. In: Focused Access to XML Documents, 6th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2007, Dagstuhl Castle, Germany, December 17-19, 2007.[DFI07]

- *Role based Access Control for the interaction with Search Engines.* Alessandro Bozzon, Tereza Iofciu, Wolfgang Nejdl, Antonio Vincenzo Taddeo, Sascha Tönnies. In: Proceedings of 1st International Workshop on Collaborative Open Environments for Project-Centered Learning, September 2007, Crete, Greece[BIN$^+$07]

- *Integrating databases, search engines and Web applications: a model-driven approach.* Alessandro Bozzon, Tereza Iofciu, Wolfgang Nejdl, Sascha Tönnies. In: Web Engineering, 7th International Conference, ICWE 2007, Como, Italy, July 16-20, 2007, Proceedings, pp. 210-225, 2007, Springer, 978-3-540-73596-0.[BINT07]

- *ExpertFOAF Recommends Experts.* Tereza Iofciu, Jörg Diederich, Peter Dolog, Wolf-Tilo Balke. In: Proceedings of the 1st International ExpertFinder Workshop, Berlin, Germany, 16th January 2007.[IDB07]

- *The Beagle++ Toolbox: Towards an Extendable Desktop Search Architecture.* Ingo Brunkhorst, Paul-Alexandru Chirita, Stefania Costache, Julien Gaugaz, Ekaterini Ioannou, Tereza Iofciu, Enrico Minack, Wolfgang Nejdl, Raluca Paiu. In: Semantic Desktop Workshop (SemDesk), November 2006, Athens, GA, USA.[BCC$^+$06]

- *Extracting Semantics Relationships between Wikipedia Categories.* Sergey Chernov, Tereza Iofciu, Wolfgang Nejdl, Xuan Zhou. In: SemWiki2006, First Workshop on Semantic Wikis - From Wiki to Semantics, Proceedings, co-located with the ESWC2006, Budva, Montenegro, June 12, 2006.[CINZ06]

- *Keywords and RDF Fragments: Integrating Metadata and Full-Text Search in Beagle++.* Tereza Iofciu, Christian Kohlschütter, Raluca Paiu, Wolfgang Nejdl. In: Workshop on The Semantic Desktop - Next Generation Personal Information Management and Collaboration Infrastructure at the International Semantic Web Conference 6 November 2005, Galway, Ireland.[IKNP05]

Contents

Table of Contents **ix**

List of Figures **xi**

1 Introduction **1**

1.1 Problems Addressed in this Thesis 2

1.2 Proposed Solutions 5

1.3 Thesis Structure 6

2 Related Work **9**

2.1 User Profiling and Recommendations 9

2.2 User and Entity Identification 11

2.3 Entity Search on Wikipedia and on the Web 13

3 User Profiles on the Web **17**

3.1 Tag based profiles and recommendations 18

3.1.1 A Tag-Based User Profile Generator 19

3.1.2 Approaches to Creating and Maintaining User Profiles 21

3.1.3 Using Tag-Based Profiles for Recommendations 23

3.1.4 Time based Tag Recommendation using Direct and Extended Users Sets 26

3.1.5 Evaluation 29

3.1.6 Discussion 30

3.2 Users profiles in Social Networks 33

3.2.1 Research Challenge 33
3.2.2 Matching Users across Sites 34
3.2.3 Matching Users based on their Tags 34
3.2.4 Matching Users based on Explicit Usernames 37
3.2.5 Matching Users based on Combined Profiles 38
3.2.6 Evaluation 39
3.2.7 Discussion 51

4 Entity Search on the Web 53
4.1 Finding Entities in Semi-Structured Data 54
4.1.1 Category expansion in Wikipedia 55
4.1.2 Link-based Relevance Feedback for Entity Ranking 56
4.1.3 Evaluation 59
4.1.4 Per Topic Analysis 65
4.1.5 Discussion 65
4.2 Finding Entities exploiting Click-through and Session data 67
4.2.1 Constructing and Entering the Graphs 68
4.2.2 Walking the Graphs for Entity Ranking 69
4.2.3 Evaluation 74
4.2.4 Discussion 77

5 Conclusions and Future Work 79

A Curriculum Vitae 83

Bibliography 85

List of Figures

3.1 Tags recommended based on the set of users who have annotated the query resource. . . . 28

3.2 Tags recommended based on the set of users who have annotated the query resource and users in the immediate neighborhood of the *direct user set*. . . . 29

3.3 How much do the distinct tags of the individual, service-specific profiles overlap? 43

3.4 Success@1 and MRR vs. profile size 46

3.5 Profile mappings based on username, reciprocal rank for the different string similarity distances 48

3.6 Success at rank 1 for each pair of systems. S stands for StumbleUpon, F for Flickr and D for Delicious. For the mixture approach the best tag- and username-based approaches are combined with mixture coefficient λ=0.86. 49

3.7 Success at rank 1 for different approaches (mixture coefficient λ=0.86) and settings when dealing with aggregated profiles. *D_S* stands for the setting where individual profiles are aggregated from Delicious and StumbleUpon, etc. 50

4.1 Three layer graph, with input node entity 'Boston, Massachusetts'. Solid edges indicate hard categories, dashed edges indicate smooth categories. 57

4.2 Comparison of runs submitted at INEX 2008 for the XER task when merged with ReFER using the topic examples for different λ. 60

4.3 Improvement of xinfAP for each run using all results in top-k retrieved as seed of the algorithm, combining with $\lambda = 0.5$. 62
4.4 Improvement of xinfAP for each run using only relevant results in top-k retrieved as seed of the algorithm, combining with $\lambda = 0.5$. 63
4.5 Schemes of a Click Graph (A), connecting an ER query q with entities $q_{l,i}$ via URLs $U_{l,j}$ where l indicates the level, and of a Session Graph (B) connecting a ER query q with queries $q_{2,i}$ on level 2. 68
4.6 Selection of walked queries for the query "salad recipes". 70

1 Introduction

Social Web applications have become an integrated part of online life. Nowadays people have online accounts on various social Web portals where they leave plenty of multifaceted profile data. Users share their pictures, videos, documents, bookmarks and other information on social platforms (such as *Flickr*[1], *YouTube*[2], *Del.icio.us*[3], *etc.*), digital libraries (such as *CiteSeer*[4] or *GoogleScholar*[5]) or online encyclopedias (such as *Wikipedia*[6]). Users then annotate these objects using tags (free-text keywords) to facilitate retrieval of these resources, to express their opinion or expertise or merely to present themselves (cf. user incentives described in [MNBD06]). The tagging activity of a user can be interpreted bidirectionally. First, one can see the tags as an enrichment of the description of the objects they are assigned to, and conversely, these tags are also describing the user and the user's interests ([FNP07]).

The ease of use of the Social Web makes it possible for mostly anyone to express their opinion about online resources. The tags assigned by the users to objects, the objects and the users themselves are an important resource used by the bookmarking systems for enhancing the user experience [Mic07]. The collaboratively created data is then used by systems in services such as resource search, user search, personalization or recommendation. In order for these services to perform well it is important that both the Entities (users and objects) involved are understood. In this thesis we focus on learning how to build tag-based user and object profiles from the data available in various online collaborative systems. We also show how building meaningful profiles improves the performance of various information retrieval services, such as user and object search and recommendation.

[1] Flickr. http://www.flickr.com

[2] YouTube. http://www.youtube.com

[3] Delicious. http://delicious.com

[4] CiteSeer. http://citeseerx.ist.psu.edu/

[5] Google Scholar. http://scholar.google.com/

[6] Wikipedia. www.wikipedia.org/

1.1 Problems Addressed in this Thesis

Because of the success of social Web applications, vast amounts of data are daily published by more and more Web users. As the users need not be technically experienced in order to publish on the Web nowadays, the content people post online is no longer controlled. Also, as people can post any type of content, users are not required to tag objects following an expert ontology. Thus, on the side of reusing the published content, managing the data and finding useful information on the social Web is often a time consuming and difficult task. It is also difficult for Web applications to always understand the user's intent and context. In each system they use, users leave a plethora of information which then has to be interpreted by the system. A core problem is that interoperable models and techniques for representing and managing profile information across the Web are not yet established. In contrast, users often have accounts on different Web systems which are not connected with each other. Consequently, user profiles are not interlinked and then each system has only a piece of the user profile.

Also, in the past, it was quite clear what users were searching for: documents. Given the different types of objects users publish these days on the Web, it is also the case that now users are not searching only for documents anymore. One can perform specialized search on each Web platform, i.e. when searching for videos going directly to Youtube. However, users may prefer to have a uniform search interface such as *Google*[7] which should direct them to the right type of results and rather serves as an question answering machine than a pure document retrieval system. Therefore, it is, on one hand, important to understand the meaning of Web resource and, on the other hand, the demands of the users who are seeking for certain types of Web resources. Social tagging provides indicators for both the meaning of Web resources and the preferences of the users: the tags of a resource may, for example, describe the content or utility of a resource while the tags that have been applied by a user may describe the preferences and demands of the user. Understanding both the Web resources that convey information that people may demand and understanding the demands and preferences of the people are essential research challenges that have to be solved in order to engineer enhanced Web applications that can serve as adaptive answering machines.

In this thesis, we investigate both of the aforementioned challenges. In particular, we propose solutions to the following research problems.:

Problem 1 *How to build and maintain user profiles given the user's tagging activity?*

Having a good model for collecting and maintaining user profiles from information published on the Web is a crucial step when trying to understand the user. Once

[7] Google. www.google.com

a social Web application has managed to collect meaningful information regarding the users, it can help the users with important services, such as personalization, recommendation or re-visiting. Considering that the main activity a user has on social publishing Web applications is posting and tagging objects, it has become a common approach to define the user's preferences via the tagged objects and the assigned tags. In such Web applications, the user's interaction with the system, i.e. the user's profile, is automatically used in the services provided on the platform. A well known problem is that users may interact differently with a system in different contexts (for example, when being at home as opposed when being at work). Also a user's interest usually changes over time. Thus, user profiles have to capture the different user personae, so that the services provided by the system do not irritate the users. Additionally, many of the social tagging systems are specialized on a certain type of published items (e.g. YouTube for videos, LastFm[8] for music, etc), thus also the tags people assign to items will be domain specific. It is thus important to be able distinguish which are the representative tags defining a user's interests independent of the domain, or depending on the domain.

Problem 2 *How to exploit user profiles for improving the user's experience across social Web applications?*

The mass publishing on the Web is basically useless if the information cannot be discovered and consumed by other users. When dealing with tagged objects, there are several services that tagging systems focus on: finding similar users, recommendation of new objects to the users, rediscovering of entities that have been previously of interest, searching for relevant entities and personalization. For people in a community (such as professors and students in the research community), a well-defined user profile expressing their current and past interests is highly valuable. Such profiles can help to find persons who work on related topics and, thus, help to facilitate cooperation within the community. Similarly, social Web systems such as FaceBook[9] recommends to a user other people that the user "might know". In most of these services the users' tagging activity is used for understanding the user and the tagged objects. Users are compared based on their profiles (tagged objects and tags). The main problem when comparing users based on the tagged objects is sparsity, thus, as there are less possible tags than objects, it is more useful to define the user profiles based on the assigned tags. Additionally, as many users have different accounts on many systems, thus, they have different profiles on different systems (e.g. tags related to music on LastFm and general tags on StumbleUpon). In this situation, for understanding completely the user it is important to be able to identify these profiles and combine the profiles.

[8] LastFm http://www.last.fm

[9] FaceBook www.facebook.com

Problem 3 *How to exploit explicit user tags for improving typed item retrieval?*

The main activity people perform when on the Web, besides browsing, is searching for information in general or for specific entities. In this context an entity is a typed object, where the type can have different granularities. Most of the search application already provide typed search at a general level, i.e. the user is able to search for videos of images or publications, but the user is not easily able to search for objects with more specialized type, such as persons born in Germany or hybrid cars, etc. On the Web published objects can be described via textual content, manual or automatically extracted attributes and implicitly or explicitly assigned tags. As not all objects have a textual content (e.g. videos and images have none) and manually specifying attributes is a tedious task for the publishers, it is very important to be able to use the tags assigned by people in order to further improve the retrieval process. Nevertheless, entity searching is a complex task, on one side the user intention has to be correctly understood from the query (i.e. what type of entities is the user interested in and what are their attributes) and on the other side the system has to retrieve objects matching the user requirements, both in type and in attributes.

Problem 4 *How to exploit users behavior for allowing typed item retrieval on the Web?*

Current Web search engines retrieve textually relevant Web pages for a given keyword query even if the information need targets entities. An Entity Retrieval (ER) system should find entities directly. Instead of the user browsing through all Web pages retrieved by the search engine, a list of relevant entities can be presented to the user. This would not only save the user's time but also improve search experience. Consider queries that should return a list of entities and how difficult it is nowadays to compile such a list based on keyword search results. A user looking for a list of "films shot in Venice" or "Spanish dishes" will have difficulties to find suitable answers. She has to manually compile the result list by extracting entities from the retrieved documents which is a cumbersome task. The task of aggregating information about entities is also expensive on the search system's side, considering the amount of published information on the Web. As users post millions of queries per day to search engines, the search behavior, i.e. queries and clicks, can prove to be a valuable source of information. The fact that many users rephrase a query or that they click on the same URL for different queries implies that there are certain relations between the queries. One of the main advantages of exploiting queries also for Entity search is that usually queries are short and less expensive to mine.

1.2 Proposed Solutions

Our proposed solutions to the mentioned challenges are based on tags - the short textual descriptions that people assign to content objects, such as pictures, videos, Web pages, published publications, etc. The increasing popularity of Web applications and their simplicity of use makes it possible for more and more content to be published online. Also because people collaboratively tag objects online, systems can use these tags to better understand the users and the objects and the relations between them. In such Web applications where tagging is possible, tags are the main information resource used for profiling the user and the objects.

Tag-based User Profiling. In this thesis we propose a model for creating and maintaining user profiles in a tagged corpora and show how these profiles can be used in different recommendation services. Additionally, as users have accounts on various social Web applications, for providing users with good personalization services, it is important to be able to aggregate the profiles each user has on the different sites. We propose an approach for identifying users across social Web sites based on their tags and also for aggregating their profiles. The set of possible tags used in a system is to some extent limited to the vocabulary of one or more languages, whereas the amount of tagged objects can be unlimited, millions of new objects are created and published everyday. Thus, for avoiding the sparsity issue, we model a user profile as the set of the tags the user assigned to various objects.

Exploitation of Tag-based Profiles. As people usually have accounts on different social Web applications, each such system sees only a part of the user's profile. For fully understanding a user it would be useful to identify the different accounts the user has on the Web and aggregate the incomplete profiles. Connecting the different online accounts of a user brings benefits and drawbacks. While the aggregation of (tag-based) profiles reveals more information about users that is beneficial for personalization [AHHK10], the interlinkage of profile information might also be risky. For example, recently *PleaseRobMe*[10] attracted public's attention as they exploited *foursquare*[11] to detect the current location of Twitter users and identify – given the linkage to the address of these users – houses and apartments that are easy to burgle as the inhabitants are currently traveling. With the Social Graph API[12] Google is pursuing the related goal of tracking user identities and friendships across different Web 2.0 platforms.

(Un)fortunately, automatically connecting the different Social Web identities of the users is difficult because they might (possibly on purpose) use varying usernames or have unequal profiles (e.g. fields such as homepage, birthday, etc.) on the different systems [CC09]. Another source for profile mapping is implicit preference data from user interactions such as tagging. However, types of tags and their usage vary across

[10] http://pleaserobme.com

[11] http://foursquare.com

[12] http://code.google.com/apis/socialgraph/

systems as they are influenced by system design like tagging support, tagging rights, tagged objects or connectivity [MNBD06]. Having different tags or variance in tag prevalence across different systems makes it difficult for identifying users based on their implicit tagging profile.

Using Tags for improving Entity Search. While there are many types of content objects published on the Web the automatically extracted metadata doesn't always fully describe the objects. For example, for multimedia objects the only available textual description available is most of the times only the title. The available tags are thus a rich source of information, enhancing the objects descriptions and also helping in the task of knowledge discovery.

Exploiting User Search Behavior for Entity Search. In search applications, a rich source of implicit tags for Web object is the click-through data, where a query posted by a user is considered to be an implicit tag for a retrieved and clicked URL. These "tags" can then be further used to improve typed search. Additionally relations between queries can be inferred based on the user's search behavior. Many times users refine their queries, this action may mean that the different query terms are related. Some of the terms could represent entities and some terms their attributes.

The contributions of this thesis are manifold: (i) Firstly, we propose a model for tag-based user profiles, then (ii) we present usage scenarios of the tag-based profiles, in recommender services and in user identification and profile aggregation; and (iii) we show how tags can be used for enhancing performance of existing typed object search systems, i.e. entity search systems; and (iv) additionally we show how entity search is possible directly at tag level, by using click-through and session data from search engines.

1.3 Thesis Structure

The rest of the thesis is structured as follows:

In Chapter 2 we give an overview of the related work, which covers three main areas: user profiling and recommendations in Section 2.1, user and entity identification in Section 2.2; and entity search on Wikipedia and on the Web in Section 2.3.

The first two problems (*Problem 1* and *Problem 2*) are addressed in Chapter 3, where we start with defining a model for building and maintaining tag-based user profiles in tagged corpora on the Web. We show how these profiles, with the help of standard recommender techniques, can be used to provide users with suggestions of related resources or even persons with similar interests. We then present a prototype implementing a rudimentary system for creating tag-based user profiles in the digital library domain and using a user-item based recommender system to find potential people to extend a user's community of practice. We present first approaches for tag recommendation using graph information. We base our approaches on two sets of users, the direct set of users who have tagged a specific resource and the extended user set consisting also of the neighbors of the direct users. Additionally, we investigate how the temporal information of tag assessment can improve the recommendation effectiveness.

In the second part of the chapter, in Section 3.2, we focus on the topic of user identification on the Web. We propose different strategies for user matching across systems based on two types of user profiles: the implicit, given by the users' tagging practices, and explicit, in our case the usernames. We examined the Web profiles of users from three different social networking websites, Flickr, Delicious and StumbleUpon. For the tag-based profiles we introduce a symmetric variant of BM25 using site specific statistics and compare it against different standard measures. We also experiment with various string similarity measures for the username comparison and with combining and aggregating the different information sources. We evaluated the different approaches on more than 300 users with public profiles on the three analyzed systems. The best results were achieved by using Longest Common Sequence (LCS) based distance for username comparison and BM25 with site specific IDF for the tag specific profiles. We also experimented with aggregated user profiles from different system pairs and show how this technique can lead to better understanding the users.

Chapter 4 focuses on the other aspect of tag usage, namely, to enhance object description (*Problem 3* and *Problem 4*). Here we show how tags can be used to aid typed object (entity) retrieval. Based on the way tags are assigned, two different types of tags are considered in this chapter: explicit (Section 4.1) and implicit (Section 4.2) tags. In Section 4.1 we present a model for relevance feedback (RFB) in the entity retrieval scenario. The proposed model is based on weight propagation in a directed acyclic graph that represents links between entity descriptions. As experimental setting we use Wikipedia as a repository of such entity descriptions and evaluate our approach on the INEX 2008 benchmark. We use the submitted runs as baselines and show, firstly, that performing fusion with the result of our algorithm using relevant entity examples as initial seed always improves over the baseline effectiveness. We also evaluate our algorithm in a pseudo-RFB fashion, by using only the top retrieved entities as seed and show how top 10 entities yields the best improvement.

In the second part of the chapter, in Section 4.2, we present approaches for answering ER queries exploiting human behavior stored in search engine query logs. From the query logs we construct click and session graphs with queries and clicked URLs as nodes. We then perform a Markov random walk on the graphs in order to rank queries which contain relevant entities to a given ER query. We created and made available for download a gold standard of 81 Entity Ranking queries based on Wikipedia "List Of" pages. Given the created ground truth we experiment with both graphs and show how integrating results from both the click and the session graph yields best effectiveness.

Finally Chapter 5 summarizes the contributions of this thesis and discusses future ideas and problems for future work.

2 Related Work

The work presented in this thesis is closely related to the tasks of user and entity profiling, and on how these profiles can be used for improving Web services provided to the users. The following paragraphs present and discuss existing approaches for user profiling and recommendations (Section 2.1), user and entity identification (Section 2.2), entity search on Wikipedia and on the Web (Section 2.3).

2.1 User Profiling and Recommendations

There are different approaches to extracting user profiles from users' past activities and using them for discovering and analyzing communities. In [CDNS05], the similarity between peers in social collaboration networks is used to improve search in a peer-to-peer network. The similarity is computed based on publications and their references. The user profile is build based on the publications the user has stored on her desktop. This approach is too broad as the documents a user stores are usually not focused enough. The system takes into account all publications found, including ones dealing with topics the user may no longer have interest in or that the user has stored without even reading them or working on the topic.

Middleton et al. [MSR04] present a recommender system for online academic publications where user profiling is done based on a research paper topic ontology. The system monitors what research papers a group of person has downloaded from the web and stores them on a server. For all downloaded research papers, terms are extracted from the full text using standard information retrieval techniques to be able to represent the paper with term vectors. The system uses different classifiers to assign topics to the papers. User profiles are automatically built based on the vector-representation of those research papers, downloaded by a particular person in the monitored group of persons, and can be refined based on relevance feedback. Finally, the system gives recommendations for each user based on the user's profile. While an automatic update of the profile based on actual browsing of papers (similar to other publication recommender systems [AHLP05, PMPG04]) can reduce the efforts for creating and maintaining user profiles, this is in contrast to the issue that user profiles are typically rather stable over time, while the 'browsing task' is often focused on a short-term goal (e.g., help a colleague to find something or explore a topic which finally turns out not to be interesting). Hence, not all browsed documents are relevant to the user, even if we take into account the time spent on the respective document. Also, we would like to limit the collection of explicit relevance feedback which can create quite a workload for the user. Furthermore, the approach is pretty intrusive as it requires the monitoring of the browsing behavior of a group of persons. In contrast, our approach is based on publicly available information about objects and manually-assigned tags of objects. As manually assigned tags are assumed to be highly accurate, our approach does not suffer from the inaccuracy of an automatic classification system.

Existing systems to recommend publications in the domain of research are mainly keyword-based search engines (e.g., google scholar, ACM digital library etc.). They are mainly intended to fulfill short-term search objectives (find a paper with a specific title, find the paper for a specific author etc.). However, some papers are difficult to find based on keywords only, especially if a research domain is already well known. Furthermore, once a researcher has written a paper, she might turn to a different topic within her research interests, but still would like to be informed about the development in some of the topics, she has previously worked on. Hence, a recommender system for research papers [MAC+02] based on a long-term user profile is highly desirable. While the issue of user profiles has been found to be highly relevant for recommender systems [MLdlR03], it has not been addressed sufficiently in the literature at the time the work in Chapter 3 was done, and there were no existing systems which shared the user profiles they were using to take advantage of the distributed knowledge about the users. This gap was intended to be filled by our TBProfile prototype presented in Section 3.1.

Previous work on tag recommendation mainly distinguish between those looking at the content of the resources and those looking at the structure connecting users, resources, and tags. Approaches looking at content of resources for tag recommendations are, for example, [Mis06] which looks at content-based filtering techniques. In [XFMS06] the authors also look at collaborative tag suggestion in order to identify most appropriate tags.

A specific area of this field looks at recommending tags focusing on an individual user rather than providing general recommendation for a resource. In [Lip08] they first create a set of candidate tags to be recommended and then they filer it based on the previous tag a particular user has assigned in the past. In [JMH+08] the FolkRank algorithm is evaluated and compared with simpler approaches. This is a graph based approach that computes popularity scores for resources, users, and tags based on the well-known PageRank algorithm exploiting the link structure. The assumption is that resources which are tagged with important tags by important users become important themselves. Similarly to FolkRank, our approach exploits the link structure between users, resources, and tags, but rather looks at the vicinity of a post (i.e., a [resources,user] pair) in order to compute a weight for the most appropriate tags.

