USING CONTEXTUAL AND SOCIAL LINKS IN INFORMATION RETRIEVAL

Von der Fakultät für Elektrotechnik und Informatik

der Gottfried Wilhelm Leibniz Universität Hannover

zur Erlangung des Grades

Doktor der Naturwissenschaften

Dr. rer. nat.

genehmigte Dissertation von

Master of Science Sergey Chernov

geboren am 17 September 1980 in Wolgograd, Russische Föderation

Hannover, Deutschland, July 2012

Referent: Prof. Dr. Wolfgang Nejdl
Ko-Referent: Prof. Dr. Heribert Vollmer
Tag der Promotion: 03 July 2012

ABSTRACT

The amount of information on the Web, in enterprise networks and on local computers is huge and continues to grow. A search engine is no longer a specific tool for a knowledge worker, but a routine Web site for everyone. The available ranking algorithms are normally based on textual content and hyperlinks, but with the recent boom of social platforms like Facebook or Twitter the social links come into play. Such social links become the third most important dimension for relevance calculation in future search engines.

The majority of social links are missing when it comes to search on a single desktop. However, the context data regarding recently viewed files, email interactions or switching between applications could be converted into contextual links representing virtual connections between pieces of information. In this thesis, we investigate how both social and contextual links could improve user's search experience. We provide a number of algorithms and applications, and support our findings with experimental results.

We start with considering contextual and social links in the desktop search domain. We elaborated on the need for a common collection for desktop search evaluation and describe the design of such a dataset. Next, we develop the necessary logging tools, characterize a sample activity-based desktop dataset and analyze observed user behavior. In addition, we consider the problem of personal resources being stored across Web 2.0 platforms. We address it with the desktop search application that combines a single overview from results found on a desktop with results from social networks like Flickr, YouTube, and Delicious. The effectiveness of this method is supported by a user study.

We continue with a study of social links in the enterprise settings and present our solution to personalized social search. Our algorithm uses the user's social relations and search results are re-ranked according to connections with individuals in the user's social network. We evaluate the effectiveness of several types of social networks for personalization by an off-line experiment and by a user study. We show that in both experimental setups social network based personalization significantly outperform topic-based personalization and non-personalized social search.

Finally, we exploit search and recommendation in social networks on the Web. First, we propose a new method to identify landmark photos using tags and social Flickr groups. A user study shows that the proposed method outperforms state-of-the-art systems for landmark finding. Next, we consider a wider set of Web 2.0 applications and present a mobile search application that provides user with overview of landmark resources from social sites like Flickr, YouTube, or Delicious. In addition, we introduce the novel problem of recommending links to users of microblogging platforms. We study hyperlink recommendation based on two types of social connections and propose two algorithms. The evaluation on the Twitter data shows that recommendation based on social information alone achieves high accuracy level.

In this thesis we show that both social and contextual links provide a valuable source for different search tasks. Based on these links, our algorithms and applications improve overall search effectiveness in several information retrieval domains.

Keywords: *Information Retrieval, Personalization, Web 2.0*

ZUSAMMENFASSUNG

Die Menge an Informationen im Web, in Firmennetzwerken und auf lokalen Computern ist riesig und wächst weiter. Eine Suchmaschine ist nicht mehr nur ein bestimmtes Tool für einen Experten, sondern eine Webseite für jedermann. Die zur Verfügung stehenden Ranking-Algorithmen waren in der Regel auf Textinhalte und Hyperlinks beschränkt, aber mit dem Boom der sozialen Plattformen wie Facebook oder Twitter werden soziale Links immer wichtiger. Solche sozialen Links werden in Zukunft die drittwichtigste Dimension für die Berechnung der Relevanz für Suchergebnisse darstellen.

Die Mehrheit der sozialen Links fehlt, wenn bei der Suche nur ein einziger Desktop zur Verfügung steht. Allerdings könnten Kontext-Informationen wie zuletzt angezeigte Dateien, E-Mail-Interaktionen oder das Umschalten zwischen Anwendungen als kontextuelle Verknüpfungen angesehen werden. In dieser Arbeit untersuchen wir, wie soziale und kontextuelle Links Benutzern die Suche vereinfachen kann.

Wir beginnen mit kontextuellen und sozialen Links im Bereich der Desktop-Suche. Desweiteren erläutern wir die Notwendigkeit einer gemeinsamen Testkollektion für Desktop-Suche und beschreiben den Aufbau einer solchen. Als Nächstes beschreiben wir die notwendigen Logging-Werkzeuge, charakterisieren einen auf Aktivität basierenden Desktop-Datensatz und analysieren beobachtetes Nutzerverhalten. Darüber hinaus betrachten wir das Speichern persönlicher Ressourcen in Web 2.0-Plattformen. Wir schlagen dafür eine Desktop-Suche für, welche die Ergebnisse einer Desktop-Suche mit Ergebnissen von sozialen Netzwerken wie Flickr, YouTube und Delicious kombiniert. Die Wirksamkeit dieser Methode wird in einer Benutzerstudie belegt.