2.2 User and Entity Identification

The issue of identifying users via their interaction over the web has been recently addressed in various application scenarios, such as personalization [VHS09], search, recommendations, building communities of practice (presented in Sections 3.1 and 3.1.3), etc. Hardt[1] with his proposed "Identity 2.0" introduces user-centric environments where the same user identity is shared among a variety of web applications. Applications for social network integration, also known as social network mashup, are already available on the Web, e.g. Spock[2] or 123People[3]. These applications aggregate the information publicly available in different social networks and then provide "real time people search service", whether the users want to or not.

By aggregating the profiles an individual user has at different systems it is possible to create more valuable user profiles [AHHK10] than when considering just the explicitly provided profile information (e.g. name, hometown, etc.) or just the implicitly provided tag-based profiles (e.g. tags assigned to bookmarks). Such enriched information about a user may be exploited, e.g. to overcome the cold-start problem or to enable cross-domain search and recommendations.

Carmagnola and Cena introduce an approach for user identification across application boundaries that bases heuristics on profile attributes such as username, name, location or email address of a user [CC09]. They model the users as attribute-value pairs, which are then compared across systems for user identification across user-adaptive systems. Similar to our study, in [VHS09] the authors examine user profiles from two different social networking websites to find which fields in the profiles are best suitable for user cross-system identification. Unlike our approach where we focus on indirect user profiles, i.e. tags that users assigned to items, they rely on the explicit user profiles. Also, in their work they consider similar social systems (i.e. Facebook and StudiVZ[4]) where the user profiles have the same main structure (e.g. education, website, birthday, gender, etc.). In our analysis we considered social systems where the users are publishing and tagging different kinds of items: photos on Flickr and Web resources on Delicious and StumbleUpon.

Zafarani and Liu experiment with connecting user accounts across 12 diverse communities by exploiting explicit profile information [ZL09]. Given a username in one

[1] http://identity20.com

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

[3] http://www.123people.com

[4] http://www.studivz.net

base community, a Google search is performed and candidate target community usernames are extracted from the returned profile pages, which are identified based on the occurrence of the username in the URL. The set of candidates is then preprocessed, i.e. filtered for common words, and expanded by adding/removing prefixes/suffixes commonly found in the username dataset. Finally, username candidates not existing in the target community are deleted. An average accuracy of 66% is reported, however, from the description the reliability of the ground truth remains unclear. As the authors note themselves same or similar usernames do not necessarily identify the same person. In our work, we guarantee to map real life identities as specified manually by users in their Google profiles.

Focusing on implicit tagging information, Szomszor et al. aim at aligning the tag-based profiles users have in Flickr and Delicious [SCA08]. First, syntactic filtering and stop word removal are applied. Compound names/multi-word phrases without a delimiter and misspellings are resolved by getting spelling suggestions via Google's "did you mean" mechanism. Furthermore Wikipedia is employed to link potential abbreviations, acronyms, etc. to Wikipedia entities. After stemming, synonymous tags are merged via WordNet[5]. In an evaluation on 502 users, alignment between the users tag clouds improves on average about 2% in terms of distinct tags and about 13% for tag assignments. For correlating tag-clouds between the two systems cosine similarity is measured for raw tag profiles and compared to similarity based on aligned profiles. Since the filtering process highlights profile differences to nearest neighbors, it is suggested to employ it for candidate profile selection. However, identifying the right tag cloud for a user is not attempted. Thus, no accuracy or success of this method for the task at hand is known. In this paper we evaluate their suggested approach for identification as a baseline (TF).

On the contrary, recent effort has been made regarding the privacy issues that arise in social networks. Fang and LeFevre propose a template for the design of a social networking privacy wizard in [FL10]. In [WHK+10], the authors present a novel and low-effort user de-anonymization attack that exploits group membership information available on social networking sites. They show how, for Xing[6], a medium-sized social network, 42% of the users who use groups can be uniquely identified. A related approach for enriching user profiles based on network connectivity is presented in [MVGD10], where they predict user attributes based on the attributes of other users, in combination with the social network graph.

More generally, the problem of identifying users can be regarded as an instance of duplicate detection – also known as record linkage or entity resolution – a long standing problem in computer science. Initially, duplicate detection was mainly employed for deduplicating census data (see e.g. [Jar89]) in an era, when data about people were scarce and needed to be acquired explicitly. Ever since it has been applied to a wide variety of domains. There exists a large body of work on matching

[5] http://wordnet.princeton.edu

[6] http://www.xing.com

resources using literal content in the form of structured records (e.g. [LF06]) or of bag of tokens [CRF03]. The common theme for these approaches is to weight the matching evidence provided by content based on its uniqueness and similarity. For a more comprehensive overview on general duplicate detection see [EIV07]. In a sense the approaches and experiments described in this paper return to the very root application domain of duplicate detection – identifying individuals – though under quite different circumstances. By tagging and other forms of interactions, Web 2.0 users provide a rich but fairly noisy trace, which as we will show can be readily exploited for identifying them.

2.3 Entity Search on Wikipedia and on the Web

Finding entities instead of just documents on the Web is a recent topic in the field of Information Retrieval. The first proposed approaches [BCSW07, CC07, CYC07] mainly focus on scaling efficiently on Web dimension datasets but not much on search quality.

Approaches for finding entities have already been developed in the Wikipedia context. For example, Pehcevski et al. [PVT08] use link information for improving effectiveness of ER in Wikipedia. The authors of [DFIN08] improve ER effectiveness by leveraging on a highly accurate ontology for refining the search on the Wikipedia category hierarchy. Compared to these approaches, we propose, in Section 4.1, an orthogonal view on the problem that can be applied to any ER approach via relevance feedback.

A different approach to the problem is to rank document passages that represent entities. In [ZRM$^+$07] the authors present an ER system that builds on top of an entity extraction and semantic annotation step followed by running a passage retrieval system using appropriate approaches to re-rank entities.

A related task is the *entity type ranking* defined in [VZ08]. The goal is to retrieve the most important entity types for a query, e.g., Location, Date, and Organization for the query Australia. Our algorithm also uses entity type information and the entity-category graph in order to find the most important entity types. Moreover, we apply it to improve the effectiveness in the ER task. Another related task is *expert finding* which has been mainly studied in the context of the TREC Enterprise Track [BdVCS07]. In this case the entity type is fixed to people and the query is finding knowledgeable people about a given topic. Entity ranking goes beyond the single-typed entity retrieval and relevance is also more loosely defined.

An important related area of research is entity identity on the Web. It is crucial for the ER task, being able to globally identify entities on the Web so that the search engine can return a list of identifiers to the user who can afterwards navigate the result descriptions. A strong discussion already started in the Web research community [BSTH07]; solutions for entity identity resolution on the Web have been proposed [BSB08]. In this thesis we consider Wikipedia URLs as identifiers, following the approach taken at the Entity Ranking Track at INEX (see [dVVT+08] and Section 4.1.3) as well as one of the proposals of [BSTH07].

Because we assume each entity to be represented by its Wikipidia page, approaches for Entity extraction and resolution (see, e.g., [McC05]) are not necessary for our goal of ranking entities as response to a user query while they would be very relevant in case of performing the ER task on other collection in order to create profiles (to be then ranked) by aggregating knowledge about identified entities.

The first proposed approaches for finding entities on the Web [BCSW07, CC07, CYC07] mainly focus on scaling efficiently on Web dimension datasets but not on the effectiveness of search. In more detail, the authors of [CC07] tackle the ER task with a two component approach: one for extracting entities from the web and one for querying the database containing the extracted entities. Their broad notion of entity (e.g. a "pdf" is considered an entity) allows to integrate various data types into the search engine. A drawback of the approach – besides the missing evaluation – is that the user has to explicitly state their information need by defining the type of the results they are looking for. In [CYC07] an evaluation for the above sketched system is given. The test queries are limited to queries for phone numbers and email addresses. To retrieve them, each entity finding in the extraction phase is assigned a confidence score which is summed up with other findings of the same entity in different documents. A semantic search engine based on SPARQL queries, an optimized index structure, and an ontology is described in [BCSW07]. During indexing time, any occurrence of an instance of the ontology is annotated with the class and with relation information. The so gained data structure allows to answer SPARQL queries efficiently. The system is implemented using YAGO([SKW07]), a Wikipedia and WordNet based ontology, and Wikipedia itself as a corpus. The main differences of the above mentioned systems to our approach are that the user has to follow certain rules for querying the system; either stating the entity type that he is looking for or even some more complex structure requirements to transform the query into a SPARQL representation. We do not make any assumptions about the user query facilitating the interaction considerably. We also do not limit our system to certain entity types and use the Web as a corpus instead of e.g. Wikipedia.

In the wake of the INEX[7] challenge a couple of systems were presented to solve Entity Ranking in the Wikipedia context. More or less explicit user queries in natural language had to be answered with a ranked list of entities. This limited setting compared to Entity Search in the Web improves accuracy of the systems and allows to evaluate the competitors automatically. Different strategies were used by the participants: The authors of [PVT08] use link information on the Wikipedia pages; [DFIN08] make use of the category information present in Wikipedia and incorporate an ontology to improve effectiveness; [DFI+08] use Natural Language Processing techniques; [VTP08] leverages user provided example entities. A probabilistic frame-

[7] http://www.inex.otago.ac.nz/

work for ER is proposed in [BBdR10]. More approaches for the Wikipedia setting are described e.g. in [dVVT+08].

Our Entity Ranking algorithm, presented in Chapter 4, exploits graph structures. Session Graphs or Click Graphs were previously used beneficially in various tasks. In [KT09] the authors perform an analysis of web search query logs and user activities concluding that 50% of queries are about entities. A probabilistic approach for named entity recognition in queries is presented in [GXCL09]. In [CS07] the authors describe how to use a Click Graph to improve Web search. They apply a Markov random walk model on a large click log, producing an effective probabilistic ranking of documents for a given query, including also relevant documents that have not been clicked on for that query. Click logs and random walks have also been made use of for other retrieval tasks, such as query expansion [CTC05] or finding closely related queries [JRMG06]. In [CW07] session data of users is used to make query suggestions. User session information is also used in [ABD06] for improving Web search results. A study of a large query log has been done by [BYT07]. They show how most of the relations in a query log can be described with a power-law distribution, e.g., the distribution of click frequencies on query results. Most interesting, they identify semantic relationships only using the query log data. In our work we apply a Markov random walk model on both Click Graph and Session and analyse how the data can be used for answering Entity Search tasks.

3 User Profiles on the Web

This chapter introduces and presents the model for collecting and maintaining and re-using user profiles in Web applications with tagging resources as a main activity. When having well defined user profiles, a system can better understand the user preferences and, for example, provide people with better search results, or recommend them other objects based on tags other people assigned. From the problems presented in the introductory Chapter 1, we address here Problem 1 (*How to build and maintain user profiles given the user's tagging activity?*) and 2 (*How to exploit user profiles for improving the user's experience across social Web applications?*), while Problem 3 (*How to exploit implicit and explicit user tags for improving typed item retrieval?*) and Problem 4 (*How to exploit users behavior for allowing typed item retrieval on the Web?*) will be addressed in Chapter 4.

This chapter deals with the following issues:

1. A model for building user profiles in tagged corpora (folksonomy)
2. Exploiting tag-based user profiles for recommendations in folksonomies
3. Exploiting tag-based user profiles for user identification across social Web applications

In this thesis, we propose to use tagged corpora of objects to create user profiles in domains, where such folksonomies are available. We utilize the folksonomy model as defined by Hotho et. al [HJSS06a]:

Definition A *folksonomy* is a quadruple $\mathbb{F} := (U, T, R, Y)$, where U, T, R are finite sets of instances of *users*, *tags*, and *resources*. Y defines a relation, the *tag assignment*, between these sets, that is, $Y \subseteq U \times T \times R$.

Having well defined user profiles in a folksonomy, the system can, for example, provide people with better search results, or recommend them other objects based on tags other people assigned. One particular problem is the one of recommending relevant tags to users for objects they have introduced in the system. Being able to effectively recommend tags would, firstly, simplify the tasks of the users on the web who want to tag resources (e.g., bookmarks, pictures, . . .), and, secondly, would allow an automatic annotation of resources that enables, for example, a better search for resources or an improved resource recommendation.

We first show in Section 3.1 a complete system - TBProfile, a Web application for creating and maintaining re-usable tag-based user profiles in the domain of digital libraries. One application of the created user profiles is to provide the users with recommendations about related objects, tags or users with similar users, as shown in Sections 3.1.3 and 3.1.4. In the second part of this chapter, in Section 3.2, we show different methods for identifying users across different social Web applications based on their profiles and how aggregating their profiles helps with identifying users in Section 3.2.5.

3.1 Tag based profiles and recommendations

For people in a social Web application or in a community (such as professors and students in the research community), a well-defined profile expressing their current interests is highly valuable. As one main application, such profiles can help to find persons who work on related topics and, thus, help to facilitate cooperation within the community.

For creating user profiles two steps are necessary:

1. Determine the user profile schema, i.e., how the user profile should look like.
2. Determine how to populate the user profiles with actual data for particular users.

Both steps are interrelated: In general, the higher the accuracy of the user profile is, the more data the profile schema comprises, and a large schema in general leads to more complex handling and maintenance of the profiles. Especially the problem of populating user profiles with actual and accurate data is difficult to solve for large profiles as accurate data is mostly based on human inspection.

In this work we propose to use tagged corpora of objects to create user profiles in domains, where such folksonomies are available. The basic idea is to let people create their profiles by specifying the most relevant objects in the folksonomy. Afterwards, this *intermediate profile* comprising the objects is translated into the tag domain, assuming that the manually specified tags describe the objects with a high accuracy. Hence, the representation of the *final user profile* is based on the tags of the most relevant objects. This has the advantage that users only have to specify comparatively few objects to generate a reasonably large user profile. Furthermore, it is easier to find related user profiles as tags are typically shared by several objects.

We apply our approach to the domain of digital libraries, using a subset of the DBLP data set as object corpus, which has been enhanced with ‘tags’, e.g., the keywords that were manually specified by the authors of the publications. The resulting user profiles, generated by our prototypical *TBProfile system*, are represented by keyword vectors and are exported to RDF (as already proposed in the eLearning domain [DAD05]), so they can be reused in other domains with similar tags. The TBProfile system uses standard recommender system technology on these profiles to recommend other publications, other relevant keywords (for refining the user profile), and finally other relevant persons. These persons, being relevant for the user, are potential candidates to collaborate with and, thus, to be added to the user's Community of Practice, for example.

3.1.1 A Tag-Based User Profile Generator

This section presents our approach to creating and maintaining user profiles. The basic idea is to relate a user with a set of tagged objects and store them in an intermediate user profile. The final representation of the user profile is based on the tags associated with the objects. An example set of objects (publications from the Semantic Web domain) forming an intermediate user profile is shown in Table 3.1.

Publication title	Tags (Keywords)
Magpie: supporting browsing and navigation on the semantic web	named entity recognition (NER), semantic web, semantic web services, ...
Bootstrapping ontology alignment methods with APFEL Swoogle: a search and metadata engine for the semantic web	alignment, mapping, ontology, ... rank, search, semantic web, ...

Table 3.1 Example: Intermediate user profile comprising a set of tagged publications

A user having selected only these three publications will be described by the final user profile shown in Table 3.2.

Occurences	Tags
2	Semantic Web
1	NER, SW Services,Alignment, Mapping, ontology, rank, search

Table 3.2 Example for the final representation of a user profile

Using the tags in the user profile has several advantages:

- A more accurate description of the user's interests based on the content of the selected objects.
- A denser population of the user profile, i.e., less non-empty values (assuming that the objects are on average tagged with more than one tag). This approach can be extended to adding those tags to the user profile, which are clearly subsumed by another tag (such as 'RDF' being a sub-topic of 'Semantic Web'). These can automatically be derived, for example, using the GrowBag approach [BTD06] and can further reduce the sparsity of the user profile.
- A lower dimensionality of the user profile if the number of tags is smaller than the number of tagged objects. For this purpose, a controlled dictionary [Seb02] can been derived from the set of all tags. As tags are typically power-law distributed [JMH+08], removing the rarely-used tags can reduce the dimensionality of the user profiles by several orders of magnitude (in our experiments, 8600 tags out of 130,000 represented 60% of all occurrences of tags).
- A higher connectivity among the different user profiles as the user profiles are more dense and because the tags in folksonomies tend to be power-law distributed.

In our approach we want to support several different ways of creating user profiles starting from a corpus of tagged objects:

1. Search or navigate through the set of available tags, selecting a subset of the most interesting ones to be able to present the objects associated with this subset of tags, from which the user can select the most interesting ones. This can make use of automatically derived relations between tags as proposed in the GrowBag approach [BTD06].
2. Browsing through the set of objects already existing in the user profile, adding / deleting objects and / or single tags.
3. Browsing through the list of recommended objects (such as publications or persons in the publication domain) and tags and adding the most interesting ones to the profile.

Each user has the possibility to individually modify her profile by adding new objects or removing objects the user is no longer interested in. Also, it should be possible to mark certain topics as 'not interesting': If an object has been tagged by several persons, not all the tags of an object may describe the interests of one particular person. In the publication domain, for example, this means that not all the keywords of a publications with several authors may be relevant for the interests of one particular author; the non-relevant keyword might be referring to a part of the publication written mainly by another co-author.

The tags are typically gained using a manual ‘tagging’ approach (e.g., in the publication domain, the authors already provide a set of keywords describing their publications). Alternatively, keywords can be retrieved using Information Retrieval methods, for example, from the title, the abstract, or the full text of the publication, though they are typically of lower quality.

3.1.2 Approaches to Creating and Maintaining User Profiles

Which of the three earlier mentioned ways of creating user profiles are best suited for a particular user strongly depends on the type of user: For users without a profile, we first try to bootstrap a user profile based on the tags, the user herself has contributed to the folksonomy system (if existing). While this is easy in general folksonomy systems, problems arise in the publication domain because of missing user ids. Hence, it is necessary to match the user name with the names of all authors in the publication dataset and present a list of papers, where the author names match the user name. The user can subsequently process this list to eliminate publications from other authors having the same name.

If a new user has not tagged any objects herself, she can alternatively search the set of available tags to find those tags which best describe her interests. They are used as a conjunctive query to identify a list of potentially interesting publications. To accommodate too large / too small result lists, tags can be added / removed on-the-fly to get a reasonable size of the result list. Tag hierarchies as generated by the GrowBag system [BTD06] can be used to easier navigate through related tags.

After having selected a set of tags, a user can preview and browse the current intermediate user profile comprising the list of objects that are annotated with these tags, adding interesting objects to the user profile or deleting those objects, which are no longer interesting. This also means that the tags associated with this object are added to or removed from the final tag-based profile. This approach enables an automatic assignment of cardinalities in the user profile. For example, if a user has selected five objects as interesting from which three are tagged with ‘Semantic Web’, the cardinality of the tag ‘Semantic Web’ in the user profile will be three. In contrast, if the user chooses the interesting tags directly, she would have to assigned the cardinalities manually.

Based on the user profile, the system can also recommend other possibly interesting items or even related tags (cf. Sect. 3.1.3). They can be used to further extend and refine the user profile, in case the user agreed with some part or with all recommendations. This is especially useful for people who already work in their community for quite some time and want to monitor the dynamics of the community.

After the user has finished editing her profile we want to export the profile in the RDF format (similar to a FOAF file) which the user can put on her homepage. This allows for an easy exchange of user profiles within a community. Furthermore, other tools can be used to change and maintain the user profile and re-introduce it again to our system later. Hence, we export both the tag-based user profile and also the collection of objects on which the user profile is based. For this purpose, we need unique identifiers for the objects, such as a URL. Moreover, users can also directly view their profile with any RDF viewer and see how their interests overlaps with their colleagues.

The TBProfile System

The TBProfile system applies our ideas to the digital library domain, where the tagged objects are publications and the tags are the keywords, manually annotated by the authors of the publication.

We have used the DBLP collection of around $650,000$ computer science related publications, providing the URLs for about $330,000$ of the publications. As described in [BTD06], all manually annotated keywords were extracted from the provided URLs using a wrapper-based approach. From about 53.000 URLs, proper tags could be found, resulting in a ‘folksonomy’ of tagged publications with around 130,000 popular unique tags. All tags were post-processed using acronym replacement (e.g., WWW → World Wide Web) and Porter stemming and the tags which were mentioned less than five times were filtered out. This resulted in a controlled vocabulary of about $8,600$ ‘main’ tags, representing 60% of all occurring tags due to the power-law distribution of tags.

The TBProfile system comprises also a web application which allows the users to select tags from the controlled vocabulary of tags, either by browsing the set of available tags or by starting from the set of defaultly assigned publications and using the recommender system. For the selected tags, a user can search for publications and select the ones relevant to her current interests. When the user has finished editing her list of publications, she can view her profile and get recommendations about other publications, tags, and persons.

As an example, Table 3.3 shows the tag-based profile of an example user , which has been gained only using his publications available in our tagged DBLP collection.

The column ‘Occurrences’ denotes the number of times the keyword appears in the profile and ‘Global Frequency’ represents how many times the keyword appears in all publications of the community.

Additionally, we also want to let the users explore different sources for the tags assigned to an object. In the digital library domain, this can be, for example, keywords derived from the publication title, or keywords derived from the abstracts. While manually created keywords usually have a very high quality, using keywords extracted from the title / the abstract leads to a larger set of tagged documents for the case

Keyword name	Occurrences	Global Frequency
XML	1	554
UML	1	302
Web services	1	193
Ontology	1	158
Adaptation	1	102
Semantic Web	5	190
Peer-to-peer	4	123
Personalization	4	92
Standards	1	61
Query languages	1	63
Hypermedia	1	93
Generalization	1	25
Web search	1	49
E-learning	1	59
Network management	1	49
Diagnosis	1	49
Ranking	1	31
Pagerank	1	38
Web engineering	1	35
Adaptive hypermedia	2	30
Meta-modeling	1	9
XML scheme	1	23
XMI	1	9
Asynchronous collaboration	1	8
Synchronous collaboration	1	5
Adaptive Web	2	5

Table 3.3 Example of a tag-based user profile

that not all documents were manually tagged by the authors.

3.1.3 Using Tag-Based Profiles for Recommendations

One application of the created user profiles is to provide the user with recommendations about related objects or tags (i.e., to use in regular search engines), and related users with similar interest, who are candidates for collaborations. The main intention is to deeper analyze the research community.

Basic Idea The basic idea is to use the tag-based profiles as input to standard recommender system technology [RV97], to be able to recommend related objects, tags and persons. Hence, we combine the 'user profile' aspect of collaborative filtering systems with the feature-representation aspect of content-based systems. This means, we combine the idea of letting users ‘recommend’ items, which is a different interpretation of users tagging objects, with the characteristics of legacy information retrieval systems and the derived content-based recommender systems, where objects are represented by their features, typically a vector of terms.

The TBProfile system comprises a user-item recommender system, that computes similarities between users based on a cosine function, that has been extended with the concept of an ‘inverse user frequency’ [BHK98] as the analogue concept to TFxIDF in the recommender system domain. The similarity between two users $U1$ and $U2$ is computed as shown in Eq. (3.1)

$$cos\_iuf(U1, U2) = \frac{\sum_i v_{U1}(i) * iuf(i) * v_{U2}(i) * iuf(i)}{\sqrt{\sum_k (v_{U1}(k) * iuf(k))^2 * (v_{U2}(k) * iuf(k))^2}} \tag{3.1}$$

with $v_U(i)$ being the normalized ‘vote’ of user U for the item i, and $iuf(k)$ defined as shown in Eq. (3.2)

$$iuf(k) = log(\frac{\text{number of users}}{\text{number of votes for k}}) \tag{3.2}$$

As an example, for a user $U1$ having selected three publications for her profile with in total 10 distinct keywords K_{U1}, $v_{U1}(i)$ will be 1/10 for $i \in K_{U1}$.

The neighborhood N_U for each user U is computed using the k-nearest neighbor approach [SKKR00] with $k = 20$. Finally, we compute the recommendation for a certain item I by aggregating the votes of all neighbors of U in a similarity-weighting [HKR02] approach according to Eq. (3.3)

$$rec(U, I) = \frac{\sum_{j \in N_U} v_j(I) * cos\_iuf(U, j)}{\text{neighborhood size}} \tag{3.3}$$

The neighborhood size can at most be k, but may be smaller if only very few similar users are found for the given user U.

Our system can provide several kinds of recommendations:

1. Objects based on users.
2. Users based on objects.
3. Users based on co-tagging.
4. Tags based on users.
5. Users based on tags.

In the first case, the recommender system uses a standard user-object matrix to be able to recommend related objects (e.g., publications in the digital library domain [MAC+02]). In the second case, the matrix is transposed to be able to recommend users instead of objects. This is one variant to get information about other users in the community. In the third variant, the recommendation is based on a matrix of users having tagged the same objects. This can also be used to get information about people in the community. The fourth case is the first one, where we actually use the tag-based user profiles to create a user-tag matrix and finally recommend tags for the users in that matrix. By transposing this matrix, we are able to recommend users based on the tags users have annotated, which is the last variant described here.

Evaluation

Our TBProfile application can give recommendation for publications, keywords and other users of the system. For our experiment we have selected the top 60 authors who have published publications with the topics “semantic web” and “OWL”. For these authors we have built their profiles based on the keywords of the papers they have authored. The intermediate profiles comprised on average 34 publications while the number of keywords per authors was only 16 due to the fact that only 20% of the publications in our database are tagged.

For the profile from Table 3.3 we show the recommendations in the following tables regarding recommended authors. We only provide the user with at maximum the top ten results.

Table 3.4 is the result of case 3 for the user in Table 3.3 , i.e., based on a co-author matrix. These recommendations clearly focus on the ‘senior’ people, having long lists of publications. In this recommendation, tags have not been used at all. In contrast, the recommendations based on the tags (cf. Table 3.5), are based on the content and are not related to the number of publications. Hence, also ‘junior’ people are recommended by our main scheme. For comparison, we also show the result of case 2 in Table 3.6, where we use the transposed user-publication matrix to recommend users. We can see, that only four persons can be recommended here,

Recommended author	score
Rudi Studer	0.0512828
Dieter Fensel	0.0362056
Ian Horrocks	0.0238108
Peter F. Patel-Schneider	0.0221371
Raphael Volz	0.022023
Alexander Maedche	0.0183598
York Sure	0.013157
Timothy W. Finin	0.0268965
Nenad Stojanovic	0.00993426
Enrico Motta	0.00619568
Daniel Oberle	0.0060706

Table 3.4 Recommendations based on coauthorship

Recommended collaborators	score
Steffen Staab	0.390822
Axel Polleres	0.311705
York Sure	0.299058
Siegfried Handschuh	0.253242
Nigel Shadbolt	0.214939
Dieter Fensel	0.21334
Ruben Lara	0.206428
Yuan-Fang Li	0.193029
Bijan Parsia	0.187487
Carole Goble	0.17375

Table 3.5 Recommended collaborators based on keywords

Recommended collaborators	score
Siegfried Handschuh	0.411228
Rudi Studer	0.274152
Dieter Fensel	0.137076
York Sure	0.137076

Table 3.6 Recommended collaborators based on publications

for other users of the system this list of recommendations was even empty. This is because the user-publication matrix is in general less connected than the matrix based on the tags as people tend to share tags and use some of them very often (the ‘stars’ in the power-law distribution).

3.1.4 Time based Tag Recommendation using Direct and Extended Users Sets

Another useful type of recommendation is tags to resources. Being able to recommend tags to resources would help, firstly, with enriching the resource description, and also it can make it easier for the users to tag resources of interest.

When we want to assign a tag to a resource (or, to predict which tag a user would assign to a resource) a possible approach is to use the most popular tags for the given resource of the given user. Of course, this is not working well because users can tag resources which are different and people tag the same resource in different ways. For this reason most effective approaches look at the content of the resources and perform more complex analysis of the structure connecting users, resources, and tags.

Previous approaches focus on the content of resources (e.g., textual content of a web page) or on the structure of the tripartite graph composed of users, resources, and tags. The approaches we propose in this paper do not take into account the content of the resources but only the connection structure in the graph. Additionally, we put more importance on more recent tags with the assumption that users' interests might change over time.

We adapt an algorithm proposed for ranking entities in Wikipedia [CDGI08] based on a set of initial relevant examples (e.g., already tagged resources) and on the structure of hyperlinks connecting pages and categories containing them. As we defined hard links between documents and categories they belong to and soft links between documents and categories containing linked documents, so we define these types of links between resources/users and tags in the tag recommendation setting.

Graph Based Algorithms

In this section we describe the algorithms we designed and used for the graph based task that have been run at Discovery Challenge (DC) 2009, formally described in [HJSS06b].

Using the Resource-User Graph In both submitted approaches, starting from the input *query post* (i.e., the input posts from the test file) we retrieve the resource it refers to. We call this resource the *query resource*. For the query resource we retrieve, using the train data, all the users that have annotated it in different posts. We call this set of users the *direct user set*. We then use this set of users as an input for the algorithm and retrieve all tags the users have assigned. In the second algorithm, in addition to the set of direct users, we also retrieve the user neighborhood (i.e., users that used at least once a tag in common with the given user). We then use the reunion of the two user sets as input for recommending tags. We call the reunion of the two user sets, the *extended user set*. As a third approach we retrieve just the tags that have previously been assigned to the resource as baseline for comparison.

As seen in Figure 3.1, by traversing the post - resource - users graph, we obtain the set of direct users that have annotated the resource given in the query post. The extended user set is obtained by adding also the neighborhood users to the direct user set, see Figure 3.2. We considered two users as being neighbors if they had common tags.

As a baseline approach we considered the recommendation of the most popular tags for a resource, where we only kept the tags assigned by the *direct users* to the resource of the query post.

Comparison to the Wikipedia scenario The algorithms described in this paper are adapted from those developed for finding relevant results for Entity Retrieval queries in the Wikipedia Setting [CDGI08]. This work was performed in the context of the Entity Ranking track at the evaluation initiative INEX 2008 [DdVIZ08] and is explained in more detail in Section 4.1. In the following section we describe how we

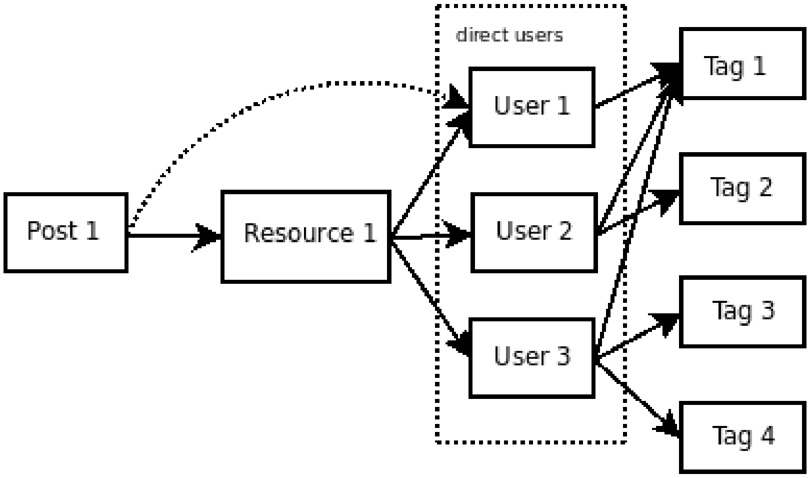

Figure 3.1 Tags recommended based on the set of users who have annotated the query resource.

can map the Entity Ranking setting with the tag recommendation one.

In the Wikipedia setting we have as input a set of example entities. The goal is to extend such set with other relevant entities. If, for example, the initial set for the query “European Countries” contains Italy, Germany, and France, then the goal is to extend this list with entities such as Spain, Slovenia, Portugal, . . . Our approach is to retrieve other entities based on common assigned Wikipedia categories. We extract two sets of categories, hard categories as direct categories (similarly to the *direct user set*) and soft categories from the neighboring entities (i.e., following hyperlinks between Wikipedia articles). As neighboring entities we considered the most frequent entities the example entities linked to (similarly to the *extended user set*). In the Wikipedia setting entities link to entities via hyperlinks, and each entity has several categories assigned to it.

Time dependent tag ranking Following the intuition that tags can get outdated over the years, and, thus, older assigned tags should be weighted less for recommendation, we introduced a time decaying function for posts. Scores are assigned to posts based on the time when they have been issued compared to the time the latest test post has been issued. The time decaying function is defined by the following formula:

$$postScore_i = \lambda^{\Delta Time_i} \tag{3.4}$$

with the decaying factor lambda being smaller than 1 and the time difference being calculated in years. The tag scores are computed based on the tag specificity (i.e., how often they have been assigned) defined as:

$$tagSpecificity_i = \log(50 + tagCount_i) \tag{3.5}$$

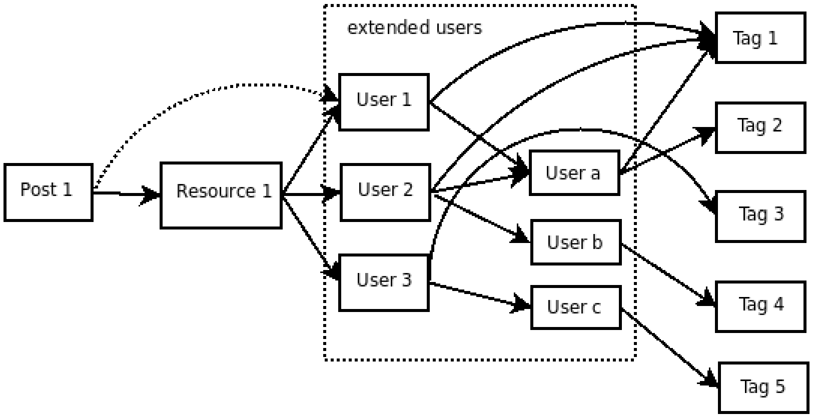

Figure 3.2 Tags recommended based on the set of users who have annotated the query resource and users in the immediate neighborhood of the *direct user set*.

Given the different user sets for a query post, we extract from the training data the most frequent common tags the users have assigned. The tag score is computed based on the formula:

$$tagScore_i = \frac{\sum_j (postScore_j)}{tagSpecificity_i} \tag{3.6}$$

where a post j was considered only if it was posted by one of the users from the *direct user set* for the first approach and from the *extended user set* for the second approach. The tags are sorted based on this score and the top five tags are kept and recommended.

As a baseline, we ranked the tags based on popularity within the resource (i.e., how often a tag has been assigned to a resource) also keeping into account when they had been assigned to the resource, based on the formula:

$$tagScore_i = \sum_j (postScore_j) \tag{3.7}$$

3.1.5 Evaluation

Experiments were performed on the DC 2009 benchmark[1] in order to evaluate the proposed tag recommendation algorithms.

Starting from the query posts in the test file we recommended for each post the top five tags using the two described approaches and the baseline. In Table 3.7 we present the results for the first approach, where we used only the *direct user set*. In

[1] http://www.kde.cs.uni-kassel.de/ws/dc09

Table 3.8 we present the results for the second approach, where we used the *extended user set*.

Number of Tags	Recall	Precision	F-measure
1	0.026747552647295558	0.09511568123393316	0.041753556185248966
2	0.05703121500036672	0.10154241645244216	0.0730397271135134
3	0.083617496470967	0.09511568123393316	0.08899674075110432
4	0.10989880941680434	0.09318766066838047	0.1008556892588314
5	0.13078217483744478	0.08868894601542453	0.10569894753226992

Table 3.7 Effectiveness values for *Direct user* approach (λ=0.9).

Number of Tags	Recall	Precision	F-measure
1	0.02737691828308793	0.09125964010282776	0.04211868151994092
2	0.04984503499927664	0.08740359897172237	0.06348530143810115
3	0.06638265690322252	0.07926306769494446	0.07225331251442199
4	0.08992130762696322	0.07872750642673522	0.08395292150525424
5	0.11024156862845816	0.07609254498714679	0.09003785037043334

Table 3.8 Effectiveness values for *Extended user* approach (λ=0.9).

In the Tables 3.9, 3.10 and 3.11 we measure the impact of using the time information when recommending the most popular tags for a resource. With a value of 0.9 for λ, in the time decaying function, the scores were slightly lower than when using just the popularity information (Table 3.9). When using a value of 0.95 for λ, there is a small improvement over the baseline (see Tables 3.9 and 3.11). We ran experiments also with values smaller than 0.9 for λ, and the precision and F-measure decrease quite a lot (with 3% for F-measure for λ of 0.1).

Number of Tags	Recall	Precision	F-measure
1	0.14737576405314196	0.4190231362467866	0.21805782047815803
2	0.23379593842318802	0.3476863753213368	0.2795877380061556
3	0.28613252999499794	0.3048414738646105	0.2951908598165891
4	0.3278925837087792	0.27420736932305056	0.2986566012722124
5	0.3669509047079738	0.25820479862896273	0.3031196354799055

Table 3.9 Effectiveness values for *Most Popular Tags per Resource* (baseline) approach.

3.1.6 Discussion

Having a well-defined user profile can be very helpful, especially in research communities where people are explicitly interested in finding out firsthand about what

Number of Tags	Recall	Precision	F-measure
1	0.14554572105985988	0.4087403598971722	0.21465597802775144
2	0.23093200715308707	0.34640102827763497	0.27711937419401894
3	0.28312374215330527	0.3018423307626398	0.2921835442517161
4	0.3249527001133686	0.2738860325621251	0.2972419816826678
5	0.3644761479594383	0.25820479862896284	0.3022719449006498

Table 3.10 Effectiveness values for *Most Popular and Recent Tags per Resource* (baseline) approach using the time dependent function with $\lambda = 0.9$.

Number of Tags	Recall	Precision	F-measure
1	0.1432448014710226	0.41131105398457585	0.21248777626720058
2	0.2327090040778989	0.3496143958868895	0.2794269228509583
3	0.28266668350215923	0.2997000856898032	0.29093428316589054
4	0.3293738395280813	0.2764567266495287	0.3006042237004772
5	0.36806287828138706	0.2600042844901452	0.3047378719582792

Table 3.11 Effectiveness values for *Most Popular and Recent Tags per Resource* (baseline) approach using the time dependent function with $\lambda = 0.95$.

happens in their line of work. No matter if people are interested in finding new relevant publications, related topics or about people to collaborate with, their user profile can support the information flow in their Community of Practice. The main contributions of the work done in the first part of this chapter can be summarized as fallows:

- We proposed a new model for creating and maintaing tag based user profiles in social bookmarking systems on the Web.

- We use the tags from a folksonomy system to build user profiles and feed them to a recommender system, especially to identify related persons in the community. This unique combination of the user profile aspect of collaborative recommender systems with the feature-based schema to describe user profiles (as used in content-based recommender systems) is intended to better capture the interests of the users in the recommendation process and to reduce problems with sparse user profiles.

- We have shown the TBProfile prototype, implementing a rudimentary system for creating tag-based user profiles in the digital library domain and using a user-item based recommender system to find potential people to extend a user's community of practice. Even though only 20% of the publications in our database are tagged, we have shown evidence that using tag-based profile can give more recommendations than standard object-based user profiles.

- In the second part of the section we presented our first approaches for tag recommendation using graph information. We proposed two approaches, where, given a query post, we retrieve two sets of users. Based on the tags assigned by users in these sets we recommend new tags. The first set of users, the direct user set, consists of the users that have tagged the resource referred to by the query post. The second set of user, the extended user set, consists of the direct user set as well as the users who are neighbors based on commonly assigned tags to the users in the direct set. The tag scores have been computed keeping into account also the time when they have been assigned. With the proposed approaches, we evaluated the effect of the tag posting time. We compared a time dependent ranking to a tag popularity.

3.2 Users profiles in Social Networks

Nowadays, people have online accounts at diverse Web portals where they leave plenty of multifaceted profile data. In particular, with the advent of Web 2.0 users became content providers themselves [O'R05]. Users share their pictures, videos, or bookmarks at platforms such as Flickr, YouTube, and Delicious and annotate these resources using tags to facilitate retrieval of the resources, to express their opinion regarding some resource or merely to present themselves (cf. user incentives described in [MNBD06]). Tagging Web resources has become a popular activity mainly due to the availability of tools and systems making it easy to tag and also due to the advantage users see in tagging their resources. The tags assigned by a user to various objects reflect to some extent the user interests [FNP07] and also extend the object description, thus facilitating re-usability.

In the second part of this chapter we analyze whether the individual tagging practices can be exploited to link the different social media accounts of a user. We analyze profiles of users from three collaborative tagging systems: Flickr, Delicious and StumbleUpon. While the latter two systems are for organizing public Web resources, Flickr is mainly for sharing personal pictures with friends and people rarely tag other people's photos. We introduce and compare various user identification strategies and show that by combining tagging characteristics and string similarities for comparing user ids, we achieve an alarming accuracy of over 60%.

3.2.1 Research Challenge

We investigate the tagging behavior of users across social tagging systems. We analyze how well individual tagging activities can be applied to characterize users and study the following research question: is it possible to identify users across systems based on their profiles? User profiles can be constructed based on implicit and explicit user feedback. With explicit feedback, we refer to the data the user herself provides to the system directly, e.g. during the registration process. Usually such explicit data is structured as attribute-value pairs. In our approaches we experiment with the usernames as explicit profile information. With implicit feedback, we refer to the users' tagging activities within the folksonomy systems. As it is mentioned in the beginning of this chapter, for our research we utilize the folksonomy model as defined by Hotho et. al [HJSS06a]:

Definition A *folksonomy* is a quadruple $\mathbb{F} := (U, T, R, Y)$, where U, T, R are finite sets of instances of *users*, *tags*, and *resources*. Y defines a relation, the *tag assignment*, between these sets, that is, $Y \subseteq U \times T \times R$.

Given the folksonomy model we can define the research challenge investigated in this section as follows.

> **User Identification Challenge.** Given u_a, the tag-based profile and/or username of user X in system A, and U_B, the set of profiles from system B, the challenge of the user identification strategies is to rank the profiles from system B so that $u_b \in U_B$, the profile of X in system B, appears at the very top of the ranking.

Identifying users based on their implicit profiles has not been been studied extensively in literature yet. Considering the dissimilarity of the analyzed systems (e.g. Flickr *and* Delicious and StumbleUpon) having a success rate of 30% for identifying users based only on their tagging activity is a promising result, given the success of 13% or 20% for the standard baseline approaches TF and TFIDF respectively.

The section is organized as follows. In the subsequent subsection we report on related work. Subsection 3.2.3 provides a description of approaches that are used for identifying users across different systems based on their tagging behavior. Subsection 3.2.4 focuses on approaches used when taking also usernames into account for identification. In Subsection 3.2.6 we present the dataset which we used in our analysis, we give an overview of the metrics we used for evaluation and present the experimental results.

3.2.2 Matching Users across Sites

In the following section we present the approaches we used for identifying users across social systems. We present user matching approaches based on different types of user profiles: first, in Subsection 3.2.3, we focus on the implicit user profiles extracted from the user tagging activity; then in Subsection 3.2.4 we describe approaches for explicit user information, such as username, which is a mandatory field in all systems. In Subsection 3.2.5 we present how to combine different types of profiles and also aggregate user profiles to improve cross-system user identification.

3.2.3 Matching Users based on their Tags

For identifying users across social systems based on their tagging behavior, we experiment with standard techniques like TF, TFIDF and BM25 and compare it against approaches based on language models and a new symmetric variant of BM25 using site specific statistics.

Baselines

One of the most straightforward approaches to match tag user profiles exploits tag frequencies [SAC$^+$08]. We evaluate this approach as one baseline (further called TF). However, this approach does not take into account the specificity of tags. Tags used by many users such as “web” contribute much less evidence for a match than more specific tags such as "NYC". To take into account tag specificity, frequency is typically combined with the inverse document frequency of a tag. Since together with the vector space model it is used as a standard method in Information Retrieval, we evaluate this approach as another baseline (further called TFIDF).

More formally, each user profile u is modeled as a vector, where each dimension contains the TF or TFIDF value of tag $t \in T$:

$$\begin{aligned} u &= (w(u,t_1), w(u,t_2), ..., w(u,t_n)) && (3.8) \\ w(u,t) &= TF(u,t) * IDF(u,t) \\ TF(u,t) &= c(t,u)/|u| \\ IDF(t) &= log\frac{N}{n(t)} && (3.9) \end{aligned}$$

where $c(t,u) = |\{(u,t,r) \in Y : u \in U, r \in R\}|$ is the number of times user u has assigned tag t, $|u|$ is the overall number of tags user u has assigned, $n(t) = |\{u \in U : (u,t,r) \in Y, r \in R\}|$ is the number of profiles that contain tag t, and N is the overall number of profiles.

The matching score of two profiles u_1 and u_2 is then determined by the cosine distance on their weight vectors.

BM25

A well known weakness of the TFIDF weighting scheme is that term frequency is not a very good indicator for the relevance of a term. If a document contains a term 20 times, it is not 20 times as relevant as occurring just once. For matching user profiles based on their tags, this weakness strikes even more. If a user assigns a tag 20 times, this is not 20 times as relevant as assigning a tag just once. Okapi BM25 [JWR00] addresses this weakness by tempering term frequency such that it quickly saturates with a maximum value:

$$\begin{aligned} w(u,t) &= TF(u,t) * \sqrt{IDF(t)} \\ TF(u,t) &= \frac{c(t,u) * (k_1 + 1)}{c(t,u) + k_1 * (1 - b + b * \frac{|u|}{avgU})} \\ IDF(t) &= max(0, log\frac{N - n(t) + c}{n(t) + (1 - c)}) && (3.10) \end{aligned}$$

where $c(t,u)$, $|u|$, $n(t)$, and N are defined as in Eq. 3.9. k_1 is a tuning parameter that determines how strongly $c(t,u)$ influences the weight; for $k_1 = 0$, only binary occurrence is taken into account, for large k_1, the frequency influences the weight almost linearly. b is another tuning parameter that determines the influence of the size $|u|$ of a user profile on the weight. For $b = 0$, the weight does not depend on the size at all, and thus large profiles with many tags will get higher scores. For $b = 1$ the weight is fully normalized by the size of a profile. c is yet another tuning parameter that estimates the prior probability $P(t|rel)$ that a tag t is relevant.

As the BM25 weight vectors are already normalized by document length, we use the Euclidean distance between the vectors as a matching score, rather than the cosine distance. By using the square root of $IDF(t)$ for weights this is equivalent to the standard approach of BM25, which uses the sum of products of the TF components of two term vectors and the IDF component.

Site specific IDF and BM25

Tagging behavior is influenced highly by the site's domain and design choices. People tag differently music items, images or Web resources [BFNP08]. For example, in our experimental data set "tools" is used for more than half of the resources in Delicious, but only few times in Flickr, conversely, "arts" is used very often in Flickr, and rarely in Delicious. Hence, "tools" is a very discriminative tag when matching against Flickr, while "arts" discriminates well against profiles in Delicious. As a consequence the document frequency of particular tags may differ substantially among the sites. For the same reason of dependency on system design choices like tagging rights, object type and ownership, etc. [MNBD06], also profile lengths may be very different. To this end, we suggest to use BM25 together with a site specific IDF and site specific average profile lengths:

$$\begin{aligned} w(u,t,s) &= TF(u,t,s) * \sqrt{IDF(t,s)} \\ TF(u,t,s) &= \frac{c(t,u) * (k_1 + 1)}{c(t,u) + k_1 * (1 - b + b * \frac{|u|}{avgU(s)})} \\ IDF(t,s) &= max(0, log\frac{N(s) - n(t,s) + c}{n(t,s) + (1-c)}) \end{aligned} \tag{3.11}$$

Thereby, TF takes into account the site specific profile length $avgU(s)$ and IDF takes into account the site specific document frequency of a tag $n(t,s)$. As shown in Subsection 3.2.6 this approach leads to significantly improved matching accuracy.