Desweiteren analysieren wir soziale Links in Unternehmensnetzwerken und präsentieren unsere Lösung für personalisierte soziale Suche. Unser Algorithmus verwendet die sozialen Links der Benutzer zusammen mit den Suchergebnissen basierend auf den Verbindungen der Benutzer des sozialen Netzwerks. Wir bewerten die Effektivität der Personalisierung bei verschiedenen Arten von sozialen Netzwerken mit Hilfe von Simulationen und Benutzerstudien. Wir zeigen, dass in beiden Versuchsanordnungen Personalisierung basierend auf sozialen Netzwerken die Themen-basierte Personalisierung und nicht-personalisierte soziale Suche signifikant übertrifft.

Als letzten Punkt betrachten wir Suche und Empfehlungen in sozialen Netzwerken außerhalb von Unternehmensnetzwerken. Zunächst schlagen wir eine neue Methode zur Erkennung von Wahrzeichen auf Fotos vor basierend auf Tags und sozialen Gruppen in Flickr. Eine Benutzerstudie zeigt, dass das vorgeschlagene Verfahren andere Systeme zur Erkennung von Wahrzeichen übertrifft. Anschließend betrachten wir eine breitere Palette von Web 2.0-Anwendungen und stellen eine mobile Suche-Anwendung vor, die Benutzern einen Überblick von Wahrzeichen aus sozialen Netzwerken wie Flickr, YouTube oder Delicious bieten. Darüber hinaus stellen wir das neue Problem der Link-Empfehlung in Microblogging-Plattformen wie Twitter vor. Wir studieren Hyperlink Empfehlungen basierend auf zwei Arten von sozialen Verknüpfungen und schlagen zwei Algorithmen dafür vor. Die Auswertung der Daten zeigt, dass Twitter Empfehlungen nur basierend auf sozialen Informationen gute Ergebnisse erzielt. In dieser Arbeit zeigen wir, dass sowohl soziale als auch kontextuelle Links eine wertvolle Quelle für verschiedene Such-Aufgaben darstellen. Basierend auf diesen Links erzielen unsere Algorithmen und Anwendungen eine Verbesserung in mehreren Anwendungsbereichen der Informationssuche.

Schlagwörter: *Information Retrieval, Personalization, Web 2.0*

Published Work

The algorithms presented in this thesis have been published at various conferences and workshops. Below we provide a list of our publications, arranged according to the thesis structure. In Chapter 3 we describe personalization in desktop search based on the following contributions:

- *deskWeb2.0: Combining Desktop and Social Search*. Sergej Zerr, Elena Demidova and Sergey Chernov. In Desktop Search Workshop at SIGIR 2010, Geneva, Switzerland, July 23, 2010, pp. 9-12. [Zerr *et al.*, 2010]

- *Enterprise and Desktop Search*. Pavel Dmitriev, Pavel Serdyukov and Sergey Chernov. In Proceedings of the 19th International Conference on World Wide Web, WWW 2010, Raleigh, North Carolina, USA, April 26-30, 2010, pp. 1345-1346, 978-1-60558-799-8. [Dmitriev *et al.*, 2010]

- *Task Detection for Activity-Based Desktop Search*. Sergey Chernov. In Proceedings of the 31th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2008, Singapore, July 20-24, 2008, p. 894, 978-1-60558-164-4. [Chernov, 2008]

- *Evaluating Personal Information Management Using an Activity Logs Enriched Desktop Dataset*. Sergey Chernov, Gianluca Demartini, Eelco Herder, Michal Kopycki, and Wolfgang Nejdl. In Personal Information Management Workshop at CHI 2008, PIM 2008, Florence, Italy, April 5-6, 2008. [Chernov *et al.*, 2008]

- *Converting Desktop into a Personal Activity Dataset*. Sergey Chernov, Enrico Minack, and Pavel Serdyukov. In Proceedings of 8th Nationational Russian Conference on Digital Libraries, RCDL 2007, Pereslavl, Russia, October 15-18, 2007. [Chernov *et al.*, 2007a]

- *Building a Desktop Search Test-bed*. Sergey Chernov, Pavel Serdyukov, Paul-Alexandru Chirita, Gianluca Demartini, and Wolfgang Nejdl. In Proceedings of the 29th European Conference on Information Retrieval, ECIR 2007, Rome, Italy, April 2-5, 2007, pp. 686-690, 978-3-540-71494-1. [Chernov *et al.*, 2007b]

Chapter 4 presents the personalization in enterprise search and refers to work published in:

- *Personalized Social Search Based on the User's Social Network*. David Carmel, Naama Zwerdling, Ido Guy, Shila Ofek-Koifman, Nadav Har'El, Inbal Ronen, Erel Uziel, Sivan Yogev, and Sergey Chernov. In Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM 2009, Hong Kong, China, November 2-6, 2009, pp. 1227-1236, 978-1-60558-512-3. [Carmel *et al.*, 2009]

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

- *Guideme! the world of sights in your pocket*. Sergej Zerr, Kerstin Bischoff, and Sergey Chernov. In Proceedings of the 27th International Conference on Data Engineering, ICDE 2011, Hannover, Germany, April 11-16, 2011, pp. 1348-1351. [Zerr *et al.*, 2011]

- *Exploiting Flickr Tags and Groups for Finding Landmark Photos*. Rabeeh Abbasi, Sergey Chernov, Wolfgang Nejdl, Raluca Paiu, Steffen Staab. In Proceedings of the 31st European Conference on Information Retrieval, ECIR 2009, Toulouse, France, April 6-9, 2009, pp. 654-661, 978-3-642-00957-0. [Abbasi *et al.*, 2009a]

In the beginning of the Ph.D. studies I published several papers addressing problems of personalized distributed information retrieval, expert finding and link analysis in Wikipedia. They have a slightly different focus so I do not include them in this work. But these publications helped me to shape ideas for the thesis, so I want to mention them as follows:

- *Enhancing Expert Search through Query Modeling*. Pavel Serdyukov, Sergey Chernov, and Wolfgang Nejdl. In Proceedings of 29th European Conference on Information Retrieval, ECIR 2007, Rome, Italy, April 2-5, 2007, pp. 737-740, 978-3-540-71494-1. [Serdyukov *et al.*, 2007]

- *A Plugin Architecture Enabling Federated Search for Digital Libraries*. Sergey Chernov, Christian Kohlschuetter, and Wolfgang Nejdl. In Proceedings of 9th International Conference on Asian Digital Libraries, ICADL 2006, Kyoto, Japan, November 27-30, 2006, pp. 202-211, 3-540-49375-1. [Chernov *et al.*, 2006c]

- *Extracting Semantic Relationships between Wikipedia Categories*. Sergey Chernov, Tereza Iofciu, Wolfgang Nejdl, and Xuan Zhou. In Proceedings of 1st International Workshop: "SemWiki2006 - From Wiki to Semantics" at ESWC 2006, SemWiki 2006, Budva, Montenegro, June 12, 2006. [Chernov *et al.*, 2006b]

- *L3S Research Center at TREC 2006 Enterprise Track.* Sergey Chernov and Gianluca Demartini and Julien Gaugaz. In Proceedings of the 15th Text REtrieval Conference, TREC 2006, Gaithersburg, Maryland, November 14-17, 2006. [Chernov *et al.*, 2006a]

- *Database selection and result merging in P2P Web search.* Sergey Chernov, Pavel Serdyukov, Matthias Bender, Sebastian Michel, Gerhard Weikum, and Christian Zimmer. In Proceedings 3rd International Workshop on Databases, Information Systems and Peer-to-Peer Computing, DBISP2P 2005, Trondheim, Norway, August 28-29, 2005., pp. 26-37, 978-3-540-71660-0. [Chernov *et al.*, 2005]

ACKNOWLEDGMENTS

Even PhD thesis is considered an individual work, it is not possible to complete it without support and help from your colleagues and friends. I would like to thank all the people who helped me in different ways during all these years.

Thanks to my supervisor Wolfgang Nejdl for his guidance and support in so many areas. His research knowledge and visionary ideas improved my work and constantly motivated me in a course of PhD studies. His advice, optimism and patience, helped me in research and in everyday life.

I am greatly thankful to my daily supervisor, Eelco Herder, who helped me every time when I needed it. He was always around to discuss my work, to provide a feedback on my writings or suggest a new solution when I got stuck. His invaluable input made it possible to complete this thesis.

Thanks to my dear friend and talented researcher Pavel Serdyukov, for all these years of friendship and fruitful collaboration. His performance and passion for quality always encouraged me to push forward in professional life.

I would like to thank David Carmel, Shila Ofek-Koifman, Nadav Har'el and all the people from the IBM Haifa Research Lab. Their scientific excellence, high professional standards and friendly environment made my research visit to Haifa an unforgettable experience.

Many thanks to my collaborators and colleagues Paul Chirita, Raluca Paiu, Michal Kopycki, Gianluca Demartini, Kerstin Bischoff, Rabeeh Abbasi, Elena Demidova, and Sergej Zerr. It was a pleasure to work with you all and this work would not be accomplished without your contributions.

I am greatly thankful for the friendship and support in surviving the PhD years to Tereza Iofciu, Julien Gaugaz, Ralf Krestel, Mohammad Alrifai, Stefania Costache, Claudiu Firan and all the L3S people, which are too many to mention.

And the biggest thanks and my sincere gratitude I address to my wife Olga Kulakova. Her love, patience and care, was the greatest help in my research work and personal life.

Contents

List of Figures

1 Introduction

"I am astounded by people who want to "know" the universe when it is hard enough to find your way around Chinatown."

Woody Allen

The amount of information on the Web, in enterprise networks and on local computers is huge and continues to grow. Millions of users download files from the Web, upload their photos and videos to social platforms, exchange messages in blogs and instant messengers. Information stored on the Web and Web 2.0 has become a shared knowledge pool for users around the world. The volume and variety of this data creates a tremendous value, but also makes it difficult to search for relevant information. A search engine is no longer a specific tool for a knowledge worker, but a routine Web site for everyone. Its usage is so widespread, that in 2006 the Oxford English Dictionary added a new verb *to google*, reffering to Google search engine and meaning obtaining information on the Web. The weather forecast, recent news about politics and celebrities, prices and offers, recommendations on new movies and books — all of it is just a tiny portion of what people are looking for.

The majority of the Web data is unstructured and does not follow a well defined data model. Moreover, an increased amount of multidemia resources like music and videos is difficult to index using traditional text-based approaches. Last but not least, while we observe a boom of interesting user-generated content we also face a problem of low-quality information being mixed information resources created by professionals. It severely complicates search and frustrate users

Despite increased importance of the Web in everyday information tasks, there is still a lot of data stored on a user's desktop. Thanks to large harddrive capacities a number of files on a single PC could be dozens of thousands. Traditional browsing for locating necessary information pieces is no longer sufficient and desktop search difficulties reinforce Web search problems.