Language Models

As an alternative to BM25, language models have been proposed [LC01, LZ03]. Rather than estimating the probability of a match $P(M|u_1, u_2)$ given two profiles u_1 and u_2, these approaches estimate the conditional probability $P(u_1|u_2)$, where u_1 and u_2 are language models estimated from the tag profiles. There exists a wide variety of approaches to estimate the language models. In our experiments, we follow the approach described in [ZL04] using Dirichlet smoothing:

$$\begin{aligned} logP(u_1|u_2) &= \sum_{i=1}^{|u_1|} P(t_i|u_2) \\ &\propto \sum_{c(t_i,u_2)>0} log\frac{P_s(t_i|u_2)}{\alpha_{u_2}P(t_i|C)} + |u_1|log\alpha_{u_2} \\ P_s(t_i|u_2) &= \frac{c(t_i|u_2)+\mu P(t_i|C)}{|u_2|+\mu} \\ \alpha_{u_2} &= \frac{\mu}{|u_2|+\mu} \end{aligned} \tag{3.12}$$

where $|u|$ is the overall number of tags in a profile, $P(t_i|C)$ is the collection frequency of tag t_i, μ is a smoothing parameter in the range of 3000 to 10000 to interpolate between the local tag frequency $c(t, u)$ and the collection frequency, and α_{u_2} is a profile specific normalization factor.

The most striking difference of the language modeling approach to BM25 is the way tag frequencies are handled. Whereas BM25 uses a heuristic saturation function (see Eq. 3.10), the language modeling approach uses the log of smoothed probability estimates for the profiles in u_2, but effectively uses the raw tag frequencies from the profile u_1. For ad hoc querying scenarios this asymmetry does not matter much, because short queries rarely repeat a term. However, when matching tag profiles, the query (u_1) and the document (u_2) exhibit repeated terms. Thus treating term frequency in u_1 differently from u_2 appears inadequate for matching profiles by mutual relevance. An adaptation of the language modeling approach to mutual relevance is subject for future work. Another difference is that selectivity of tags is estimated by their collection frequency $P(t_i|C)$ rather than their document frequency. Like with BM25 we have experimented with site specific collection frequencies as well as the overall collection frequency, but for the language modeling approach using the overall collection frequency achieved much better accuracy.

3.2.4 Matching Users based on Explicit Usernames

In this subsection we present approaches for identifying users based on their usernames in different systems. Often this is the only explicit and publicly available user attribute common to various tagging systems. As opposed to profiles based on tag assignments, where we can have more tags assigned by a user; when dealing with the username we have to take into account that a user has only one username per account in a system.

Services such as Flickr or Delicious do not support OpenID and therefore do not enable their users to apply a globally unique identifier that relates their different online accounts. However, people still can (and often do) utilize the same or at least similar usernames for their different online accounts [ZL09]. Hence, a straightforward approach for identifying users across systems is to analyze their usernames. For this approach we make the following hypothesis: The more similar two usernames are the higher the probability that both usernames refer to the same user entity.

There exist several metrics to measure the similarity of two strings (usernames). For our experiments we apply the following metrics.

Exact Match We give a maximum score to identical usernames.

Jaccard We apply the Jaccard similarity metric by representing both usernames by their sets of characters and computing the ratio of shared characters divided by the number of total characters.

Levenshtein The Levenshtein similarity [Lev66] basically describes the minimum number of editing operations (substituting, deleting or adding a character) that are required to transform one character string into another character string.

Smith-Waterman Similar to the Levenshtein similarity, the Smith-Waterman similarity [SW81] measures the costs of aligning two strings by comparing segments of all possible lengths between two strings.

Longest Common Substring (LCS) is a variation to Levenshtein distance by allowing only addition and deletion, not substitution.

3.2.5 Matching Users based on Combined Profiles

Now we present approaches to merge different sources of user information, first by combining implicit and explicit profile information and, second, by aggregating profiles from two systems to map against a third system.

Combining Username and Tags

In order to combine the different types of profiles, tag- and username-based, we use a mixture model:

$$w(u_1, u_2) \quad = \lambda * w_t(u_1, u_2) + (1 - \lambda) * w_u(u_1, u_2) \tag{3.13}$$

where $w_t(u_1, u_2)$ is the normalized score obtained based on the tags the user assigned, as presented in Section 3.2.3; and $w_u(u_1, u_2)$ is the string similarity of the usernames of the two users.

As the BM25 scores are not normalized between 0 and 1, we scale them to the same range as the scores on username similarity by dividing them with the maximum score of all compared user tag profiles between two systems. Thereby the choice of λ indeed reflects the relative importance of the two scores used for matching.

Aggregated Profiles

As shown in Subsection 3.2.6, matching accuracy via tags depends heavily on the number of tags given by a users. Thus, starting from the assumption that one knows the user mapping between two systems, we want to further identify the users on a third system. There are nowadays more and more users who explicitly interlink their profiles on the Web, for example on their Google accounts. This information can be used for further aggregating the information about the user.

We create an aggregated tag-based user profile by considering all the tags the user has assigned in two systems. The tag frequencies are accumulated. We then apply the same comparison approaches for matching users between the aggregated profile and a third system as presented in Subsection 3.2.3.

For creating an aggregated username-based profile from two systems, we consider the two usernames as matching candidates. When calculating the distance between the aggregated username profile and a third profile we select the highest matching username pair:

$$w(u_{12}, u_3) \quad = max(w_u(u_1, u_3), w_u(u_2, u_3)) \tag{3.14}$$

where u_1 and u_2 are the two usernames corresponding to the aggregated profile from system 1 and 2, and w_u is the string similarity measure between two profiles.

3.2.6 Evaluation

In our experiments we evaluate the proposed algorithms presented in the previous subsections with respect to the user identification challenge defined in Subsection 3.2.1. In particular, we investigate the following research questions.

1. Is it possible to identify users across systems based on their tagging practices and/or based on their user ids?
2. Which algorithm performs best for the user identification challenge?
3. Does knowing more about the user improve the identification performance?

Data Set

To investigate the questions above, we crawled public profiles of 421188 distinct users via the Social Graph API[2]. The Social Graph API makes information about connections between user accounts available via Web service. For example, by exploiting Google profiles of users, who explicitly interlinked their different online accounts, the API provides the list of accounts associated with a particular user. We applied the

[2] http://code.google.com/apis/socialgraph/

following strategy to crawl profiles: (1) we used common first names (e.g., *John*, *Peter*, *Mary*, *Sarah*) as search query at Google's profile search interface[3] to obtain profile URIs[4] and perform a Social Graph lookup[5] and (2) we crawled the profiles of friends that were linked by users which were obtained in the first step.

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

The tagging statistics of these 321 users having tag-based profiles at Flickr, Delicious, *and* StumbleUpon are listed in Table 3.12 (*FDS dataset*). Overall, these users performed 387786 tag assignments (TAS). In Flickr users tagged most actively with, on average, 532.99 tag assignments, followed by Delicious (483.58 TAS) and StumbleUpon (191.48 TAS). It is interesting to see that Delicious tags constitute the largest vocabulary although most tagging activities were done in Flickr: the Delicious folksonomy contains 21239 distinct tags while the Flickr folksonomy covers just 18240 distinct tags. Correspondingly, tag-based Delicious profiles have, on average, 66.17 distinct tags in contrast to 56.82 distinct tags for the Flickr profiles.

Accordingly, Table 3.13 lists the tagging statistics of these users who have a Flickr *and* Delicious account, but are not necessarily registered to StumbleUpon. With 387786 tag assignments performed by 1467 users in Delicious and Flickr, this dataset (*FD dataset*) will be applied to confirm the results of our experiments on a larger scale (see Subsection 3.2.6).

Tag Overlap

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

Figure 3.3 shows to which degree the profiles of the individual users in the different services overlap with each other. For each user u and each pair of service A and B, we compute the overlap as follows:

$$overlap(u_A, u_B) = \frac{1}{2} \cdot (\frac{|T_{u,A} \cap T_{u,B}|}{|T_{u,A}|} + \cdot \frac{|T_{u,A} \cap T_{u,B}|}{|T_{u,B}|}) \tag{3.15}$$

$T_{u,A}$ and $T_{u,B}$ denote the set of distinct tags that occur in the tag-based profile of

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

[4] Example Google URI: http://www.google.com/profiles/106144680131189887520

[5] Example Social Graph API lookup request: http://socialgraph.apis.google.com/lookup?q=http://www.google.com/profiles/106144680131189887520

Table 3.12 FDS dataset: Tagging statistics for the 321 users who have an account at Flickr, Delicious, and StumbleUpon.

	Flickr	**Delicious**	**Stumble Upon**	**All**
distinct tags	18240	21239	8663	39399
TAS	171092	155230	61464	387786
distinct tags/user	56.82	66.17	26.99	122.74
TAS/user	532.99	483.58	191.48	1208.06

user u in service A and B respectively. Hence, $|T_{u,A} \cap T_{u,B}|$ is the number of distinct tags that occur in both profiles, u_A and u_B. Figure 3.3 illustrates that the individual Delicious and StumbleUpon profiles have the biggest overlap. However, the overlap is rather small: for more than 50% of the users the overlap of their Delicious and StumbleUpon profiles is less than 20% and there exist only 6 users for whom the overlap is slightly larger than 50%. It is interesting that the overlap is so small, as in both Delicious and StumbleUpon the same type of resources are tagged, probably the tools are used for separate task. Flickr and StumbleUpon profiles offer the least overlap as for more than 40% the overlap is 0%. In summary, the small overlaps of the individual profiles indicate that user identification based on tagging behavior is a non-trivial task. We will show that our algorithms nevertheless manage to succeed in identifying users based on their tagging activities.

In order to better understand the dataset, we also mapped the tags from the three sets of profiles to WordNet synsets. WordNet is a lexical database for the English language and words are grouped into sets of synonyms called synsets. For each tag word in the profiles we retrieved from WordNet the main synset and then compared in Table 3.14 the overlap between the three systems. Most of the multi-word tags are written in one word, out of the 39399 distinct tags less than 1% are explicitly multi-worded (e.g. the space between words is marked with underscore). Only 34% of the tags could be mapped to WordNet categories (i.e. synsets). Out of 8726 distinct synsets only about 16% synsets overlap in more than two systems, which is close to the overlap at tag level (e.g. 20%). After preprocessing tags this overlap may increase. Interestingly however, as reported in [SCA08] extensive preprocessing increased the number of overlapping tags between Delicious and Flickr by only 2%.

Table 3.15 shows the top synsets that are overlapping between at least two profiles or are present just in one type of system. The synsets are ordered based on the

Table 3.13 FD dataset: Tagging statistics for the 1467 users who have an account at Flickr and Delicious.

	Flickr	**Delicious**	**All**
distinct tags	72671	59275	119056
TAS	892378	683665	1576043
distinct tags/user	49.54	40.41	81.16
TAS/user	608.30	466.03	1074.33

Table 3.14 Statistics for overlapping synsets between the various user profiles.

System	**Count**	**Average**
Delicious+Flickr	574	6.57%
Flickr+StumbleUpon	218	2.50%
Delicious+StumbleUpon	504	5.78%
Delicious+Flickr+StumbleUpon	111	1.27%

cumulated frequencies with which the tags belonging to them have been assigned by the users in the compared systems. We can see that the Flickr categories are usually more related to photography (e.g. gray, day, red) than the Delicious and StumbleUpon categories. As presented in [BFNP08], when tagging images and Web data, users tend to describe the topic of the tag item. For images they also focus on location and time information. Similarly, Table 3.16 shows the most overlapping tags among the three social systems in our dataset.

Method and Metrics

Given the data described in the previous subsection, the user identification algorithms have to identify for each user profile the corresponding profile that refers to the same user in another system, i.e. each algorithm is tested in different settings which are given by the different service constellations. For example, (i) given the Flickr profile of user u, the algorithm has to rank Delicious profiles so that u's Delicious profile appears at the very top of the ranking, (ii) given the StumbleUpon profile of user

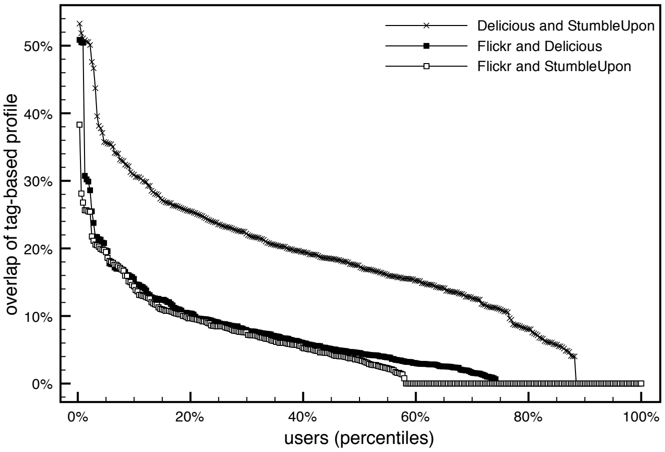

Figure 3.3 How much do the distinct tags of the individual, service-specific profiles overlap?

u, rank Flickr profiles so that $u's$ Flickr profile is ranked as high as possible, etc. To measure the quality of the user profile rankings we use the following metrics (cf. [SvZ08]).

MRR The *MRR* (Mean Reciprocal Rank) indicates at which rank the correct profile occurs on average. Its best value is at 1.

S@k The Success at rank k (*S@k*) stands for the mean probability that the correct profile occurs within the top k of the ranked results.

In case of tied scores between the correct user profile pair and some other pairs, we penalized both metrics by dividing them by the number of tied scores.

Results

Comparing users based on their different profiles types, both individually and combined, for user identification across systems provides some interesting insights.

Matching Users based on Tags

Table 3.17 compares the various approaches for identifying users based on their tag profiles. BM25 clearly outperforms TFIDF, and BM25 with site specific IDF also clearly outperforms BM25 with global IDF. The language modeling approach LM is about in the same range as BM25 with global IDF. This suggests that accounting for site and domain specific characteristics in tag weighting may be more promising than employing sophisticated modeling. All methods yield substantially better results than the baseline approach using TF with cosine similarity suggested in [SCA08], BM25

Table 3.15 Top synsets of tags from different systems.

System	**Synsets**
just Flickr	Buddha, gray, smile, wall, alley, boot, destroy, nap, chair, red, day
just Delicious	rate, use, associate, charge, tip, film, analyze, woo, prize, implant, career
just StumbleUpon	knit, curse, humor, python, humanitarianism, FTP, perplex, humanistic discipline, cartoon, doctrine
just Delicious and Flickr	state, cover, have, weave, color, jump, crop, plan, show, shoot, flip
just Flick and StumbleUpon	peddle, embroider, airplane, school, sail, raise, annoy, pot, packer, cab
just Delicious and StumbleUpon	chop, break, blend, dress, comment, appliance, buy, Muslim, help, classif
Delicious and Flickr and StumbleUpon	travel, be, model, put, visualize, turn, check,ma, fly, tag, capture

with site specific IDF improves its *S@1* by even 2.5 times. All improvements are significant according to a 2-tailed t-test at p-level ¡ 0.05. Comparing our work to the approach described in [SCA08] is difficult, since they do not report any accuracy or success measures beyond tag profile alignment.

The tuning parameters for BM25 were experimentally determined as follows: $k_1 = 3.75$, $b = 1$, and $c = 1$.

Matching accuracy strongly depends on the size of user profiles. The more distinct tags two profiles of a user contain the more likely they can be identified. To investigate this in more detail, we sort the profile pairs by the minimum number of tags in one of the profiles, and compute the running average of Success@1 and MRR (see Figure 3.4). Evidently, there exists a strong correlation between profile size and matching accuracy (0.93). For example, the top 6 profile pairs by size in the range of around 500 distinct tags are all perfectly matched.

In order to better understand the problem of identifying users between systems with different domains we run our experiments also on a subset of 1467 user with profiles both in Flickr (items tagged are photos) and Delicious (items tagged are web resources). Table 3.18 compares the results for the individual distance metrics. The first row gives the results for 321 users (FDS dataset, Subsec. 3.2.6) and the second row for 1467 users (FD dataset, Subsec. 3.2.6). By operating on a larger set of users the chance of a mismatch increases for all metrics. However, the relative ordering is consistent. BM25 with site specific IDF outperforms all other approaches and

Table 3.16 Top tags from the different systems.

System	Tags
just Flickr	january, viapixelpipe, cameraphone, geotagged, catalunya, pcc, lon, warmthinwinter
just Delicious	web2, tweecious, plus, imunitas, filetype, extra, ekstra, imported, susu, immune, faktor
just StumbleUpon	cyberculture, liberties, stumblers, humanitarianism, homemaking, capistrano99, scultping
just Delicious and Flickr	unitedstates, o, yellowknife, de, ir, joey, alizio, gerdleonhard, star, nolimitdomains, barbecue
just Flick and StumbleUpon	1stangel, techzulu, southwestseo, jimatwood, boating, caribbean, skateboarding, jimctc, michigan
just Delicious and StumbleUpon	css, golostra, activism, entrepreneurship, psychology, management, hacking, colostrum
Delicious and Flickr and StumbleUpon	design, video, photography, music, blog, google, art, politics, travel

looking at MRR it is less influenced by the higher number of users and thus potential mismatches.

Matching Users based on Username

Table 3.19 compares the various approaches for matching profiles based on usernames. Smith-Waterman and exact match are clearly not good measures for the username comparison task. Levenshtein and the Longest Common Subsequence based distance perform fairly similar and outperform both Jaccard and Smith-Waterman distances. Note that success rates increase only slightly with increasing k, whereas they increase fairly substantially depending on k for matching based on users' tags. This is to be expected. User names tend to be much more unique than the tags assigned by users. In Figure 3.5 we plotted the performance (1/rank) for each user for the five similarity metrics. For the best metric, string similarity works well for approximately 55% of the users but fails for the other 45%. Though this result is lower than the 66% average accuracy reported in [ZL09], we ensure to identify indeed the same real life entities.

Table 3.17 Matching results based on user tags

Strategy	MRR	S@1	S@2	S@3	S@10
TF	0.181	0.126	0.161	0.180	0.278
TFIDF	0.267	0.207	0.253	0.277	0.380
LM	0.295	0.243	0.287	0.312	0.383
BM25	0.301	0.242	0.292	0.317	0.405
BM25 specific IDF	0.345	0.291	0.337	0.360	0.443

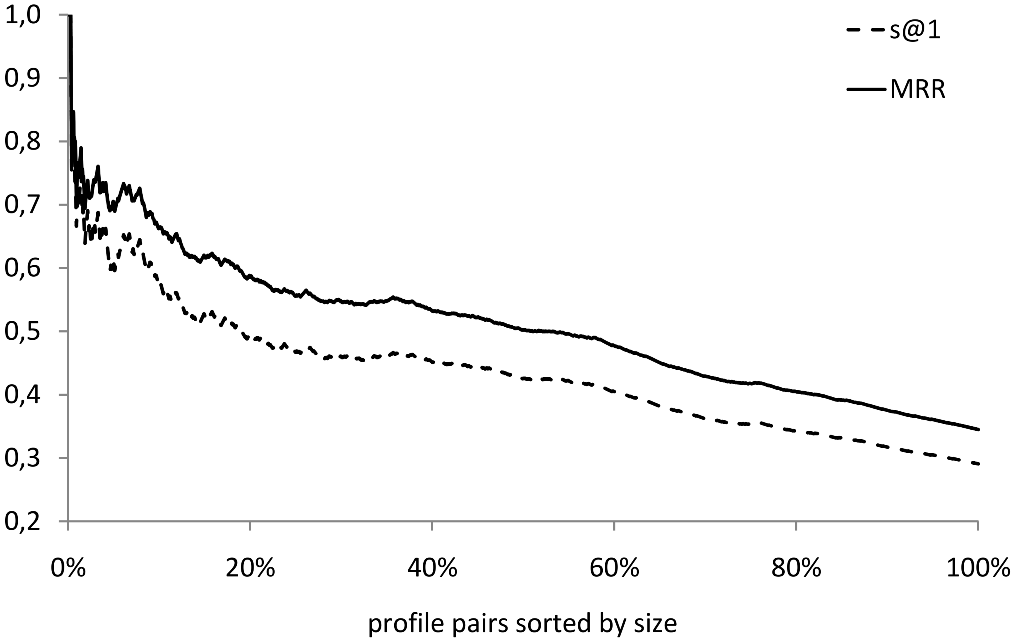

Figure 3.4 Success@1 and MRR vs. profile size

Mapping Users based on Tags and Username

When combining the best performing measures for the two types of profiles (see Subsection 3.2.5), i.e. BM25 with site specific IDF for the tag-based profiles and LCS for the username profile, we gain major improvements as listed in Table 3.20. We achieve an absolute improvement of 35% compared to the approaches that exploit just the tag-based profiles (see Subsection 3.2.3) and of 8.9% compared to the username-based approaches (see Subsection 3.2.4).

In Figure 3.6 we analyze how the user identification strategies perform for the different service settings: all approaches work best when comparing profiles from StumbleUpon and Delicious. Identifying users across Flickr and Delicious/StumbleUpon is not as successful. This result is to be expected considering that type of resources dif-

Table 3.18 Matching results based on user tags between Flickr and Delicious (FD dataset) and accordingly between Flickr, Delicious and StumbleUpon (FDS dataset)

Strategy (dataset)	**MRR**	**S@1**	**S@2**	**S@3**	**S@10**
TF (FDS)	0.181	0.126	0.161	0.180	0.278
TF (FD)	0.108	0.070	0.091	0.110	0.178
TFIDF (FDS)	0.267	0.207	0.253	0.277	0.380
TFIDF (FD)	0.184	0.124	0.168	0.197	0.302
BM25 (FDS)	0.301	0.242	0.292	0.317	0.405
BM25 (FD)	0.259	0.204	0.246	0.274	0.370
BM25 specific IDF (FDS)	0.345	0.291	0.337	0.360	0.443
BM25 specific IDF (FD)	0.343	0.250	0.303	0.330	0.428

Table 3.19 Results based on username.

Strategy	**MRR**	**S@1**	**S@2**	**S@3**	**S@10**
ExactMatch	0.387	0.372	0.375	0.375	0.375
Jaccard	0.535	0.501	0.526	0.536	0.577
SmithWaterman	0.462	0.357	0.371	0.475	0.607
Levenshtein	0.574	0.552	0.567	0.572	0.591
LCS	0.582	0.552	0.578	0.586	0.600

fer between these systems. While the items tagged in Flickr are photos (and videos), StumbleUpon and Delicious are more similar as they both focus on bookmarks. Another remarkable observation is that it seems that many users tend to use similar usernames on StumbleUpon and Delicious as the success of username-based approach is higher than 70%, whereas this approach is less successful ($S@1 < 50\%$) for Flickr profiles. In summary, the results are still impressive, given that we found only a small actual overlap in tags between Delicious and StumbleUpon (less than 20% tag overlap for more than 50% of users).

Using aggregated profiles

Table 3.21 compares the various approaches for matching aggregated tag-based profiles, i.e. the union of tags from two individual profiles for a given user (see Subsection 3.2.5). We then compare each aggregated profile with the remaining profile (e.g. Flickr-Delicious to StumbleUpon, Delicious-StumbleUpon to Flickr, etc.) and vice-versa. Again, BM25 clearly outperforms TFIDF, and BM25 with site specific

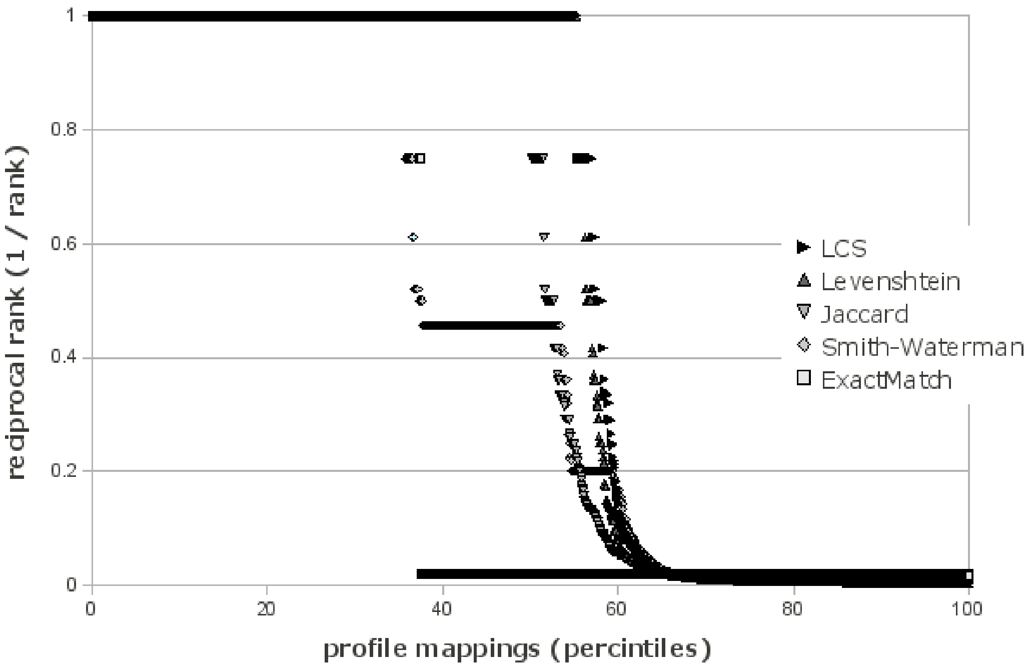

Figure 3.5 Profile mappings based on username, reciprocal rank for the different string similarity distances

Table 3.20 Results when combining tags and username best approaches, with mixture coefficient λ=0.86

Strategy	MRR	S@1	S@2	S@3	S@10
BM25(tags)	0.345	0.291	0.337	0.359	0.436
LCS	0.582	0.552	0.578	0.586	0.600
Mixture	0.677	0.641	0.681	0.697	0.728

IDF also leads to a significant improvement (2-tailed t-test at p-level ¡ 0.05) over BM25 with global IDF. In summary, knowing more (tags) about the user improves the user identification performance clearly. For example, S@1 improves from 0.291 (see Table 3.17) to 0.393 (see Table 3.21) for BM25 with site specific IDF.