All types of information sources including the Internet, intranets and desktops are using *ranking* to support users with fast and effective information access. The first generation of search engines in the 80-90s relied on the text features alone for document ranking, which was sufficient only in relatively small document collections and in absense of spam. When the linkage-based ranking algorithms like PageRank and HITS were proposed in 1998-1999, the search engines dramatically improved their performance and the second generation of search engines provided sufficient search quality for the last decade. Nowadays, with recent boom of social platforms like Facebook or Twitter the social links come into play. The distinct feature of social systems is that users explicitly specify connections to their friends, colleagues and people they like. Such social links are becoming the third most important dimension for relevance calculation in future search engines.

The available ranking algorithms are normally tuned to satisfy majority of users and biased towards most common users' preferences. This approach works good in general, but less effective for users with specific search requirements. An importance of *personalization* is recognized long ago and in recent years we observe a rising popularity of personalized search and recommendation algorithms. Personalized search engines look at user's previous searches, identify how information need of a particular user is different from others, and adapt search ranking accordingly. Similarly, recommender systems look at user's likes and dislikes, and recommend next songs to listen or news articles to read.

The personalization methods differ depending on the current search environment and its specific features. For example, analyzing search history of many users helps to extract user preferences and to produce a better ranking in the Web search. But when it comes to the desktop, this collective data is missing and one cannot blindly apply Web ranking algorithms to a desktop search system. Fortunately, the desktop contains context information like recently viewed files, Web visits history, email interactions and switching between applications. This data could be converted into contextual (activity) links representing virtual connections between consequently accessed files or regularly opened folders. In the large enterprise, where we observe both local hyperlinks and activity links together, we also have a complex of explicit social connections. An enterprise search could be adapted to a user's social network and benefit from combined knowledge of user's colleagues and peers.

1.1 Problems Addressed in this Thesis

Personalized ranking provides better search results considering user profile and currently performed task. While traditional text- and hyperlinks-based personalization approaches have been intensively studied in last years, just few works considered other important factors like user social and contextual links.

We would like to answer three research questions regarding application of social and contextual links in desktop (Problem 1), enterprise (Problem 2) and social networks on the Web (Problem 3). Taking into account specific features of each of these three domains, we address these problems separately in several targeted research projects.

Problem 1. How to improve desktop search using information regarding user's activities and social connections?

When we talk about resources on a desktop we often have to deal with numerous files with similar textual content. In such situation it is difficult to distinguish between relevant and non-relevant resources solely based on content. We have to consider other factors like file access or email exchange history, and convert them into contextual or, in other words, *activity* links. The activity links could help to improve search and navigation experience on a desktop.

We might also face the problem that some of the resources are uploaded on online sharing resources, while the user still wants to treat them as part of her personal data stored locally. So the problem is to collect activity data regarding user preferences and consider resources stored elsewhere but belonging to a personal information space. This problem might be solved using user's social links, which are scattered across different Web 2.0 platforms.

Problem 2. How to personalize enterprise search given a user's social links?

This scenario is typical for a large company in which a large enterprise network is available, including local blogging platforms, bookmarking and file sharing services, wiki and project pages, social networks, etc. A user could benefit from previous interactions with some of these resources, as her interests are implicitly reflected in assigned tags, bookmarks and social connections. Furthermore, social networking assumes that there are connected people with similar interests and preferences, so a user might re-use their knowledge and expertise to fulfill personal information need.

Problem 3. How to improve multimedia search and microblogs browsing using social links?

In the last scenario, we consider scarce textual description problem typical for multimedia objects and short micro-blogging platforms. While the number of available images or posts in micro-blog is extremely large, the associated text is usually extremely short and does not allow effective ranking. We would like to explore possibilities to enhance resource representation in such a case with information regarding user's social preferences and provide a better ranking as a result.

1.2 Our Contribution

Within this thesis we systematically study the application of social and contextual links in desktop, enterprise and social networks on the Web. We explore different ways to personalize user search and recommendation experience, present several novel algorithms and demonstrate achieved improvement in several user studies.

To address the scenario described in **Problem 1**, we consider user's interactions with files and applications. The problem with ranking of desktop data is complicated by absence of hyperlinks between files. However, user's switching between files, Web pages and emails could be seen as a natural indication of information importance and helps to identify items which belong to a particular task. We define a set of possible interesting activity features and develop a logging framework which collects necessary data. In the course of a user study, we create, characterize and analyze the activity-based dataset. The developed *Logging Framework* both provides a tool for experimental evaluation and helps in building a standard dataset for desktop search experiments. To support this claim we outline some external research on desktop task detection and activity-based ranking which used our logging framework.

We also consider the situation in which a user stores a part of personal data in some online Web 2.0 sharing services. We present the *GuideMe!* application, which combines results found on a desktop with results from social networks. The effectiveness of this method is supported by a small-scale user study.

Moving from a desktop scale to **Problem 2**, we hypothesise that social links could also be used in personalizing enterprise search. We systematically study how different aspects of user's social activities. When large information volume is used by many users there is a possibility to enrich user profile based on interests of other users. We introduce notions of social similarity and familiarity and show how they could be used to improve effectiveness of personalized social search over documents, people, bookmarks and tags. The algorithms and experiments presented in this chapter were completed during a research internship at IBM Haifa Research Lab.

To answer **Problem 3** we exploit social links in cases where limited text description is available and there are millions of potentially relevant items. We propose an algorithm for landmark finding using Flickr social groups. It identifies best photo matches for a landmark of interest given available Flickr groups. The experimental results from a user study confirmed that our algorithm outperformes existing approaches. To address a microblogging scenario, we experiment with the recommendation of links on Twitter using user's social network. We show how link recommendation could be enhanced with user's social connections and provide an experimental evaluation of our algorithms.

1.3 Thesis Outline

In the rest of this work, we present various approaches for personalized search and recommendation using user's social and activity links. The body of the thesis continues with Chapter 2, introducing general background information on information retrieval problems in different environments. Next, in Chapter 3 we present our approach for using desktop activity and social links to improve desktop search quality. In Chapter 4 we consider an intranet search scenario and report our work on personalizing social enterprise search. Moving on to a larger scale, in Chapter 5 we present our solutions to landmark finding and the problem of link recomendation in photo-sharing and microblogging social networks. Chapter 6 concludes the thesis with a summary of the accomplished work and possible future research directions.

2 General Background

We start with an introduction to information retrieval and personalization in several search domains considered in this thesis. We cover primarily Web, enterprise and desktop search areas. Here we provide reader with main definitions and pointers to relevant literature. The detailed research problems and related work for different search environments we discuss in subsequent chapters, corresponding to each retrieval domain.

2.1 Information Retrieval and Personalized Search

Information Retrieval (IR) comprises finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from large collections (usually stored on computers) [Manning *et al.*, 2008]. It deals with architectures of search engines, algorithms and methods used for information search in the Internet, digital libraries, and text databases. The main goal is to find the relevant documents for a query from a collection of documents. Usually, the documents are preprocessed and placed into an index, which provides the base for retrieval.

A typical search engine is based on a single database model of text retrieval [Witten *et al.*, 1999]. In the model, documents from the Web and local sources are collected into a centralized repository and indexed. The whole model is effective if the index is large enough to satisfy most of the users' information needs and if a search engine uses an appropriate retrieval system. A retrieval system is a set of retrieval algorithms for different purposes: ranking, stemming, index processing, relevance feedback and so on [Baeza-Yates and Ribeiro-Neto, 1999]. The effectiveness of search engines is typically evaluated using user studies, for a survey of evaluating methodologies see [Voorhees and Harman, 2005].

Regular search engines and portals provide the same results for different personalities, intentions and contexts. In advanced search tools, *personalization* is used to customize the Web for individuals by filtering out irrelevant information based on interests and context of individual [Brusilovsky *et al.*, 2007]. The goal of personalization is to provide users with what they need without requiring them to ask for it explicitly. A system generates useful, actionable knowledge about users called *user profile*, and uses it for personalizing the output.

User profiling is tightly connected to privacy problems [Kobsa, 2007b]. In general, people are comfortable sharing preferences, demographic and lifestyle information, but not comfortable sharing medical, financial or purchase related information. Having sensitive information indexed by search engines constantly raises public discussions. This problem has been partially addressed in research areas of Privacy Preserving Data Mining [Vaidya *et al.*, 2005] and Private Information Retrieval [Asonov, 2004], but in many respects this problem currently does not have a general solution.

In addition to personalized search, we observe the rising popularity of *recommender systems* [Jannach *et al.*, 2011]. These systems use information filtering to recommend information items like movies, news, images, or social elements like people, events and groups, that are likely to be of interest to the user. The recommender algorithms are using either *content based recommendation* or *collaborative filtering*. It provides a different view on personalization, as previously liked items or explicit user ratings are used to infer user interests.

An important factor to account is a user's information need. Different users pursue goals and have different search tasks. A commonly accepted taxonomy of such tasks has been proposed in [Broder, 2002], distinguishing between Web queries of three types:

- Navigational. User wants to open a particular Website.
- Informational. User collects information from one or several Web pages.
- Transactional. User wants to make a purchase or perform another Web-based transaction.

Besides query types, the search space could be divided into several environments like desktop search, enterprise search and Web search, see Figure 2.1. The smallest information domain is desktop search, which deals with files and emails from local, shared and removable data storages. The enterprise search represents a bigger set of documents available in intranet, corporate archives and databases, mailing lists and wiki discussions, profiles for people and different departments. The largest volume of data is publicly accessible on the Web and includes social networks and digital libraries, Web sites and different online services.

In the rest of this chapter we review some basic characteristics for each of these three domains, starting with desktop search.