Correspondingly, Table 3.22 compares the approaches for matching profiles based on usernames when dealing with aggregated profiles, i.e. the union of two usernames (see Section 3.2.5). Again, we observe that knowing more (username) about the user improves the user identification performance. For example, S@1 increases from 0.552 (see Table 3.19) to 0.701 (see Table 3.22) for the Levenshtein and LCS measures which are again outperforming the other approaches.

Finally, Table 3.23 lists the results for the mixture approach that combines the

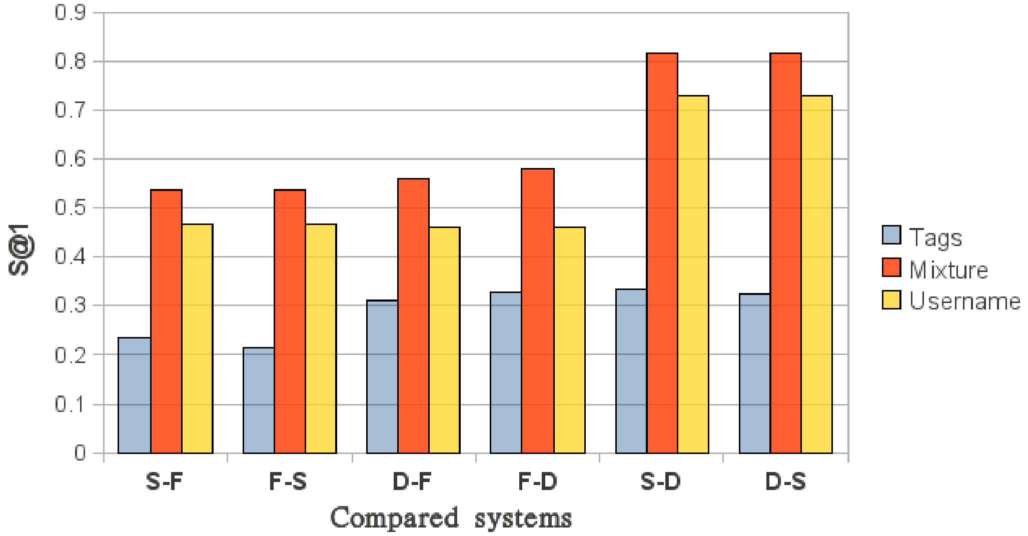

Figure 3.6 Success at rank 1 for each pair of systems. S stands for StumbleUpon, F for Flickr and D for Delicious. For the mixture approach the best tag- and username-based approaches are combined with mixture coefficient λ=0.86.

Table 3.21 Matching results based on user tags when using aggregated profiles.

Strategy	**MRR**	**S@1**	**S@2**	**S@3**	**S@10**
TFIDF	0.335	0.259	0.323	0.356	0.470
BM25	0.391	0.326	0.385	0.414	0.505
BM25 specific IDF	0.453	0.393	0.451	0.474	0.560

best tag- and username-based user identification strategies. We see again that having more user information increases the precision of the user identification challenge. For the aggregated profiles the mixture of the tag-based BM25 approach using site specific IDF and LCS for measuring similarity of usernames leads to an improvement of 15.1% over the setting where no profile aggregation is done.

Figure 3.7 summarizes the user identification performance for different profile aggregation settings. It is interesting to see that for all settings where Flickr profiles are unified with Delicious or StumbleUpon profiles, the combination of tag- and username-based strategies (*mixture*) achieves a success (S@1) of nearly 90%.

Table 3.22 Results based on usernames for aggregated profiles.

Strategy	MRR	S@1	S@2	S@3	S@10
ExactMatch	0.555	0.542	0.547	0.547	0.547
Jaccard	0.684	0.654	0.678	0.686	0.717
SmithWaterman	0.437	0.217	0.224	0.476	0.747
Levenshtein	0.721	0.701	0.718	0.722	0.735
LCS	0.727	0.701	0.725	0.731	0.746

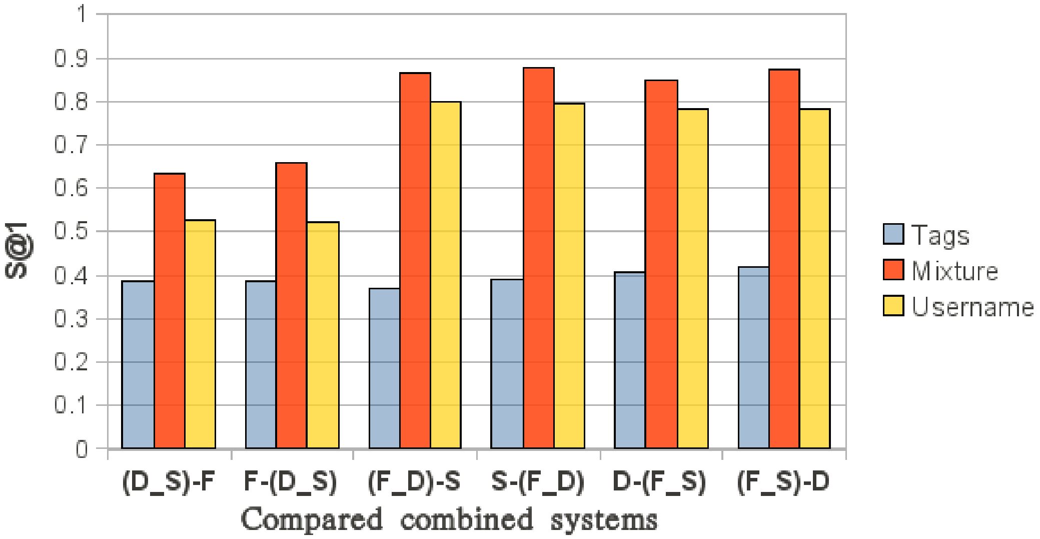

Figure 3.7 Success at rank 1 for different approaches (mixture coefficient λ=0.86) and settings when dealing with aggregated profiles. *D_S* stands for the setting where individual profiles are aggregated from Delicious and StumbleUpon, etc.

Synopsis

Table 3.24 summarizes the results for the 1467 Flickr and Delicious profiles (see FD dataset, Section 3.2.6) that confirm our findings. The mixture of the best tag-based approach (BM25 using site specific IDF) and best username-based approach (LCS) leads to significant improvement from 0.250/0.452 to 0.543 for the tag/username based approaches regarding S@1. In summary, we conclude that (1) it is possible to identify users across systems based on their tagging behavior and user ids, (2) for the user identification based on tag profiles our new approach of BM25 in combination with site specific IDF outperformed the other approaches significantly and (3) knowing more about the user (profile aggregation) and combining tag- and username-based approaches (mixture) further improves the performance to an accuracy of 79.2% with

Table 3.23 Results for the approach of combining the best tag- and username-based strategies on aggregated profiles (mixture coefficient λ=0.86).

Strategy	**MRR**	**S@1**	**S@2**	**S@3**	**S@10**
no aggregation	0.677	0.641	0.681	0.697	0.728
aggregation	0.816	0.792	0.818	0.832	0.855

respect to S@1.

Table 3.24 Summary of results for identifying users across Flickr and Delicious on the larger FD dataset (mixture coefficient λ=0.86).

Strategy	**MRR**	**S@1**	**S@2**	**S@3**	**S@10**
BM25(tags)	0.343	0.250	0.303	0.330	0.428
LCS	0.564	0.452	0.489	0.502	0.535
Mixture	0.632	0.543	0.573	0.590	0.624

3.2.7 Discussion

In order to better understand the Web user, systems should find ways to aggregate the information the user publishes on the Web. The main contributions of the work done in this section and experimental results can be summarized as follows:

- We proposed different strategies that allow for the identification of users across systems based on the users' tagging practices and on their chosen usernames. Thus, we examined the Web profiles from three different social networking websites, Flickr, Delicious and StumbleUpon. We exploited explicit (usernames) and implicit (tagging behavior) feedback to construct user profiles for identifying users across different systems.

- For tag-based profile mapping, we introduce a symmetric variant of BM25 using site specific statistics and compare it against measures like TF, TFIDF and conventional BM25 as well as against probabilistic language models. The results show that it is important to account for the specifics of a site, since tagging behavior and thus kinds of tags vary a lot between tagging systems. We also experiment with various string similarity measures for the username comparison and with combining and aggregating the different information sources.

- We evaluate the different matching approaches in experiments with more than 300 users, considering their public profiles from three different social tagging networks, Flickr, Delicious and StumbleUpon. Even though the tagging behavior varies considerably between the analyzed systems, Flickr (e.g. personal and public images) and Delicious and StumbleUpon (e.g. public Web resources), we managed to achieve a success rate of 30% when identifying users based on tags alone. We show how by combining implicit and explicit profiles we reach an accuracy of over 60% when identifying users across systems. We confirm the results on a bigger set of 1467 users having profiles in Flickr and Delicious. These best results were achieved by using Longest Common Sequence (LCS) based distance for username comparison and BM25 with site specific IDF for the tag specific profiles. This newly presented adaptation of the weighting scheme also outperforms approaches based on language modeling.

- Furthermore, with aggregated profiles we reached an average success rate of almost 80% for identifying the user across social tagging systems and for some settings even nearly 90%, thus showing the benefits (and risks) of knowing more about users. Being able to aggregate profiles from different systems can lead to better personalized services in the respective systems, especially when dealing with the “cold start” problem.

4 Entity Search on the Web

In the previous chapter we introduced a model for collecting and maintaining user profiles in systems where users tag resources. Also, we showed how the tags users assign can help describing the users' interests, thus aiding with recommendation services and with user identification. In this chapter we focus on how the tags assigned implicitly and explicitly by users to resources can be used to enrich the resource description, thus aiding the search process, for example. From the problems presented in the introductory Chapter 1, we address here Problem 3 (*How to exploit implicit and explicit user tags for improving typed item retrieval?*) and Problem 4 (*How to exploit users behavior for allowing typed item retrieval on the Web?*), while Problem 1 (*How to build and maintain user profiles given the user's tagging activity?*) and 2 (*How to exploit user profiles for improving the user's experience across social Web applications?*) were addressed in Chapter 3.

This chapter deals with the following issues:

1. A model for exploiting explicit tags for improving typed item search

2. Exploiting implicit tags from search logs to find entities, or typed items

The main component of our data model is the typed item we refer to as an entity, representing a real word object. An entity is a data structure consisting of a unique identifier and a set of attributes describing its type and attributes.

Web search increasingly deals with structured data about items (i.e. people, places and things), their attributes and relationships. In such an environment an important task is matching a user's unstructured free-text query to a set of relevant entities. The most challenging problem is to find relevant entities, of the correct type and characteristics, based on the free-text query.

We first show in Section 4.1 an entity ranking relevance feedback model, based on example entities specified by the user or on pseudo feedback. The model employs the Wikipedia category structure, where entity categories are explicit tags assigned by users to entities and give information about the types of the entities. In the second part of this chapter, in Section 4.2, we propose an approach to entity retrieval by using Web search engine query logs, where the queries posted by users can be interpreted as implicit tags.

4.1 Finding Entities in Semi-Structured Data

Finding entities of different types is a challenging search task which goes beyond classic document retrieval and also single-type entity retrieval such as, for example, expert search [BdVCS07]. The motivation for this task is that many 'real searches' are not looking for documents to learn about a topic, but really seek a list of specific entities: restaurants, countries, films, songs, etc. Example needs include 'Formula 1 drivers that won the Monaco Grand Prix', 'Female singer and songwriter born in Canada', 'Swiss cantons where they speak German', and 'Coldplay band members', just to name few.

This is a new interesting task that goes beyond standard search engine's matching between user query and document features. In the Entity Ranking (ER) scenario the user is looking for a set of entities of the same type with some common properties, e.g., 'countries where I can pay in Euro'. This query is answered by current web search engines with a list of pages on the topic 'Euro zone', or ways to pay in Euros, but not with a list of country names as the user is asking for.

The complexity of this search task lays in the multi-step solution that should be adopted. Firstly, the system has to understand the user query, what is the entity type and which are its properties. Similarly to expert search, the index should contain entities instead of just documents, and the entity type should be represented in and matched against the user query. Therefore, several techniques from research fields such as Information Extraction and Natural Language Processing (NLP) could be used as well in order to first identify entities in a document collection. Moreover, a hierarchy of possible entity types and relations among entities and their types have to be considered [RSH08, TSR$^+$08].

Initial attempts to ER have recently been presented. The main approaches build on top of the link structure in the set of entities [PVT08], use passage retrieval techniques, language models [RSH08], or NLP based solutions [DFI$^+$08].

In this thesis section, we propose ReFER: a graph-based method to take advantage of relevance feedback (RFB) in entity retrieval, exploiting either example entities provided by the user, or the top-k results from an ER system. We show how the combination of relevance feedback results with the initial system improves search effectiveness for all runs submitted to the Initiative for the Evaluation of XML Retrieval (INEX)[1] 2008 XML Entity Ranking track. The proposed method is designed

[1] http://www.inex.otago.ac.nz/

based on the Wikipedia setting used at INEX but it could be adapted to other settings such as the one of tag recommendation (i.e., tagged web pages compared to Wikipedia articles belonging to Wikipedia categories).

4.1.1 Category expansion in Wikipedia

We present an entity ranking model based on assigning entities to 'smooth categories'. This in turn is based on the Wikipedia link and category structure. This subsection describes the two key properties of Wikipedia we rely on to develop our model. The next subsection describes the smooth category model.

Wikipedia is a free encyclopedia with 2.7 million English articles written by volunteers.[2] It is a collaborative website with an editorial process governed by a series of policies and guidelines.[3] Wikipedia has two properties that make it particularly useful for ER. The first is that many of its articles are dedicated to an entity, so the entity ranking problem reduces to the problem of ranking such articles. The Wikipedia guidelines prescribe that an entity should have at most one article dedicated to it, according to the *content forking* guidelines. Thus the entity ranking model does not need to eliminate duplicates. Many real-world entities have no Wikipedia page, according to the *notability* guidelines. To be included, an entity should have significant coverage in multiple independent, reliable sources. For example, the model can rank major-league baseball players according to some entity-ranking query, but not players in youth baseball leagues, since youth players rarely meet the notability criteria.

In this setting, a simple ER solution is to rank Wikipedia pages in a standard IR system. If we search in a List Completion manner(i.e. query by example), for 'John F. Kennedy' in an index of Wikipedia pages, the top-ranked articles are: 'John F. Kennedy', 'John F. Kennedy International Airport', 'John F. Kennedy Jr.', 'John F. Kennedy Library' and 'John F. Kennedy assassination in popular culture'. The IR system has succeeded in finding pages relevant to the topic of JFK. However, if the information need were related to finding US presidents, the system has not succeeded. It did not find entities of a similar type. As a concluding remark, note, some articles do not pertain to an entity (e.g., 'Running'); we have to rely on the entity ranking model to avoid retrieving these.

The second useful property of Wikipedia is its rich link and category structure, with the category structure being of particular interest when finding entities of similar type. Intuitively, one would say that if two entities are related by satisfying an information need, they should have at least one common category. The more common categories two entities belong to, the more related they are likely to be. The usefulness of Wikipedia's link structure has been confirmed in the INEX entity ranking experiments: participants found that category information, associations between entities and query-dependent link structure improved results over their baselines [dVVT+08].

[2] http://en.wikipedia.org/wiki/Wikipedia

[3] http://en.wikipedia.org/wiki/Wikipedia:Policies_and_guidelines

However, as Wikipedia is a collaborative effort, no strict rules enforce the guidelines for linking between entities or assigning entities to categories. Entities may belong to many categories describing its different aspects, and no limit exists on the number of categories an entity could get assigned. For example the Wikipedia page describing 'Albert Einstein' links to a wide variety of entities, including specific information such as 'Theory of relativity' and 'Nobel Prize in Physics', but also more generic facts like 'Germany' and 'Genius'. Considering the Wikipedia category structure, 'Albert Einstein' belongs to some sixty categories, varying from 'German Nobel laureates' and 'German immigrants to the United States' to '1879 births'.

The categories of a page are not weighted by their importance, so we do not know which is more important, and a page may also be missing from important categories. For example, in our snapshot of Wikipedia the article on South Korea is in the categories: 'South Korea', 'Liberal democracies' and 'Divided regions'. There are attributes of South Korea that are not described by categories.

4.1.2 Link-based Relevance Feedback for Entity Ranking

In this subsection we describe ReFER, our RFB algorithm based on the link structure of the Wikipedia model, and we then present ways of integrating it with existing ER systems.

In our model we assume a collection of categories $C = \{c_1, .., c_n\}$ and a collection of entities $E = \{e_1, .., e_m\}$ are given, where a category is a tag describing an entity.

Definition An entity e_i is a tuple $< uri, desc, C_{ei}, R_{ei} >$ where uri is the entity identifier, desc is a string describing e_i, $C_{ei} \subseteq C$ is the set of categories listed in the entity e_i and $R_{ei} \subseteq E \setminus \{e_i\}$ is the set of entities e_i links to.

Our collection graph has two types of nodes given by entities and categories in Wikipedia, where the connection between two entities is denoted by a $link =< e_i, e_j >$ and the connection between an entity and a category by $edge =< e_i, c_j >$. Thus, there is a link between entity e_i and each entity in R_{ei} and an edge between e_i and each category in C_{ei}. Edges may be of two types. The 'hard' edges represent the collection entity-category structure, giving us the 'hard' categories set for an entity e_i defined as $C_H(e_i) = \{c | e \in E,\ c \in C_{ei}\}$. From the graph structure we can then infer the 'smooth' edges for an entity as the 'hard' edges of its linked entities. Thus the set of 'smooth' categories is defined as $C_S(e_i) = \{c | c \in \bigcup C_{ej}, e_j \in R_{ei}\}$.

The ReFER Algorithm

Our entity ranking algorithm can be described as propagation of weights through a directed acyclic graph. The graph has nodes in three layers: an 'input' layer of entities, an 'intermediate' layer of hard and smooth categories and a ranked 'output' layer of

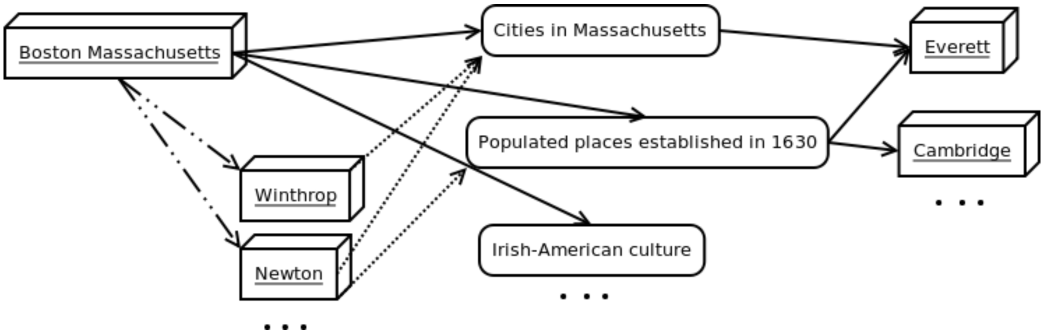

Figure 4.1 Three layer graph, with input node entity 'Boston, Massachusetts'. Solid edges indicate hard categories, dashed edges indicate smooth categories.

entities connected to the 'intermediate' categories. Weights propagate through graph and are proportional to the number of links, hard edges and smooth edges.

For the example in figure 4.1, if the article on *Boston Massachusetts* is in the category *Cities in Massachusetts*, and links to several pages that are also in that category, then the article's input node is connected to *Cities in Massachusetts* node via both a hard edge and a smooth edge. In our example, category *Cities in Massachusetts* will be weighted higher than category *Irish-American culture*, as the latter has no soft edges leading to it. Soft categories can add extra weight to hard categories, and also make associations with new categories. For 'South Korea', the original category that is most strongly supported is 'Liberal Democracies', since seven of the articles linked-to by the 'South Korea' article are 'Liberal Democracies'. The page is associated to 26 smooth categories, out of which 14 contain the word Korea. There is though some noise in the smooth categories, like 'Constitutional Monarchies' and 'History of Japan'. In order to reduce the amount of noisy smooth categories for an entity e_i we filter out the ones with less than 2 entities from R_{ei} belonging to them.

Given a query q we activate a certain set of nodes E_q as input for our algorithm. Then for each category node in $C_H(e_i) \cup C_S(e_i)$, where $e_i \in E_q$, we sum the incident edge weights from active input nodes from E_q. For category c_j let us denote the total incoming hard-edge weight as h_{cj} and smooth-edge weight as s_{cj}. In our initial experiments, we noticed that the hard-category 'coordination' between the input nodes is important. If there is one category that is common to most of the active input nodes, then that category is extremely important, and should massively outweigh the other categories. This led us to develop the following exponential category weighting heuristic:

$$cw(c_j) = \frac{\alpha^{h_{cj}} + s_{cj}}{\log(catsize(c_{cj}) + \beta)}, \tag{4.1}$$

where $catsize(c_j)$ is the number of Wikipedia pages in the category c_j and α and β are parameters[4], β being used so that the logarithm does not return negative values. The log down-weights very large categories, since these are unlikely to be discriminative. Akin to stopword removal,we eliminate categories with many entities (in our setup we considered a threshold of 1000 entities).

If there is a category that is common to all input nodes in E_q, then it will have high h and a much higher weight than any other category. For example, if the input nodes are a number of entities in the category *Cities in Massachusetts*, then that category will dominate the rest of the entity ranking process. If there is not a dominant category, then both hard and smooth categories come into play under this weighting scheme.

To rank entities, we propagate and sum the category weights to the output layer. The final entity ranking weight of output node e_k includes a popularity weight $P(e_k)$:

$$ew(e_k) = (\sum_{j=1}^{n} cw(c_j)) * P(e_k). \tag{4.2}$$

The popularity weight is based on the Wikipedia link graph where node e_k has indegree IN_k, such that $P(e_k) = min(\theta, \log(IN_k))$, θ being a parameter[5]. Static rank, a well-known concept from Web search, is a query-independent indicator that a certain search result is more likely to be relevant (see, for example, PageRank [PBMW98]). We found that connectivity in Wikipedia is an indicator that an entity is well-known, and therefore possibly a good search result.

ReFER Bootstrap and its Application to ER systems

The algorithm we propose is query independent as it just needs an initial set of entities where to start from. ER systems start from keyword queries provided by the user in order to generate a ranked list of results. We propose three ways of running our algorithm and combining it with existing ER systems.