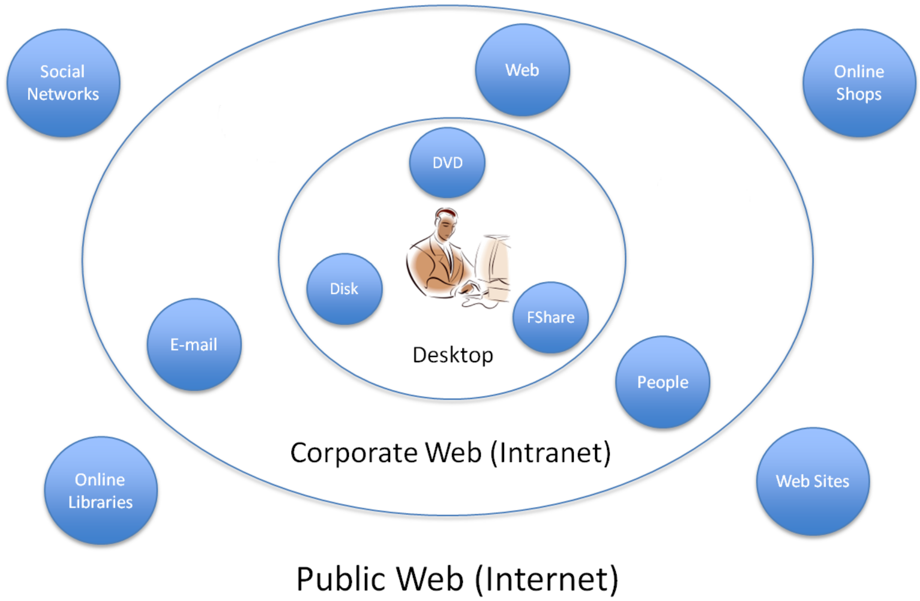


Figure 2.1 Search Domains

2.1.1 Desktop Search

Desktop search is the name for the field of tools that search the contents of a user's own computer files, rather than searching the Internet. These tools are designed to find information on the user's PC, including Web browser histories, e-mail archives, text documents, sound files, images and video [Baker *et al.*, 2007]. The desktop domain is a part of a more general field of Personal Information Management (PIM).

The beginning of Personal Information Management research is often attributed to Vannevar Bush's visionary paper from 1945 [Bush, 1945] and his description of hypothetical "Memex" device, which stores all his books, records and communications. The current definition of PIM could be characterized as following: "Personal Information Management is both the practice and the study of the activities people perform to acquire, organize, maintain, retrieve, use, and control the distribution of information items such as documents (paper-based and digital), Web pages, and email messages for everyday use to complete tasks (work-related and not) and to fulfill a person's various roles (as parent, employee, friend, member of community, etc.)." [Jones and Teevan, 2007]. While PIM covers many senses of the term "personal information", when we talk abour personalization we assume information directed to a person, rather than information that person keeps or has control of.

Problems typical for the desktop cover indexing and ranking issues. Indexing on a desktop requires integration across dozens applications and file formats. It also has to provide user-specific privacy management, as some personal information is highly sensitive and must not be searched. The ranking is different from Web search, since re-finding tasks are prevalent over search for unseen information. Also, as people are historically accustomed to browse desktop, many users still prefer to navigate through their files rather than to search [Teevan *et al.*, 2004]. Following an argument from [Freeman and Gelernter, 2007] on a similar study from [Barreau and Nardi, 1995], we believe that generalization of this observation into inherent user preference might be misleading. The current user preferences could be an indication of their poor experience with current desktop search tools and improved algorithms would encourage users to search desktop, rather than navigate.

A distinctive feature of the desktop is the rich context available, see Figure 2.2. A user context can be represented by location, people, activities, time, season, emotional state, etc. In this thesis, we focus on user activities, which could be inferred using implicit feedback like switching between applications and files, opening different files simultaneously, reaction time to read email or answer instant message, and so on.

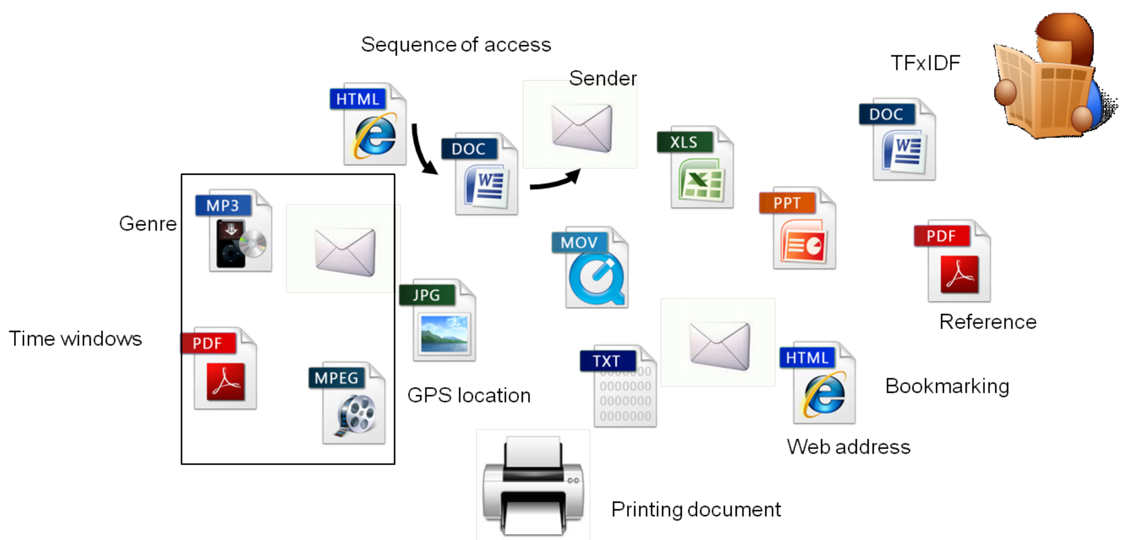


Figure 2.2 Context on a Desktop

The desktop context could be represented as a graph with nodes corresponding to files and emails, and links representing virtual connections between them. An example of such link could be a user switch from reading a document to answer a chat message or to open a different document. The social links on a desktop are inferred from user connections in address book, email messages or fetched from her accounts from different collaborative platforms. In Chapter 3 we investigate how both contextual and social links could be used to improve user search experience on a desktop.