In the first scenario the user provides also a small set of example relevant entities. We can use such set as the active nodes E_q from input layer I. We would thus obtain a ranked list of entities ordered by decreasing $ew(e_k)$ scores. It is then possible to merge, for example by means of a linear combination, the obtained ranking with one produced by an ER system which uses keywords provided by the user. In the thesis we perform ranking combination in the following way[6]:

$$rank(e_k, q) := \lambda \cdot baseline(e_k, q) + (1 - \lambda) \cdot ReFER(e_k), \tag{4.3}$$

[4] Experimentally exploring the parameter space we obtained best results with $\alpha = 10$ and $\beta = 50$

[5] Experimentally exploring the parameter space we obtained best results with $\theta = 5$

[6] A different option would be to combine RSVs of the baseline ER system with $ew(e_k)$ scores. Due to the variety of approaches that lead to the scores in different ER systems, we could estimate such scores transforming the rank of entity e_k for query q; we carried out experiments computing the rank-based scores as $(1000 - rank)$ and $(1/rank)$. As the conclusions resulting from both transformations turned out identical we perform a simpler combination of ranks.

where $rank(e_k, q)$ is the new rank for entity e_k on query q, $\lambda \in [0, 1]$, $baseline(e_k, q)$ is the rank assigned by the baseline system, and $ReFER(e_k)$ is the rank assigned to e based on the scores computed by Formula 4.2.

A second approach would be to use results of an ER system in order to bootstrap our algorithm (i.e., as elements of the input layer). Thus, in a pseudo-RFB fashion, we consider top-k retrieved entities as being part of E_q. Again, in this way we would obtain a ranked list of entities by running the ReFER algorithm. We can now combine the two available rankings, for example, in a linear combination.

A third approach, is the RFB one. After the ER system retrieves results for a query, the user selects relevant results present in top-k. We can use selected relevant results as elements of active input layer E_q. Again, we can combine the two rankings (the original one and the one generated based on Formula 4.2) by a linear combination.

4.1.3 Evaluation

We are now presenting an experimental evaluation of the proposed model for RFB in ER. We start describing the test collection we use and we then evaluate effectiveness of different applications to existing ER baseline systems.

Experimental Setting

The Entity Ranking track at INEX has developed a test collection based on Wikipedia. We perform our experiments on this test collection, for an objective and comparable evaluation. We will consider our RFB approach successful if it improves consistently upon the measured performance for most (or all) of the runs submitted to the track, essentially using the participant runs as baselines. This is an especially challenging goal in case of runs that already use the Wikipedia link structure between entities and/or categories.

The document collection used for evaluating our approach is the 2006 Wikipedia XML Corpus[DG06] containing 659,338 English Wikipedia articles. In INEX 2008, 35 topics have been selected and manually assessed by the participants[7]. An example of an INEX 2008 Entity Ranking Topic is presented in Table 4.1. The track distinguishes between the XML Entity Ranking (XER) and the List Completion (LC) tasks. In the XER task, participants use topic category and topic title; in the LC case, the example entities provided in the topics can be used by the system (and should not be presented in the results). Because the assessment pool has been created using stratified sampling, the evaluation metric used is xinfAP [YKA08], an estimation of Average Precision (AP) for pools built with a stratified sampling approach.

[7] The test collection we used is available at: http://www.L3S.de/~demartini/XER08/.

Table 4.1 INEX Entity Ranking Topic example.

Title	Italian Nobel prize winners
Categories	#924: Nobel laureates
Examples	#176791: Dario Fo #744909: Renato Dulbecco #44932: Carlo Rubbia

Using Topic Examples

In order to evaluate the combination of ReFER with previously proposed ER systems, we decided to apply our algorithm to all the submitted runs at INEX 2008 as baselines as well as to the top performing runs of a later method tested on the same collection [BBdR10]. We then combine the results with baseline systems following Formula 4.3.

We performed such experiment with both XER and LC runs. The values of xinfAP for the original runs and the combination with the ReFER run are presented in Figure 4.2 for the XER task. The Figure shows how in all cases the combination of the baseline with ReFER improves the quality of the original ER system. For the runs where the initial baseline performs well (a high xinfAP), the best average value for lambda is close to 0.25 (i.e., giving more importance to the baseline). Baselines that did not perform that well require a higher λ of 0.75, giving more importance to ReFER results. For both tasks, the value of λ that yields best absolute improvement (i.e. 6.4% for XER and 5.2% for LC) is 0.5, so we present the following experiment results only for this combination strategy.

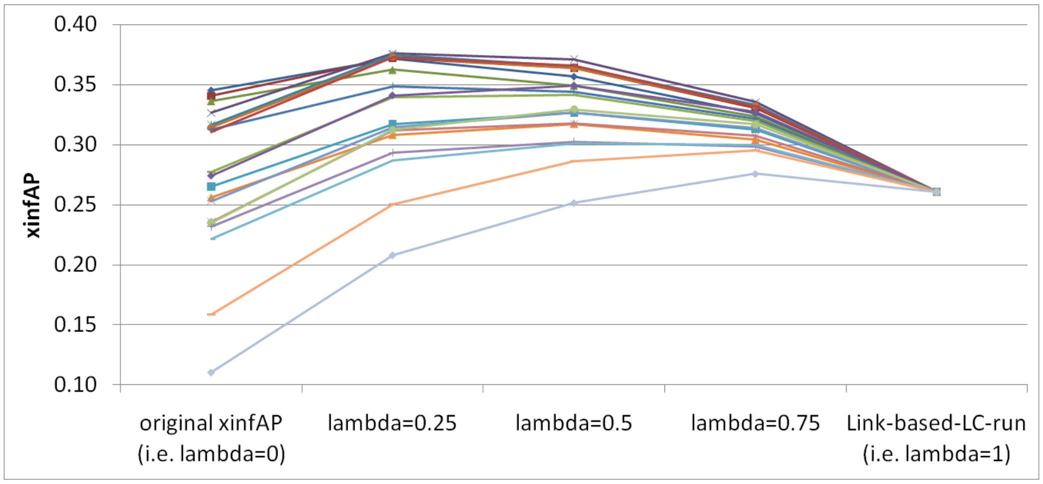

Figure 4.2 Comparison of runs submitted at INEX 2008 for the XER task when merged with ReFER using the topic examples for different λ.

Content Based Pseudo Relevance Feedback

How does the ReFER approach perform as compared to standard content based pseudo-RFB? As we do not have access to the retrieval systems used to create the various runs, we implemented a system independent method. From each run we start from the top k retrieved results, from which we take top n common terms. The terms are ranked based on the cumulated TF-IDF score from the k documents. Next, we search with both the topic title and the top n common terms in our index of the INEX Wikipedia and retrieve ranked lists of results for each run. We then combine such result set with the corresponding original run by applying Formula 4.3 with $\lambda = 0.5$.

Experimental findings show that this method performed best on average when using top 5 common terms from top 10 retrieved documents. The maximum absolute improvement achieved by the content based approach is of 2% on average. Also, the content based method improved only 79% of the runs (15 runs out of 19).

Pseudo Relevance Feedback

Instead of using the example entities provided in the topic we can use top-k retrieved results from each run. In this way, we build a system that requires no user involvement, but that just builds on top of another method for ER.

For each query q we activate the k nodes in the input layer that correspond to the top-k retrieved results from the baseline run. Figure 4.3 shows the xinfAP values for the original runs and for the combination (i.e., Formula 4.3 with $\lambda = 0.5$) with such pseudo-RFB run, for different value of k.

In Tables 4.2 and 4.3 it is possible to see that, on average, $K = 10$ gives best improvement both for xinfAP and for the expected P@20 (as used in [YKA08]). A t-test shows that the xinfAP improvement using $k = 10$ and $\lambda = 0.5$ over each baseline is statistical significant ($p \leq 0.05$) for all systems but one, where $p = 0.53$.

Table 4.2 Expected P@20 measured for different values of k in the pseudo-RFB case.

	K=5	K=10	K=15	K=20
Original	0.307	0.307	0.307	0.307
pseudo-RFB	0.284	0.290	0.277	0.269
Combination $\lambda = 0.5$	0.327	0.328	0.319	0.315
Abs. improvement	0.020	0.021	0.012	0.007

The results show how a small but effective seed leads to good results after applying the score propagation. When analysing the contribution of unique relevant results from the baseline and the pseudo-RFB we can see (Table 4.4) that most of the relevant results are present in both runs while only 4 relevant entities out of 21, on average, are not retrieved.

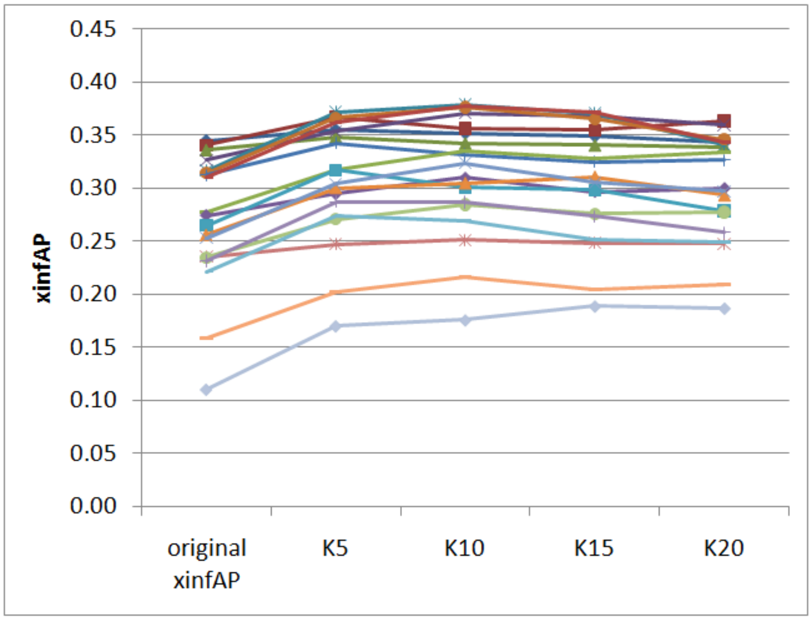

Figure 4.3 Improvement of xinfAP for each run using all results in top-k retrieved as seed of the algorithm, combining with $\lambda = 0.5$.

Table 4.3 xinfAP measured for different values of k in the pseudo-RFB case.

	K=5	K=10	K=15	K=20
Original	0.270	0.270	0.270	0.270
pseudo-RFB	0.266	0.275	0.267	0.256
Combination $\lambda = 0.5$	0.308	0.313	0.307	0.300
Abs. improvement	0.039	0.043	0.037	0.030

Relevance Feedback

In the next scenario we assume entity ranking in an interactive setting where the user can click on the relevant entities in the top-k results returned by the baseline system (i.e., RFB). Because assessing the relevance of entities returned can be considered to take a much lower effort than reading documents in a traditional information retrieval setting, we believe the ER setting justifies measuring the improvement in quality of the full displayed list (as opposed to the rank freezing or residual ranking methodologies that are more appropriate in the ad-hoc retrieval case [RL03]). When performing an entity retrieval task, the user's aim is not to read new relevant documents, but rather to obtain a precise and complete list of entities that answers the query. Thus, we use only relevant entities in top-k as seed to our algorithm. The results are shown in Figure 4.4.

For xinfAP, it is possible to see how the algorithm obtains best performances with $k = 20$ (cf. Table 4.5 and 4.6).

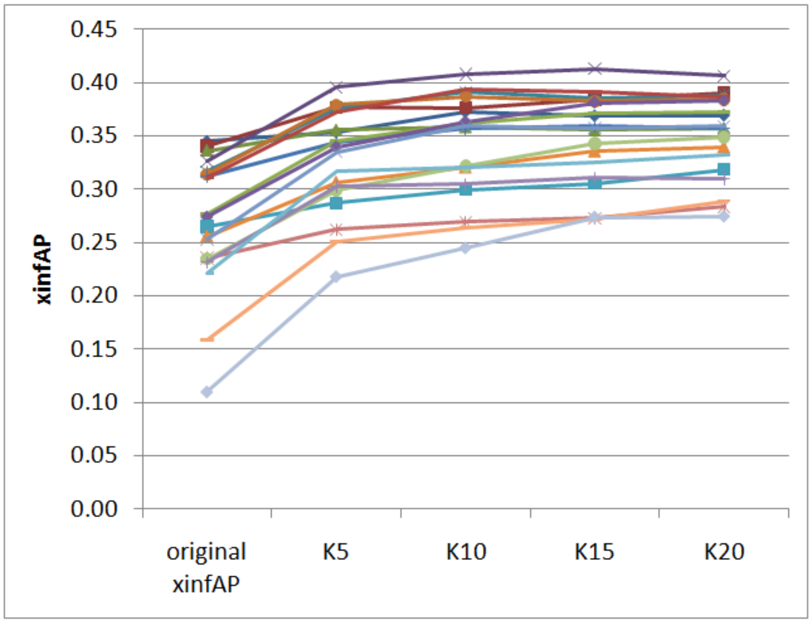

Figure 4.4 Improvement of xinfAP for each run using only relevant results in top-k retrieved as seed of the algorithm, combining with $\lambda = 0.5$.

Table 4.4 Average unique contribution of relevant results from the original baseline and the pseudo-RFB.

	K=5	K=10	K=15	K=20
Relevant in baseline	5.158	4.654	4.557	4.495
Relevant in pseudo-RFB	3.289	3.544	3.555	3.425
Relevant in both	10.694	11.198	11.296	11.358
Missed relevant	4.010	3.754	3.744	3.873

If we compare Tables 4.2, 4.3 and Tables 4.5, 4.6 we can see that in the pseudo-RFB case, the best improvement is obtained using the first 10 retrieved results. In the RFB scenario, given that input entities are all relevant, the higher the value of k, the better the improvement. We did not study the effect of $k > 20$ because we do not expect a user to select relevant results lower than rank 20. A t-test confirms statistical significance ($p \leq 0.05$) of the improvement in xinfAP between the run using $k = 20$ and $\lambda = 0.5$ and each of the baselines.

If we analyze the contribution of unique relevant results from the baseline and the RFB results (Table 4.7) we see that the baseline contributes more than the pseudo-RFB part. Compared to the contribution of uniquely relevant entities in the pseudo-RFB scenario (see Table 4.4), we find however that blind feedback works better with respect to this aspect. This result can be explained by the fact that when considering system-topic pairs in almost 20% of the cases there are no relevant results in top-k retrieved results. There are only 7 topics for which all systems had relevant results

Table 4.5 Expected P@20 measured for different values of k in the RFB case.

	K=5	K=10	K=15	K=20
Original	0.307	0.307	0.307	0.307
pseudo-RFB	0.295	0.332	0.339	0.347
Combination $\lambda = 0.5$	0.386	0.382	0.380	0.381
Abs. improvement	0.037	0.049	0.056	0.060

Table 4.6 xinfAP measured for different values of k in the RFB case.

	K=5	K=10	K=15	K=20
Original	0.270	0.270	0.270	0.270
pseudo-RFB	0.281	0.310	0.320	0.327
Combination $\lambda = 0.5$	0.327	0.341	0.347	0.350
Abs. improvement	0.058	0.071	0.077	0.081

in top 5 retrieved results. Thus in the RFB scenario we cannot apply our algorithm for all the system-topic pairs, whereas for pseudo-RFB the algorithm is applied also using only non-relevant entities.

Hard vs. Smooth Categories

What is the benefit of using hard and smooth categories? In order to observe the effect of using smoothed categories along with hard categories we experimented with various sets of categories both in the pseudo-RFB and RFB cases (see Tables 4.8 and 4.9). We used as input nodes top k=10 retrieved results from the baseline (for the RFB case we only used the relevant from top 10 retrieved results, amounting to 3.63 results per topic). In both cases the use of soft categories improves the overall performance of the analyzed systems. Furthermore, in the pseudo-RFB case, where also non-relevant entities are used as seed, the smoothed categories have a higher impact on the overall improvement.

Table 4.7 Average unique contribution of relevant results from the original baseline and the RFB.

	K=5	K=10	K=15	K=20
Relevant in baseline	7.14	5.78	5.32	4.95
Relevant in RFB	2.02	2.65	2.96	3.11
Relevant in both	8.71	10.07	10.54	10.91
Missed relevant	5.28	4.65	4.34	4.19

Table 4.8 xinfAP measured for k=10 in the pseudo-RFB case.

	C_H	C_S	$C_H \cup C_S$
Baseline	0.270	0.270	0.270
pseudo-RFB	0.269	0.126	0.2753
Combination $\lambda = 0.5$	0.308	0.213	0.313
Abs. Improvement	0.038	-0.056	0.043

Table 4.9 xinfAP measured for k=10 in the RFB case.

	C_H	C_S	$C_H \cup C_S$
Baseline	0.270	0.270	0.270
RFB	0.306	0.097	0.310
Combination $\lambda = 0.5$	0.338	0.220	0.341
Abs. Improvement	0.069	-0.050	0.071

4.1.4 Per Topic Analysis

RFB methods are not always viewed favorably, because it can happen that the improvement *on average* is positive, but that this improvement comes at the cost of many queries that perform worse than their baseline performance. To check if this would be the case, we performed also a per topic analysis of the experimental results for the pseudo-relevance feedback approach with $k = 10$ and $\lambda = 0.5$.

It is indeed the case, that for 16 topics the relevant results in the baseline (distributed over 16 systems) resulted in zero relevant results after applying our pseudo-RFB method. However, this can be mainly explained by the fact that none of these runs had a measured performance in xinfAP greater than 0.09 (average xinfAP is 0.04). That is, such runs found their relevant results very low in the ranking, and therefore did not provide a good seed to the algorithms in the top-k results.

Conversely, out of 19 ER baselines on 35 topic, 9 systems on 3 topics that had 0 initial xinfAP resulted to have relevant results in the pseudo-relevance feedback run obtaining a xinfAP of 0.16. An exceptional case, one of those runs had 35 relevant results after pseudo-RFB.

We conclude from the per topic findings that the proposed RFB method is beneficial for all retrieval methods except those that underperform anyways.

4.1.5 Discussion

Entity Ranking is a novel search task that goes over document search by finding typed entities in a collection. The retrieved entities can be used, for example, for a better presentation of web search results. The main contributions of the first part of the work done in the first part of this chapter can be summarized as fallows:

- We presented a model for Relevance Feedback in the entity retrieval scenario. The proposed model is based on weight propagation in a directed acyclic graph that represents links between entity descriptions.

- We have used as experimental setting the Wikipedia as a repository of such entity descriptions and have evaluated our approach on the INEX 2008 benchmark. We have used the submitted runs as baselines and have shown, firstly, that performing fusion with the result of our algorithm using relevant entity examples as initial seed always improves over the baseline effectiveness.

- We have also evaluated our algorithm using as seed the top-k retrieved results in a pseudo-RFB fashion. The experiments demonstrate that, while in all cases the baselines were improved, using top 10 results yields the best improvement.

- Finally, we have shown how an emulated interactive feedback session (by using only the relevant entities in the top-k retrieved results) leads to an even higher improvement when performing a fusion with the baseline (i.e., a 0.12 absolute improvement in xinfAP using the relevant entities encountered in top 20).

4.2 Finding Entities exploiting Click-through and Session data

Current Web search engines retrieve textually relevant Web pages for a given keyword query. The idea behind *Entity Retrieval* (ER) is to find entities directly. As an example, consider the ER query "hybrid cars" where relevant results would be *Toyota Prius* or *Honda Insight*, but not an informative page about hybrid vehicles. Instead of the user browsing through all Web pages retrieved by the search engine, a list of relevant entities should be presented to the user. This not only saves the user's time, but also improves the search experience. As shown in previous work, a big percentage of web search engine queries are about entities [KT09]. Similarly, when having queries that should return a list of entities, it is quite difficult to compile such a list based on keyword search results. A user looking for "films shot in Venice" or "Spanish fish dishes" will have difficulties to find suitable answers. He or she has to manually compile the result list by extracting entities from the retrieved documents which is a cumbersome task. A commercial product addressing such type of queries is Google Squared[8] where the results for queries such as "hybrid cars" is a table with instances of the desired type.

We propose an approach for answering ER queries based on search engine query logs. This enables us to answer queries that web users are usually posting to commercial search engines exploiting collaborative user knowledge. In this section we apply the results of query log analysis to the recent IR task of Entity Retrieval. By mining a very large Web search engine query log with clickthrough data and session information we are able to create two types of graphs on which we can afterwards apply our algorithms:

- We create a *Click Graph* by using queries and URLs as nodes and connecting and weighting them by their user click frequencies, and
- A *Session Graph* by using only queries as nodes with edges between them if they appear in the same user sessions, again weighted by co-occurrence frequencies.

In order to utilize this information source for improving ER we perform a Markov random walk on the graphs. We employ graph traversal techniques with different weighting schemes in order to match result entities [9] to given queries. Experimental results show that the intersection of the click graph and the session graph is the best evidence for answering ER queries when traversing the graphs. Moreover, reinforcing the results considering in-links from deeper levels improves the results. Additionally, we show how the most relevant results are placed one step away from the original ER query (i.e., they are connected in the graph).

[8] http://www.google.com/squared

[9] Note that in our scenario the results for ER queries are themselves found in the queries given in the query log and not in the Web page texts

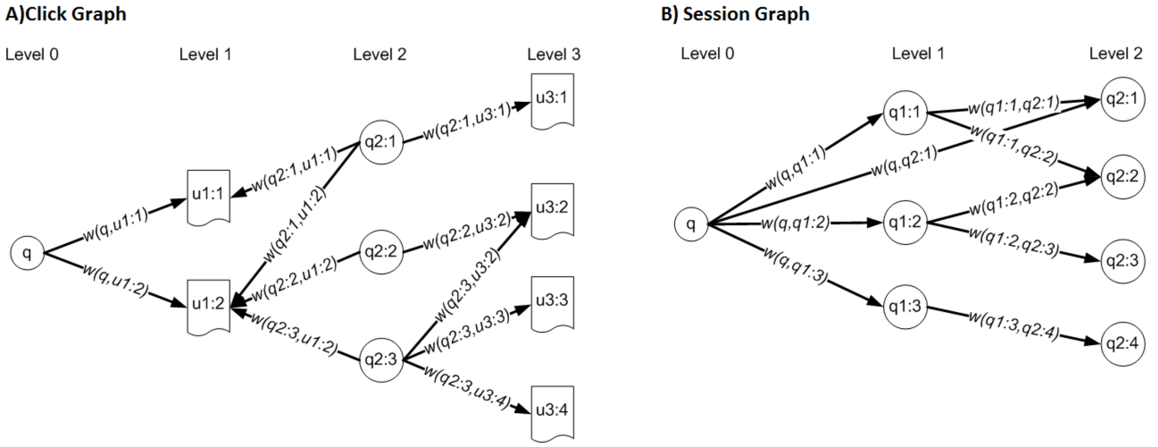

Figure 4.5 Schemes of a Click Graph (A), connecting an ER query q with entities $q_{l,i}$ via URLs $U_{l,j}$ where l indicates the level, and of a Session Graph (B) connecting a ER query q with queries $q_{2,i}$ on level 2.

The structure of the section is as follows.The following Subsection 4.2.1 describes the two types of graphs (click and session graphs) and how ER queries are matched with queries in the log. The approaches used are presented in Subsection 4.2.2 and the result of our performed experiments follow in Section 4.2.3. Finally, we conclude with a discussion of the results in Section 4.2.4.

4.2.1 Constructing and Entering the Graphs

The Click Graph.

A click log consists of a set of URLs $U = u_1, \ldots, u_n$ that users clicked on in response to queries $Q = q_1, \ldots, q_n$. Our approach for constructing the graphs is based on previous work of Craswell and Szummer [CS07]. We can build a *click graph* based on the notion of co-clicked URLs. In a click graph each unique query (i.e., a string of keywords) q_i and each URL u_j is a node. We define the set of nodes $V \equiv Q \cup U$. There is a directed edge between a query node q_i and a URL u_j if at least one user clicked u_j in the result page of the query q_i. Moreover, there is a weight on each edge computed based on the number of times u_j was clicked as result of query q_i. Such a graph represents relations between queries and web documents as well as between different queries. We define q as the starting point for such search process for entities: this is the ER query provided by the user (more details on how to properly select q are given in Section 4.2.1). We then assume queries close to q in the graph to be possible answers, that is, relevant entities q_i. In this way we can follow edges starting from q looking for relevant results (see Figure 4.5A).

The Session Graph.