2.1.2 Enterprise Search

With the increasing scale and complexity of intranets, *enterprise search* is becoming an increasingly important information retrieval domain. As described in [Hawking, 2004], enterprise search includes search of the organisations external Website, internal Website and other electronic text held in the form of email, database records, documents on fileshares and the like. This definition suggests a mixture of plain Web pages, semi-structured documents and structured databases. It also implies that user is searching for different entities. For example, a system could present a user a set of people from her social network. This is a ranked list of persons, who relate to the user either through co-authorship of a wiki page or used the same tags and commented on the same blog entry. The task of ranking persons for a given query is called *people search* or, when system is tuned to rank results based on people's competence in particular topic, *expert search* [Serdyukov, 2009].

The distinctive features of enterprise search stem from the fact that social forces driving the creation of content on the Web are different from those in the enterprise. This results in unique challenges for crawling, indexing, and ranking components of the search engine. First, an enterprise search engine crawls and indexes information from a variety of repositories in variety of formats. Second, it needs to rank different types of entities and simple ranking is often not enough, as users expect exploratory interfaces with rich navigation and filtering functionality.

One of the emerging attributes of an enterprise is a *social search* system. There are several alternative definitions of the concept social search [Hotho *et al.*, 2006b; Bender *et al.*, 2008; Amitay *et al.*, 2009]. It is used to describe several different aspects of search in Web 2.0 applications, like finding a path between users in a social network, or retrieving a set of blog posts relevant to a user. We consider social search as a search process over social data gathered from Web 2.0 applications, such as social bookmarking systems, wikis, blogs, forums and others. Such a social search system represents documents, persons, communities, and tags, together with their interrelations, and allows searching for all object types related to the user's query.

Social search provides an ideal testbed for personalization due to the explicit user interactions through Web 2.0 tools [Carman *et al.*, 2008]. A user profile that is derived from user feedback such as bookmarking, rating, commenting, and blogging, provides a very good indication of the user's interests. In Chapter 4 we show how this information could be used both for personalized social search and system evaluation.

2.1.3 Web Search and Web 2.0

A Web search engine is an information retrieval system for Web pages [Chakrabarti, 2002]. The capabilities of modern search engines are very broad: they allow queries on Web pages, photos, videos and more. General-purpose search engines can search across the whole Web, while special-purpose engines are concerned with the specific information sources or specific subjects.

The notion of *Web 2.0* is used to characterize a set of novel Web applications, which facilitate communication, knowledge sharing and collaborative work on the Web [Bozzon *et al.*, 2009]. Web 2.0 applications include tagging systems, photo and video sharing platforms, wikis and blogs. Such services encourage users to organize social groups, structured around some topic of interest or based on personal connections in the real world. This activity leads to a network of explicit social links, similar to the hyperlink structure of the Web.

The technical advances is one of the Web 2.0 attributes, but the real contribution is in conceptual changes in users' behaviour. Web 2.0 sites are used beyond simple information retrieval, as they allow users to run applications entirely through a browser. This way users are encouraged to update their content and improve the applications as they interact with them. As a result, the Web 2.0 provides rich user experience and active involvement of participants in data creation, commenting, rating and discussing content generated by other users.

One of the main Web 2.0 applications is *collaborative tagging platforms*, which allow users to create annotations for information items with a set of keywords called *tags* [Benz *et al.*, 2008]. The tags could describe time, place, name, event or any other metadata associated with a photo, blog post or video. As tags are picked freely by users, they often correlate with search keywords used later to locate this specific items and improve retrieval accuracy. Also, tags create a link structure across resources if several users are using similar tags on different information items. Since there is no single predefined taxonomy over tags, tagging platforms typically use social structures of the community users. Therefore, tags and other conceptual structures emerging in social systems are called *folksonomies* [Wal, 2007] and modeled as graphs.

One example Web 2.0 application is *Flickr*[1], one of the most popular photo sharing Website and online community platform. Flickr allows its users to describe images using tags and to search pictures using place name, subject, or other aspects of the picture. Another popular platform is *Twitter*[2], that became the top provider of online social networking and microblogging service. It lets its users send and read text-based posts of up to 140 characters, informally known as *tweets*. In Chapter 5 we propose several algorithms for improved search and recommendation in social networks, evaluating them on Flickr and Twitter data.

According to the size of each information domain, we structured our thesis from smallest to largest search area. In next chapter we study desktop environment and propose usage of contextual and social links for enhanced desktop search.

[1] www.flickr.com
[2] www.twitter.com

3 Contextual and Social Links in Desktop Search

In this chapter we consider the desktop search domain. This environment represents a rich source of a user's context data. It allows extraction of contextual links between the files and building advanced algorithms, using these connections. We show how such information could be structured, collected and utilized for desktop search evaluation. We also consider available information about user's social links and demonstrate an application, which combines desktop search results with personal resources retrieved from Web 2.0 applications.