In a session graph nodes are formed by the set of queries $V \equiv Q = q_1, \ldots, q_n$. There is a directed link from a query q_i to a query q_j if the query q_j was issued after query q_i in the same search session. Similarly, we can define q as the starting point, that is, the user's ER query. We can then follow the edges looking for relevant results (that is, queries q_i) in the queries connected to q (see Figure 4.5B). Finally, the task of finding entities can be then defined as ranking queries q_i by probability of being relevant to the ER query q. The hypothesis is that a user posing an ER query which does not yield satisfying results will reformulate the query to find useful information. Upon inspection, it seems that the reformulated query often consists of an instance of the group of entities the user is looking for, e.g. "Spanish fish dishes" and "Paella".

Finding the Entry Point in the Graph.

We investigate how we can identify a suitable subset of logged queries from which entities related to a particular topic can be extracted. We describe a possible way of selecting q (i.e., the starting point of the random walk) given the ER query issued by the user. We search the user query in the available query log and use such query as the node q. For instance, the query "salad recipes" can be found in the click graph as depicted in Figure 4.6. We then perform a random walk from this node in the graph. Beginning from this query, at the distance of two nodes out, the random walk finds such queries as "chicken salad recipe" as well as "pasta salad". Further out, the queries "green pea salad" and "caesar salad" are encountered. Specifically, we show the top ten queries with the highest transition probabilities from the node of origin (excluding the starting point), and a further five queries connected to two of these. While most of the queries directly linked to the original query are potentially useful for extracting entities, there are some queries that are less suited for this task. However, these can be understood as categorising queries that may lead to other promising queries which may otherwise not be reached from the originating node. Examples of these 'bridging queries' are the nodes "salads" and "salad recipes" – singled out in Figure 4.6.

4.2.2 Walking the Graphs for Entity Ranking

Similarly to [CS07] we perform a Markov random walk on the click and session graphs in order to find relevant results for query q. The main difference is that our goal is to rank queries connected to q rather than ranking URLs by the probability distribution computed with the random walk. Moreover, the resulting entities are found only in the log queries, disregarding the text of the Web pages pointed to by the URLs in the log.

Starting from the formalization of the graph done in Section 4.2.1, we define

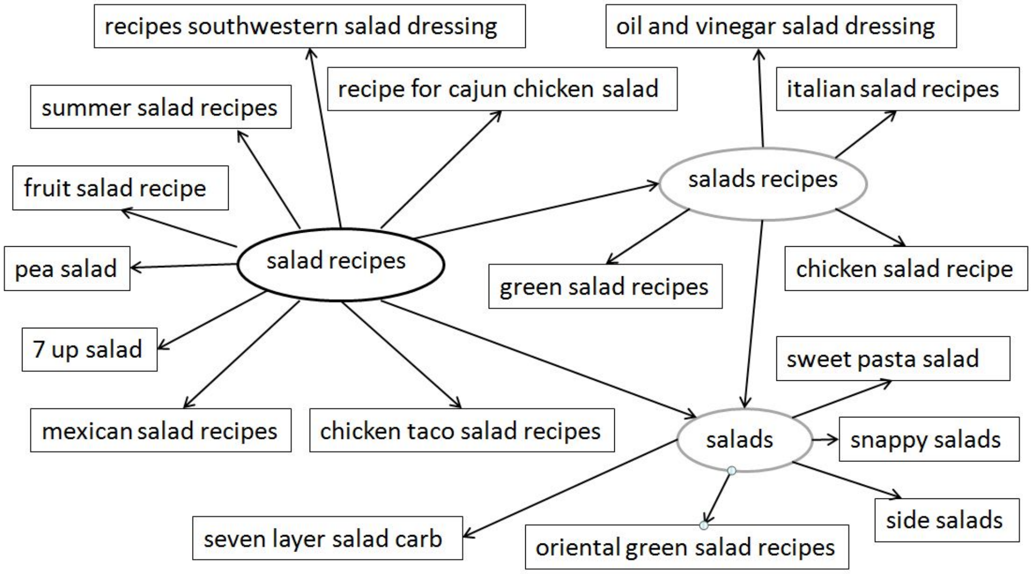

Figure 4.6 Selection of walked queries for the query “salad recipes”.

transition probabilities from a node j to a node k based on the click counts (i.e., $w(j, k)$ in Figure 4.5 A and B) as:

$$P_{t+1|t}(k|j) = \frac{w(j,k)}{\sum_i w(j,i)} \tag{4.4}$$

where i ranges over all nodes connected to j. The notation $P_{t+1|t}(k|j)$ defines the probability of moving from node j at step t to node k at step $t + 1$. Because our goal is to find relevant entities walking from a ER query in the graph, we set the self transition probability to 0 as slowing the diffusion to other nodes in the graph is undesirable for our purposes.

By storing these single step transition probabilities in a matrix A where $A[j, k] = P_{t+1|t}(k|j)$, it is possible to compute a random walk of t steps starting from node j ($P_{t|0}(k|j)$) as $[A^t]_{jk}$. That is, we sum weights on the edges encountered on all paths of length t between the node j and a node k. The more paths the higher the random walk probability of reaching k starting from j.

In [CS07] the authors also tested a transition model that normalizes transitions for documents and queries in different ways. We did not experiment with this as it is already shown to perform worse for the document ranking scenario.

Approaches Used on the Click Graph

In a search engine, a single query will attract clicks on several different URLs, establishing many outgoing links in the click graph. At the same time, a URL can be clicked on in response to several different queries. In this thesis we apply a Markov random walk model to a large click log. As shown in [CS07], this produces an effective probabilistic ranking of documents for a given query, including also relevant documents that have not been clicked on for that particular query.

Our goal is to produce a ranking of queries as response to an ER query posed by a user. We start under the assumption that Web search engine users click on the same URLs after querying for both ER queries but also for *results* of such ER queries. For example, we start from the query node "hybrid cars" in the graph and we retrieve 2-step away (through a co-clicked URL) the query "Toyota Prius". Based on such observation we define the following entity search procedure.

At search time, the given ER query is matched in the graph and set as starting node (see Section 4.2.1). Performing a random walk over the graph, using query-URL-query transitions associated with weights on the edges (i.e. click frequencies), as shown in Figure 4.5A, enables us to find relevant entities as other queries in the graph and present them as a ranked list of entity results. We retrieve all queries reached within up to ten random walk steps in the click graph (i.e. five queries deep) and five steps in the session graph from the original query. The retrieved set of results is ranked and/or filtered by one of the following methods and only results appearing two steps away (i.e. one query deep) from the original query are kept as precision values drop rapidly when considering more levels.

Simple Random Walk. This approach ranks all reached queries (interpreted as potential entities) by their random walk probability computed as described in Section 4.2.2, (using 0 as self transition probability and only forward walks) but keeps only queries which are one URL away from the original query (i.e., level 2 in Figure 4.5A) for the method labelled C_2. For the method labelled C_{10}, we keep any queries encountered up to 10 steps away from the original queries. The result queries (potential entities) are ranked by their random walk transition scores over all possible paths up to the respective depth. $C_2\_rein_{10}$ is a hybrid of these two, only keeping queries at level 2, but the probability estimates are derived by walks of up to 10 steps into the click graph.

Clustered Results. The $C_2\_cluster$ method works similar to C_2 but scores are determined solely by the probabilities of moving from each query to any of the adjacent URLs. Queries at level 2 are clustered based on their co-clicked urls. Each such URL has a score based on clicks from level 2 queries. The URL score is then added to the scores of its level 2 queries. Starting from the graph formalization in Section 4.2.1, we can define the scores for a level 1 or 3 URL u_i based on the click counts from level 2 queries as

$$S_{url}(u_i) = \sum_j \frac{C(q_j, u_i)}{\sum_k C(q_j, u_k)} \tag{4.5}$$

where j ranges over all the queries for which u_i was clicked and k ranges over all URLs connected to the query q_j. Level 2 query scores are then computed as

$$S_{query}(q_j) = \sum_i S_{url}(u_i) \tag{4.6}$$

where u_i are all the clicked URLs for query q_j. For example, in Figure 4.5A, the score of $q_{2,2}$ would be a sum of the scores of its URLs, $u_{1,2}$ and $u_{3,2}$ (where $u_{1,2}$'s score is the average of clicks from $q_{2,1}$, $q_{2,2}$ and $q_{2,3}$).

Loops in the Graph. $C_2\_loop_{10}$ differs from C_2 by keeping only queries which can be reached via multiple paths starting from the given ER query (i.e., those that are connected via URLs at deeper levels, in this case up to 10 steps). This approach would keep only $q_{2,2}$ and $q_{2,3}$ in Figure 4.5A. A level 2 query q_i is only considered if the path after ten steps from the origin goes through a different level 2 query and comes back to the query q_i. This approach still uses the computed probability distribution to rank entities but limits the retrieved set to those well connected in the click graph. Therefore, the queries ranked for $C_2\_loop_{10}$ are a strict subset of those ranked for C_2, following the same ordering.

Approaches Used on the Session Graph

In the case of using the session graph for answering an ER query, the hypothesis is that a user posing an ER query which does not yield satisfying results will reformulate the query to find useful information. Upon inspection, it seems that the reformulated query often consists of an instance of the group of entities the user is looking for, e.g. "Spanish fish dishes" and "Paella". This is not necessarily an ordered process but these kinds of co-occurrences can be found in user session logs. We collect session data from a Web search engine query log and we use it to build a graph containing each user query as a node as explained in Section 4.2.1 (see Figure 4.5). Figure 4.5.B shows a snapshot of a session graph with a query q and the connected other queries consisting of potential queries q_i. If two queries were posed in the same user session we connect the respective nodes. The direction of the edges goes from the earlier query to the others. Each of these edges is then weighted depending on the frequency of co-occurrence within different user sessions.

We also perform a single step random walk over the session graph starting from a given ER query 1 step away. Please note that 1 step on the session graph are equivalent to 2 steps on the click graph, where every other step ends on a RL, rather than a query. Similar to the Click Graph approaches, we keep only nodes in the graph which are adjacent to the starting node/query. This walk, denoted by S_1, ranks all the reached queries by their random walk probability when the random walk is performed only on the first level. That is, it does not explore the session graph at queries further away than those directly connected to the starting query.

We perform a random walk over the session graph starting from a given ER query up to 5 steps away. Please note that 1 step on the Session Graph is equivalent to 2 steps on the Click Graph, where every other step ends on a URL, rather than a query. Similar to the Click Graph approaches, we keep only nodes in the graph which are adjacent to the starting node/query.

Simple Random Walk on the Session Graph. Considering the Session Graph we compared the following approaches for ranking entities. S_5: Starting from the original query (the ER query), walk to all queries reachable in 5 steps and rank them by their random walk probability as described in [CS07]. Analogous, S_1 ranks all the reached queries by their random walk probability when the random walk is performed on the first level only. That is, it does not explore the session graph at queries further away than those directly connected to the starting query. In Figure 4.5B, these would be the queries depicted on Level 1. Similarly to $C_2\_rein_{10}$, $S_1\_rein_5$ forms a hybrid method.

Combining Click Graph Results with Session Graph Results

In order to exploit the two different graphs for answering the same query we can also use data fusion approaches given the two obtained rankings. In this thesis we follow the simple approach of summing retrieval status values (RSVs) used for ranking entities for each approach[10] and normalizing them by the maximum score. In this way we combine scores computed with the click and session graph.

Union. As first approach, we unite the two sets of results retrieved from the click and session graphs. Their relevance scores (i.e. random walk probabilities) are normalized for each of the two approaches and if a result item appears in both result lists, these scores are added. We label these approaches as $U_{C,S}$ e.g. U_{C_2,S_1} in the case of the union of C_2 and S_1.

Intersection. We also rank entities combining the results of the random walk on the two graphs by keeping only results which are retrieved by both approaches. Again, the relevance scores from the single approaches are normalized and then added together. Such approaches are labelled as $I_{C,S}$ e.g. I_{C_2,S_1} for intersecting results from C_2 and S_1.

[10] RSVs for ranking are the probabilities computed by the Markov Random Walk.

4.2.3 Evaluation

Experimental Setup

We use a query log from Bing[11]. It contains a sample of the most often clicked 35 million queries that were submitted over a period of 15 months by US American users to the search engine. This data consists of query as well as click specific details. Only query–URL pairs were retained for which at least 5 clicks were recorded overall. After some normalization of the queries (we used case folding and removed punctuation as well as non-alpha-numerical ASCII characters) there are 35 million unique queries and 44 million unique URLs. Aggregating identical query–URL pairs yields 242 million edges when constructing a click graph which can then be used for a random walk, as described in Section 4.2.1. The session data was sampled over a time period of 14 months that largely overlaps with the time period of the click data. It consists of 25 million unique queries and a total 105 million unique query reformulations were recorded. For this purpose, we define a reformulation as two queries that were issued in the same search session within 10 minutes.

Ground Truth

In order to evaluate the proposed algorithms we constructed a benchmark for ER evaluation out of Wikipedia As gold standard we use the "List of" pages from Wikipedia. The title of such a page , after removing the first two terms, is used as an ER query (e.g., "lakes in Arizona"[12]). The titles of the Wikipedia pages that are linked to from such a "List of" page are considered to be relevant results (e.g., "Canyon Lake", "Lake Havasu", . . .). In order to use only queries that are more similar to typical Web queries in terms of length, we kept only those queries that consisted of 2 or 3 terms apart from "List of". Thus we had 17,110 pages out of the total of 46,867 non-redirect "List of" pages. We matched these titles to queries in the log after lower casing and discarding any non-alphanumerical characters. In order to construct an experiment that is as realistic as possible, we filtered queries to only keep those which were posed at least at least 100 times in the query log and had at least 5 clicks on results. After this, we were left with 82 queries for evaluation[13].

For the selected queries we computed the result coverage in the two query logs. We counted how many of the queries from the ground truth are actually represented in the log files. For measuring the coverage we considered two types of measure: inclusion and complete matching. For inclusion matching, we consider a relevant result to be found if its words are included in a log query. For the complete matching, the cosine similarity between a log query and relevant entity has to be equal to one.

[11] http://www.bing.com/

[12] http://en.wikipedia.org/wiki/List_of_Arizona_lakes

[13] The test set of wikipedia titles and relevant entities is avail- able from http://www.l3s.de/~iofciu/wikipediaER/

Log	Inclusion match	Complete match	Relevant results per topic
Click	62.02	60.66	83.13
Session	52.18	50.35	83.13

Table 4.10 Query result coverage in the click log and session log

In Table 4.10 we show the number of relevant results found, averaged over the number of queries.

Results

The proposed algorithms produce, given an ER query q, a ranking of other queries which are present in the query log. In order to evaluate the proposed approaches we need to map ranked queries to relevant entities from our ground truth. In our evaluation, as described above, any entity title containing an entry in a "List of" page was viewed as a relevant entity to that ER query. As a pre-processing step, all queries, both from the ground truth and from the query logs have had the stop words removed and were stemmed afterwards. We consider a retrieved entity to be relevant to an ER query if the string representing the relevant entity includes the ER query. For example, the retrieved result "Lake Havasu pictures" would be considered relevant to the query "lakes in Arizona". Subsequent instances of the same entity (e.g. "Lake Havasu water") are ignored. This is a limitation of the current evaluation approach as it does not use exact match between retrieved and relevant entities. The current evaluation approach implicitly clusters retrieved queries based on the relevant result inclusion, and, subsequently, keeps the longest unique inclusion match. In this manner we only cluster the retrieved queries overlapping with relevant results, whereas the non-relevant results are not clustered and account then for lower scores. The exact match could be used in the case where entity extraction techniques would be applied to retrieved queries as a post-processing step. In order to compare the different ranking approaches, we computed Mean Average Precision (MAP), precision for the first ten results (P@10) and R-Precision (R-Prec) of the produced rankings. MAP is computed as: $MAP = \frac{1}{|ER|}\sum_{i=1}^{|ER|} AP_i$ where $|ER|$ is the number of ER queries and AP is obtained averaging the Precision values calculated at each rank where a relevant entity is retrieved [BYRN99]: $AP = \frac{1}{|Rel|}\sum_{i=1}^{|Rel|}\frac{i}{rank(i)}$, where $rank(i)$ is the rank of the i-th relevant entity, and $|Rel|$ is the number of relevant entities. A score of 0 is assumed for any not-retrieved relevant entities. R-Prec is defined as Precision computed after R retrieved results, where R is the number of relevant entities.

In Table 4.11 we compare our baseline runs C_2 and S_1 which are equivalent to ranking the queries directly connected to the user query by the weights on the edges. We can see that by using a Session Graph we obtain better results for ER queries. Moreover, while using the intersection of the Click and Session Graphs reduces the

Method	MAP	P@10	R-Prec	Queries Ranked	Relevant Entities Retrieved
C_2	0.1423	0.0959	0.0541$^+$	489.54	8.79
S_1	0.1864	0.1026	0.1106*	78.61	6.87
$S_1\_rein_5$	0.2011*	0.1123	0.1082*	76.37	6.63
S_5	0.0252*$^+$	0.0768$^+$	0.0410$^+$	2454724.54	40.92
U_{C_2,S_1}	0.1438$^+$	0.1054	0.0792*	537.95	11.80
I_{C_2,S_1}	0.2285*	0.1146	0.1283*$^+$	29.13	2.78

Table 4.11 Results for finding entities using click and session graphs, averaged over the 82 ER queries in the evaluation set. Differences in MAP and R-Prec are statistically significant by means of Single Factor ANOVA. A * indicates statistical significant difference to C_2 and a $^+$ to S_1 (paired t-Test with $p <= 0.05$).

Method	MAP	P@10	R-Prec	Queries Ranked	Relevant Entities Retrieved
C_2	0.1423	0.0959	0.0541$^+$	489.54	8.79
$C_2\_cluster$	0.1490	0.1069*	0.0597*$^+$	489.72	8.79
$C_2\_loop_{10}$	0.1533*	0.1077*	0.0647*$^+$	358.16	8.45
$C_2\_rein_{10}$	0.1490	0.1069*	0.0597*$^+$	489.72	8.79
C_{10}	0.0548*$^+$	0.1	0.0549$^+$	87313.18	35.48

Table 4.12 Results for finding entities using click graphs. Statistical significance numbers are given to the same baselines in the previous table.

result set size significantly (29 results instead of 489 and 78 respectively), it improves effectiveness scores. With this simple approaches recall is anyway very low as the average number of relevant results per query is 83. The approach of unifying the sets of entities retrieved from the two graphs is not performing well mainly because of the large amount of retrieved entities.

In Table 4.12 we compare results of different approaches on the click graph (see Section 4.2.2). Our baseline is again C_2, that is a 2-steps random walk starting from the user query node, which is equivalent to ranking connected queries by the weights on the edges. We can see that a longer random walk (e.g., 10 steps away from the starting node, $C_2\_rein_{10}$) gives a better estimation of the relevance of level 2 queries. Moreover, we see that retrieving only queries that are also supported at deeper levels in a 10-step walk (i.e., $C_2\_loop_{10}$) improves the effectiveness. Here, most of the relevant entities retrieved are kept while on average more than 100 non-relevant are discarded.

4.2.4 Discussion

Being able to provide users with results as entities would save a lot of effort on the user side, as opposed to manual aggregation of results spread over hundreds of relevant Web pages retrieved. On the other hand, doing the information aggregation work on the search engines site involves a lot of resources when having to analyze millions of Web documents automatically. Thus it is important to understand the type of information that can be mined from user search query logs, as the queries are usually short and faster to process. The main contributions of the second part of this chapter can be summarized as follows:

- We presented approaches for answering ER queries exploiting human behavior stored in search engine query logs.
- After constructing click and session graphs out of the logs, we perform a Markov random walk on the graphs in order to rank queries which contain relevant entities to a given ER query.
- We created a gold standard of 81 Entity Ranking queries based on Wikipedia “List Of” pages. The created ground truth dataset is available for download and it can thus be reused for evaluating and comparing ER algorithms.
- Experimental results showed that integrating results from both the click and the session graph yields best effectiveness. Moreover, the best results can be found at level 1, that is, those directly connected to the ER query. Such results are promising as they would allow to build systems that, given a user ER query, can answer in real time with no need of highly complex algorithms.

5 Conclusions and Future Work

This dissertation introduced a new methodology to address the user and entity profiling problem on the Web. More especially, the focus was on using the profiles in various services provided to the users, such as recommendations, user and entity search. The following paragraphs summarize the main contributions and explain how these overcome the challenges of mining Web data, as these are explained in Section 1.1.

Summary of Contributions

In the first part of Chapter 3 we addressed *Problem 1* announced in Chapter 1, inferring user profiles based on tagging activities that people perform on the Web. We propose a generic model for collecting and maintaining user profiles from the information users published on the Web. We use tags from social bookmarking systems to build user profiles. The user connections to different objects are translated into the tag domain, thus the user profiles overcome the problem of sparsity. After defining how to create tag-based user profiles we address *Problem 2*, how to use these profiles in order to enrich the user's experience on the Web. Thus, once we have a representation of a user's interest in the tag domain of a folksonomy, we show how we can provide the user with various types of recommendations. This thesis presented several techniques for user, tag and resource recommendations. We show how factors such as tag assignment network structure, tag importance and assignment time can play an important role in defining the users' interests. In the second part of Chapter 3 we focus on enriching the user profiles by identifying and aggregating profiles users have on various social bookmarking systems. We show how, based on their tag assignments users can be identified across social networks. In this section we have analyzed and matched user profiles between three systems: Flickr, Delicious and StumbleUpon. Furthermore, when using also the information explicitly provided by the users, such as usernames, we can match users with an effectiveness of 61%. By aggregating user profiles from different systems, we can better understand users' interests which can then lead to better personalized services.

In the first part of Chapter 4 we addressed *Problem 3*, how to use tags assigned by users implicitly and explicitly to resources in order to improve typed item retrieval. Tags can also be used for their original purpose, to better describe the various types of tagged objects, i.e. from Web resources to entities. Thus the tags can provide an extended item profile. In the first part of the chapter we presented a pseudo relevance feedback approach for using the tags assigned to entities to further improve entity search. We conducted our analysis on Wikipedia, where each page is an entity and users can assign these entities to categories, i.e. tags that reflect to some extent the entity type. We showed how by using the category tags we could further improve the effectiveness of different entity search systems from the INEX 2008 benchmark. Also the pseudo relevance feedback based on tags lead to a much higher improvement than the content based one. In the second part of the chapter we addressed *Problem 4*, analyzing user search patterns in order to answer entity search queries. We showed how from Web search logs we can construct click through and session graphs with queries and URLs as nodes. These can then be exploited and used for entity search, where the user interest is not in documents but in lists of relevant entities. In the context of this work, we created an evaluation dataset from Wikipedia ”List of” pages, which can be used for evaluating and comparing Entity Ranking algorithms. We showed how combining the results from the click and session graph leads to best effectiveness.

Open Directions

In this thesis we presented a number of applications of tags for recommendations, user identification and entity search. The proposed solutions pave the way for exploring further research directions. Some of future interesting research questions refer for example to detailed investigations regarding how to better make use of the network structure in social bookmarking systems in order to further improve recommendations. The recommendation services we presented were solely based on tag assignment, based on our results one can research the integration of also the content of the resources in the recommendation process.

Regarding the user identification problem, we experimented with social networks where the main activity is publishing and tagging. As a future direction one can aim at integrating more explicit profile attributes which are present in other types of social networking systems, e.g. user full name, location, email and other preferences, thus being able to aggregate user profiles from different types of social networks, for example Facebook, StudiVZ along with Flickr and Delicious. Other interesting attributes present in social networks are user groups and friend connections. There is a large space for further research in the area of user identification by considering the network structure along with the individual user profiles.

With respect to our pseudo relevance feedback approach for Entity Ranking, which can be easily applied to any ER system in order to improve search effectiveness, ouur model performs well on the test collection we used. A limitation of this work is the use of a single test collection. As future work, we aim at evaluating our approach on a different ER setting such as, for example, graph-based tag recommendation [HCOC02]. Also, a future direction is finding optimal parameters for the proposed model in order to get the best possible improvement in search effectiveness.

For the part on Entity Ranking based on Web search logs, future work involves developing methods for grouping retrieved queries based on different similarity measures and extracting the core representative query for each group. This way, for an entity ranking query, we can present the results to the user as a short list of query representatives.

A

Curriculum Vitae

Tereza Iofciu, born on March 31st 1982, in Bucharest, Romania.

Jul. 2011 -	Software Developer - Data Scientist, XING AG, www.xing.com
Oct. 2005 - June 2011	Junior researcher and Ph.D student at Forschungszentrum L3S, Universität Hannover
June 2008 - Sept. 2008	Research Intern Microsoft Research Cambridge
Mar. - Aug. 2005	Diploma Project in Computer Science, Universität Hannover Title of the thesis: *"Integrating Metadata and Full-Text Search"*
2000 - 2005	Diploma in Computer Science, Politehnica University, Bucharest, Romania
May 2004 - Feb. 2005	Software Developer www.aquasoft.ro
Mar. - Jul. 2004	Laboratory Instructor for the lecture "Computer System Structure" Politehnica University, Bucharest, Romania

Bibliography

[ABD06] Eugene Agichtein, Eric Brill, and Susan Dumais. Improving web search ranking by incorporating user behavior information. In *SIGIR*, 2006.

[AHHK10] Fabian Abel, Nicola Henze, Eelco Herder, and Daniel Krause. Interweaving public user profiles on the web. In *International Conference on User Modeling, Adaptation and Personalization (UMAP), Hawaii, USA*. Springer, June 2010.

[AHLP05] Nitin Agarwal, Ehtesham Haque, Huan Liu, and Lance Parsons. Research paper recommender systems: A subspace clustering approach. In *International Conference on Web-Age Information Management (WAIM)*, pages 475–491, 2005.

[BBdR10] Krisztian Balog, Marc Bron, and Maarten de Rijke. Category-based query modeling for entity search. In *Advances in Information Retrieval, 32nd European Conference on IR Research, ECIR 2010, Milton Keynes, UK, March 28-31, 2010. Proceedings*, pages 319–331, 2010.

[BCC$^+$06] Ingo Brunkhorst, Paul-Alexandru Chirita, Stefania Costache, Julien Gaugaz, Ekaterini Ioannou, Tereza Iofciu, Enrico Minack, Wolfgang Nejdl, and Raluca Paiu. The beagle++ toolbox: Towards an extendable desktop search architecture. In *Proceedings of the Semantic Desktop Workshop held at the 5th Intl. Semantic Web Conf*, 2006.

[BCSW07] Holger Bast, Alexandru Chitea, Fabian Suchanek, and Ingmar Weber. ESTER: efficient search on text, entities, and relations. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 671–678, New York, NY, USA, 2007. ACM.

[BDF$^+$10a] Bodo Billerbeck, Gianluca Demartini, Claudiu S. Firan, Tereza Iofciu, and Ralf Krestel. Exploiting click-through data for entity retrieval. In *Proceedings of the 33rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2010, Geneva, Switzerland, July 19-23, 2010*, pages 168–169. ACM, July 19–23 2010.

[BDF$^+$10b] Bodo Billerbeck, Gianluca Demartini, Claudiu S. Firan, Tereza Iofciu, and Ralf Krestel. Ranking Entities Using Web Search Query Logs. In *ECDL '10: Proceedings of the 14th European Conference on Research and Advanced Technology for Digital Libraries*, Berlin, Heidelberg, September 6–10 2010. Springer-Verlag.

[BdVCS07] Peter Bailey, Arjen P. de Vries, Nick Craswell, and Ian Soboroff. Overview of the TREC 2007 Enterprise Track. In Ellen M. Voorhees and Lori P. Buckland, editors, *Proceedings of The Sixteenth Text REtrieval Conference, TREC 2007, Gaithersburg, Maryland, USA, November 5-9, 2007*, volume Special Publication 500-274. National Institute of Standards and Technology (NIST), 2007.

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

[BHK98] John S. Breese, David Heckerman, and Carl M. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *UAI*, pages 43–52, Madison, USA, July 1998. Morgan Kaufmann Publisher.

[BIN$^+$07] Alessandro Bozzon, Tereza Iofciu, Wolfgang Nejdl, Antonio Vincenzo Taddeo, and Sascha Tönnies. Role based access control for the interaction with search engines. In *Proceedings of the 1st International Workshop on Collaborative Open Environments for Project-Centered Learning, COOPER*, 2007.

[BINT07] Alessandro Bozzon, Tereza Iofciu, Wolfgang Nejdl, and Sascha Tönnies. Integrating databases, search engines and web applications: A model-driven approach. In *Proceedings of Web Engineering, 7th International Conference, ICWE 2007,*, pages 210–225, Como, Italy, 2007.

[BSB08] Paolo Bouquet, Heiko Stoermer, and Barbara Bazzanella. An Entity Name System (ENS) for the Semantic Web. In *The Semantic Web: Research and Applications, 5th European Semantic Web Conference, ESWC 2008, Tenerife, Canary Islands, Spain, June 1-5, 2008, Proceedings*, volume 5021 of *Lecture Notes in Computer Science*, pages 258–272. Springer, 2008.

[BSTH07] Paolo Bouquet, Heiko Stoermer, Giovanni Tummarello, and Harry Halpin, editors. *Proceedings of the WWW2007 Workshop I^3: Identity, Identifiers, Identification, Entity-Centric Approaches to Information and Knowledge Management on the Web, Banff, Canada, May 8, 2007*, volume 249 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2007.

[BTD06] Wolf-Tilo Balke, Uwe Thaden, and Jörg Diederich. The Semantic GrowBag Demonstrator for Automatically Organizing Topic Facets. In *Proceedings of SIGIR2006 Workshop on Faceted Search*, Seattle, USA, August 2006.

[BYRN99] Ricardo A. Baeza-Yates and Berthier A. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press / Addison-Wesley, 1999.

[BYT07] Ricardo Baeza-Yates and Alessandro Tiberi. Extracting semantic relations from query logs. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, California, USA, August 12-15, 2007*, pages 76–85, 2007.

[CC07] Tao Cheng and Kevin Chen-Chuan Chang. Entity Search Engine: Towards Agile Best-Effort Information Integration over the Web. In *CIDR 2007, Third Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, January 7-10, 2007, Online Proceedings*, pages 108–113. www.crdrdb.org, 2007.

[CC09] Francesca Carmagnola and Federica Cena. User identification for cross-system personalisation. *Information Sciences: an International Journal*, 179(1-2):16–32, 2009.

[CDGI08] Nick Craswell, Gianluca Demartini, Julien Gaugaz, and Tereza Iofciu. L3S at inex 2008: Retrieving entities using structured information. In *Advances in Focused Retrieval, 7th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2008, Dagstuhl Castle, Germany, December 15-18, 2008. Revised and Selected Papers*, pages 253–263, 2008.

[CDNS05] Paul-Alexandru Chirita, Andrei Damian, Wolfgang Nejdl, and Wolf Siberski. Search Strategies for Scientific Collaboration Networks. In *Proceedings of 2nd P2P Information Retrieval Workshop held at the 14th ACM International Conference on Information and Knowledge Management (CIKM)*, Bremen, Germany, 2005.

[CINZ06] Sergey Chernov, Tereza Iofciu, Wolfgang Nejdl, and Xuan Zhou. Extracting semantic relationships between Wikipedia categories. In *1st International Workshop: SemWiki2006 - From Wiki to Semantics (SemWiki 2006), co-located with the ESWC2006 in Budva*, 2006.

[CRF03] William Cohen, Pradeep Ravikumar, and Stephen Fienberg. A comparison of string distance metrics for matching names and records. In *KDD Workshop on Data Cleaning, Record Linkage, and Object Consolidation*, 2003.

[CS07] Nick Craswell and Martin Szummer. Random walks on the click graph. In *SIGIR 2007: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Amsterdam, The Netherlands, July 23-27, 2007*, pages 239–246, 2007.

[CTC05] Kevyn Collins-Thompson and Jamie Callan. Query expansion using random walk models. In *Proceedings of the 2005 ACM CIKM International Conference on Information and Knowledge Management, Bremen, Germany, October 31 - November 5, 2005*, pages 704–711, 2005.

[CW07] Silviu Cucerzan and Ryen W. White. Query suggestion based on user landing pages. In *SIGIR 2007: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Amsterdam, The Netherlands, July 23-27, 2007*, pages 875–876, 2007.

[CYC07] Tao Cheng, Xifeng Yan, and Kevin Chen-Chuan Chang. EntityRank: Searching Entities Directly and Holistically. In *VLDB '07: Proceedings of the 33rd international conference on Very large data bases*, pages 387–398. VLDB Endowment, 2007.

[DAD05] Ronald Denaux, Lora Aroyo, and Vania Dimitrova. An approach for ontology-based elicitation of user models to enable personalization on the semantic web. In *Proceedings of the 14th international conference on World Wide Web, WWW 2005, Chiba, Japan, May 10-14, 2005 - Special interest tracks and posters*, pages 1170–1171, New York, NY, USA, 2005. ACM Press.

[DdVIZ08] Gianluca Demartini, Arjen P. de Vries, Tereza Iofciu, and Jianhan Zhu. Overview of the inex 2008 entity ranking track. In *Advances in Focused Retrieval, 7th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2008, Dagstuhl Castle, Germany, December 15-18, 2008. Revised and Selected Papers*, 2008.

[DFG$^+$09] Gianluca Demartini, Claudiu S. Firan, Mihai Georgescu, Tereza Iofciu, Ralf Krestel, and Wolfgang Nejdl. An architecture for finding entities on the web. In *2009 Latin American Web Congress, Joint LA-WEB/CLIHC Conference, Merida, Yucatan, Mexico, 9-11 November 2009*, pages 230–237, 2009.

[DFI07] Gianluca Demartini, Claudiu S. Firan, and Tereza Iofciu. L3S at inex 2007: Query expansion for entity ranking using a highly accurate ontology. In *Focused Access to XML Documents, 6th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2007, Dagstuhl Castle, Germany, December 17-19, 2007. Selected Papers*, pages 252–263, 2007.

[DFI$^+$08] Gianluca Demartini, Claudiu S. Firan, Tereza Iofciu, Ralf Krestel, and Wolfgang Nejdl. A Model for Ranking Entities and Its Application to Wikipedia. In *LA-WEB '08: Proceedings of the 2008 Latin American Web Conference*, pages 29–38, Washington, DC, USA, 2008. IEEE Computer Society.

[DFIN08] Gianluca Demartini, Claudiu S. Firan, Tereza Iofciu, and Wolfgang Nejdl. Semantically Enhanced Entity Ranking. In *WISE '08: Proceedings of the 9th international conference on Web Information Systems Engineering*, pages 176–188, Berlin, Heidelberg, 2008. Springer-Verlag.

[DG06] Ludovic Denoyer and Patrick Gallinari. The Wikipedia XML corpus. *SIGIR Forum*, 40(1):64–69, 2006.

[DI06] Jörg Diederich and Tereza Iofciu. Finding communities of practice from user profiles based on folksonomies. In *Proceedings of the 1st International Workshop on Building Technology Enhanced Learning solutions for Communities of Practice (TEL-CoPs06), co-located with the First European Conference on Technology-Enhanced Learning*, 2006.

[dVVT$^+$08] Arjen P. de Vries, Anne-Marie Vercoustre, James A. Thom, Nick Craswell, and Mounia Lalmas. 0verview of the INEX 2007 Entity Ranking Track. In *Focused Access to XML Documents: 6th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2007 Dagstuhl Castle, Germany, December 17-19, 2007. Selected Papers*, pages 245–251, Berlin, Heidelberg, 2008. Springer-Verlag.

[EIV07] Ahmed K. Elmagarmid, Panagiotis G. Ipeirotis, and Vassilios S. Verykios. Duplicate record detection: A survey. *Knowledge and Data Engineering, IEEE Transactions on*, 19(1):1–16, 2007.

[FL10] Lujun Fang and Kristen LeFevre. Privacy wizards for social networking sites. In *WWW '10: Proceedings of the 19th international conference on World wide web*, pages 351–360, New York, NY, USA, 2010. ACM.

[FNP07] Claudiu S. Firan, Wolfgang Nejdl, and Raluca Paiu. The Benefit of Using Tag-based Profiles. In *Proc. of 2007 Latin American Web Conference (LA-WEB '07)*, pages 32–41, Washington, DC, USA, 2007. IEEE Computer Society.

[GXCL09] Jiafeng Guo, Gu Xu, Xueqi Cheng, and Hang Li. Named entity recognition in query. In *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2009, Boston, MA, USA, July 19-23, 2009*, pages 267–274, 2009.

[HCOC02] Zan Huang, Wingyan Chung, Thian-Huat Ong, and Hsinchun Chen. A graph-based recommender system for digital library. In *Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries*, pages 65–73. ACM New York, NY, USA, 2002.

[HJSS06a] Andreas Hotho, Robert Jäschke, Christoph Schmitz, and Gerd Stumme. Information retrieval in folksonomies: Search and ranking. In *Proc. of the 3rd European Semantic Web Conference*, volume 4011 of *LNCS*, pages 411–426, Budva, Montenegro, June 2006. Springer.

[HJSS06b] Andreas Hotho, Robert Jschke, Christoph Schmitz, and Gerd Stumme. 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*, Aalborg, Denmark, July 2006. Aalborg University Press.

[HKR02] Jonathan L. Herlocker, Joseph A. Konstan, and John Riedl. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Inf. Retr.*, 5(4):287–310, 2002.

[ID09] Tereza Iofciu and Gianluca Demartini. Time based tag recommendation using direct and extended users sets. In *ECML PKDD Discovery Challenge 2009 (DC09)*, volume 497, pages 99–107, Bled, Slovenia, September 2009.

[IDB07] Tereza Iofciu, Jörg Diederich, and Wolf-Tilo Balke. Expertfoaf recommends experts. In *1st ExpertFinder Workshop, Berlin,Germany*, 2007.

[IDCdV11] Tereza Iofciu, Gianluca Demartini, Nick Craswell, and Arjen P. de Vries. Refer: Effective relevance feedback for entity ranking. In *Advances in Information Retrieval, 33th European Conference on IR Research, ECIR 2011, Dublin, Ireland, April 2011. Proceedings*, pages 264–276, 2011.

[IFAB11] Tereza Iofciu, Peter Fankhauser, Fabian Abel, and Kerstin Bischoff. Identifying users across social tagging systems. In *ICSWM'11 - Fifth International AAAI Conference on Weblogs and Social Media*, 2011.

[IKNP05] Tereza Iofciu, Christian Kohlschütter, Wolfgang Nejdl, and Raluca Paiu. Keywords and RDF Fragments: Integrating Metadata and Full-Text Search in Beagle++. In *Proceedings of Semantic Desktop Workshop at the ISWC*, 2005.

[ISC09] Tereza Iofciu, Milad Shokouhi, and Nick Craswell. Evaluating the impact of snippet highlighting in search. In *Understanding the user - Logging and interpreting user interactions in information retrieval. Workshop in Conjunction with the ACM SIGIR Conference on Information Retrieval*, 2009.

[Jar89] Matthew A. Jaro. Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida. *Journal of the American Statistical Association*, 84(406):414–420, June 1989.

[JMH⁺08] Robert Jäschke, Leandro Balby Marinho, Andreas Hotho, Lars Schmidt-Thieme, and Gerd Stumme. Tag recommendations in social bookmarking systems. *AI Commun.*, 21(4):231–247, 2008.

[JRMG06] Rosie Jones, Benjamin Rey, Omid Madani, and Wiley Greiner. Generating query substitutions. In *Proceedings of the 15th international conference on World Wide Web, WWW 2006, Edinburgh, Scotland, UK, May 23-26, 2006*, pages 387–396, 2006.

[JWR00] Karen Spärck Jones, Steve Walker, and Stephen E. Robertson. A probabilistic model of information retrieval: development and comparative experiments. parts 1 and 2. *Information Processing and Management*, 36:779–840, 2000.

[KT09] Ravi Kumar and Andrew Tomkins. A characterization of online search behavior. *IEEE Data Eng. Bull.*, 2009.

[LC01] Victor Lavrenko and W. Bruce Croft. Relevance-based language models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 120–127, 2001.

[Lev66] Vladimir Iosifovich Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707–710, 1966. Doklady Akademii Nauk SSSR, V163 No4 845-848 1965.

[LF06] Patrick Lehti and Peter Fankhauser. Unsupervised duplicate detection using sample non-duplicates. *Journal on Data Semantics VII*, 4244:136–164, 2006.

[Lip08] Marek Lipczak. Tag recommendation for folksonomies oriented towards individual users. In *Proceedings of ECML PKDD Discovery Challenge (RSDC08)*, pages 84–95, 2008.

[LZ03] John Lafferty and Chengxiang Zhai. Probabilistic relevance models based on document and query generation. *Language Modeling for Information Retrieval*, 13, 2003.

[MAC$^+$02] Sean M. McNee, Istvan Albert, Dan Cosley, Prateep Gopalkrishnan, Shyong K. Lam, Al Mamunur Rashid, Joseph A. Konstan, and John Riedl. On the recommendation of citations for research papers. In *CSCW*, New Orleans, USA, 2002.

[McC05] Andrew McCallum. Information extraction: Distilling structured data from unstructured text. *Queue*, 3(9):48–57, 2005.

[Mic07] Elke Michlmayr. Learning user profiles from tagging data and leveraging them for personal(ized) information access. In *In Proceedings of the Workshop on Tagging and Metadata for Social Information Organization, 16th International World Wide Web Conference (WWW2007*, 2007.

[Mis06] Gilad Mishne. Autotag: a collaborative approach to automated tag assignment for weblog posts. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 953–954, New York, NY, USA, 2006. ACM.

[MLdlR03] Miquel Montaner, Beatriz López, and Josep Lluís de la Rosa. A taxonomy of recommender agents on the internet. *Artificial Intelligence Review*, 19:285–330, 2003.

[MNBD06] Cameron Marlow, Mor Naaman, Danah Boyd, and Marc Davis. HT06, tagging paper, taxonomy, Flickr, academic article, to read. In *Proc. of the 17th Conf. on Hypertext and Hypermedia*, pages 31–40. ACM Press, 2006.

[MSR04] S.E. Middleton, N.R. Shadbolt, and D.C. De Roure. Ontological User Profiling in Recommender Systems. In *ACM Transactions on Information Systems (TOIS)*, pages 54–88. ACM Press, 2004.

[MVGD10] Alan Mislove, Bimal Viswanath, Krishna P. Gummadi, and Peter Druschel. You are who you know: inferring user profiles in online social networks. In *WSDM '10: Proceedings of the third ACM international conference on Web search and data mining*, pages 251–260, New York, NY, USA, 2010. ACM.

[O'R05] Tim O'Reily. What is Web 2.0? - Design Patterns and Business Models for the Next Generation of Software, September 2005.

[PBMW98] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. 1998.

[PMPG04] Dmitry Pavlov, Eren Manavoglu, David M. Pennock, and C. Lee Giles. Collaborative filtering with maximum entropy. *IEEE Intelligent Systems*, 19(6):40–48, 2004.

[PVT08] Jovan Pehcevski, Anne-Marie Vercoustre, and James A. Thom. Exploiting Locality of Wikipedia Links in Entity Ranking. In *Advances in Information Retrieval , 30th European Conference on IR Research, ECIR 2008, Glasgow, UK, March 30-April 3, 2008. Proceedings*, pages 258–269. Springer, 2008.

[RL03] Ian Ruthven and Mounia Lalmas. A Survey on the Use of Relevance Feedback for Information Access Systems. *Knowl. Eng. Rev.*, 18(2):95–145, 2003.

[RSH08] Henning Rode, Pavel Serdyukov, and Djoerd Hiemstra. Combining document- and paragraph-based entity ranking. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2008, Singapore, July 20-24, 2008*, pages 851–852, 2008.

[RV97] Paul Resnick and Hal R. Varian. Recommender systems - introduction to the special section. *CACM*, 40(3):56–58, 1997.

[SAC$^+$08] Martin Szomszor, Harith Alani, Iván Cantador, Kieron O'Hara, and Nigel Shadbolt. Semantic modelling of user interests based on cross-folksonomy analysis. In *Proc. International Semantic Web Conference (ISWC'08)*, pages 632–648, 2008.

[SCA08] Martin Szomszor, Iván Cantador, and Harith Alani. Correlating user profiles from multiple folksonomies. In *HT'08: Proceedings of the Nineteenth ACM Conference on Hypertext and Hypermedia*, pages 33–42. ACM, 2008.

[Seb02] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1):1–47, 2002.

[SKKR00] Badrul M. Sarwar, George Karypis, Joseph A. Konstan, and John Riedl. Analysis of recommendation algorithms for e-commerce. In *ACM Conference on Electronic Commerce*, pages 158–167, Minneapolis, USA, 2000.

[SKW07] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In Carey L. Williamson, Mary Ellen Zurko, Peter F. Patel-Schneider, and Prashant J. Shenoy, editors, *Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007*, pages 697–706. ACM, 2007.

[SvZ08] Börkur Sigurbjörnsson and Roelof van Zwol. Flickr tag recommendation based on collective knowledge. In *Proceedings of the 17th International Conference on World Wide Web, WWW 2008, Beijing, China, April 21-25, 2008*, pages 327–336. ACM Press, 2008.

[SW81] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of molecular biology*, 147(1):195–197, March 1981.

[TIR$^+$08] Nina Tahmasebi, Tereza Iofciu, Thomas Risse, Claudia Niederée, and Wolf Siberski. Terminology evolution in web archiving: Open issues. In *8th International Web Archiving Workshop, Aaarhus, Denmark, 18th & 19th Sep. 2008*, 2008.

[TSR$^+$08] Theodora Tsikrika, Pavel Serdyukov, Henning Rode, Thijs Westerveld, Robin Aly, Djoerd Hiemstra, and Arjen P. de Vries. Structured Document Retrieval, Multimedia Retrieval, and Entity Ranking Using PF/Tijah. In *Focused Access to XML Documents: 6th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2007 Dagstuhl Castle, Germany, December 17-19, 2007. Selected Papers*, pages 306–320, Berlin, Heidelberg, 2008. Springer-Verlag.

[VHS09] Jan Vosecky, Dan Hong, and Vincent Y. Shen. User identification across social networks using the web profile and friend network. In *Networked Digital Technologies, 2009. NDT '09. First International Conference on*, pages 360 –365, 28-31 2009.

[VTP08] Anne-Marie Vercoustre, James A. Thom, and Jovan Pehcevski. Entity Ranking in Wikipedia. In *Proceedings of the 2008 ACM Symposium on Applied Computing (SAC), Fortaleza, Ceara, Brazil, March 16-20, 2008*, 2008.

[VZ08] David Vallet and Hugo Zaragoza. Inferring the most important types of a query: a semantic approach. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2008, Singapore, July 20-24, 2008*, pages 857–858, 2008.

[WHK$^+$10] Gilbert Wondracek, Thorsten Holz, Engin Kirda, Sophia Antipolis, and Christopher Kruegel. A practical attack to de-anonymize social network users. In *IEEE Symposium on Security and Privacy*, 2010.

[XFMS06] Zhichen Xu, Yun Fu, Jianchang Mao, and Difu Su. Towards the semantic web: Collaborative tag suggestions. In *WWW2006: Proceedings of the Collaborative Web Tagging Workshop*, Edinburgh, Scotland, 2006.

[YKA08] Emine Yilmaz, Evangelos Kanoulas, and Javed A. Aslam. A simple and efficient sampling method for estimating AP and NDCG. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 603–610, New York, NY, USA, 2008. ACM.

[ZdVDI08] Jianhan Zhu, Arjen P. de Vries, Gianluca Demartini, and Tereza Iofciu. Relation Retrieval for Entities and Experts. In *Future Challenges in Expertise Retrieval (fCHER 2008), SIGIR 2008 Workshop*, 2008.

[ZL04] Chengxiang Zhai and John Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst.*, 22(2):179–214, 2004.

[ZL09] Reza Zafarani and Huan Liu. Connecting corresponding identities across communities. In *Third International ICWSM Conference*, pages 354–357, 2009.

[ZRM+07] Hugo Zaragoza, Henning Rode, Peter Mika, Jordi Atserias, Massimiliano Ciaramita, and Giuseppe Attardi. Ranking very many typed Entities on Wikipedia. In *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 1015–1018, New York, NY, USA, 2007. ACM.