3.1 Personal Information Management

In the modern world, millions of people use the Internet for everyday tasks. But besides the Web activity, a large part of everyday tasks for knowledge workers is accomplished using local documents stored on a desktop. As the number of such documents grew dramatically in last years, it became challenging to keep track of personal folders and files. A routine search for a document of interest within piles of information items on a harddrive is a tedious task.

Thanks to recent progress in desktop search technology, any knowledge worker or regular user can easily install a personal search engine on top of her local documents. These search tools are effective, yet, most of them are not tailored to the current user. The desktop search applications do not take into account the user context or social network and the quality of desktop search effectiveness is still inferior compared to other search environments. We are interested to see how we could improve current desktop search experience by incorporating contextual and social links into desktop search.

Current limited performance of desktop search tools can be explained by the high level of privacy of personal data, so that research in this area is hindered by absense of public datasets. While evaluation methodologies for retrieval on the Web and in digital libraries are well-developed, the experiments with the advanced desktop tools are hardly repeatable and difficult to compare. As privacy concerns do not allow to copy and distribute personal data, we investigate this problem and attempt to create a desktop dataset for research purposes. In this chapter we present our method for constructing a dataset from desktop data and a desktop-activity logging framework. A distinctive feature of our approach is *activity metadata* collected from user, which can be exploited for improved desktop search.

But the desktop documents and files are no longer stored on a single computer. With the availability of Web 2.0 platforms such as Google Docs[1], Flickr[2] and YouTube[3], personal information becomes increasingly distributed and shared across various online applications. Therefore, it is important to provide a quick glance at the available personal resources and facilitate their search and selective sharing. To address this problem, we developed the deskWeb 2.0 application, which combines desktop search results with resources found in user's social network. The effectiveness of such integration for different types of desktop search is supported by results of a small-scale user study.

The rest of the chapter is organized as follows. Section 3.2 reviews related work on context detection, ranking and desktop search evaluation. Section 3.3 discusses the design for the activity-based desktop dataset. Section 3.4 presents our methodology for collecting the personal data. In Section 3.5 we analyze the collected activity-based dataset and outline research work based on this results. Later, in Section 3.6 we present a novel application, deskWeb 2.0, for social desktop search and a corresponding user study. Finally, we discuss obtained results in Section 3.7.

3.2 Relevant Background

The set of problems related to desktop search is part of the Personal Information Management field, which is developed in information retrieval, human-computer interaction, machine learning and semantic Web communities. Below we cover relevant work in context detection, context-based ranking, evaluation methodologies and combination of desktop search with social networks.

3.2.1 Context in Desktop Search

The important feature of desktop search is that the desktop is a rich source of contextual information. Such context can be represented by location, activity, time, season, emotional state, etc. Modern machine learning methods allow to recognize complex user activities and classify them into high-level tasks [Rattenbury and Canny, 2007].

[1] www.google.com
[2] www.flickr.com
[3] www.youtube.com

They use signals coming from the user's interaction with local files and switches between application windows and web page visits. Being informed about the current user activities, a search engine is able to predict which desktop ranking algorithm is the most appropriate for the user at the query time [Chirita and Nejdl, 2006].

PIM systems can be centered around the creation time of a document, as done in the Lifestreams system [Freeman and Gelernter, 1996], based on tasks - example systems include UMEA [Kaptelinin, 2003] and TaskTracer [Dragunov *et al.*, 2005], - or they can be specialized in search functionality - similar to Stuff I've Seen [Dumais *et al.*, 2003b] and Phlat [Cutrell *et al.*, 2006b]. As PIM systems gradually improve in both functionality and usability, the next step would be to enrich them with activity-based search capabilities. To cite [Cutrell *et al.*, 2006a]: "They will go beyond helping us to find Stuff I've Seen and toward identifying Stuff I Should See." For such an adaptive PIM system one would need to collect context data using implicit feedback. This task requires implementation of the integrated monitoring tools, which take into account possibly sensitive data. Next, the task-related features are extracted from the gathered data and and used by activity-detection algorithm.

3.2.2 Activity Detection

Activity detection is a part of the user modeling process. If the assumptions on the user task are wrong, search improvements are likely to fail. Therefore, an appropriate taxonomy of activities has to be selected and the user tasks should be estimated based on the gathered data. One of the first attempts to predict user needs based on past actions was undertaken in the Lumiere project [Horvitz *et al.*, 1998], a prototype for Office Assistant in Microsoft Office 97. System events like menu accesses were combined into high-level user actions and transformed to observations in a Bayesian model. Recent papers estimate user activity with supervised and unsupervised learning algorithms like PLSI [Oliver *et al.*, 2006] and SVM [Shen *et al.*, 2006]. For learning they use features like like window title, filepath, email metadata, snippets from the document in focus, time intervals between window switches and number of hops between two files in an access chain.

One of the interesting systems is TaskPredictor, an enhancement of the TaskTracer system [Shen *et al.*, 2006], which adds activity detection functionalities. The Naive Bayes and Support Vector Machines (SVM) classifiers were applied to features like window title, filepath, and email metadata. The usage of the semi-supervised EM algorithm to fight the noisy data was later proposed in [Shen and Dietterich, 2007]. The SWISH system [Oliver *et al.*, 2006] is closely related to TaskPredictor, as it also focuses on the problem of automatic task identification. The important difference is that it is based on Probabilistic Latent Semantic Indexing (PLSI), can handle unsupervised learning and requires no manual input from user. A closely related research direction is pro-active retrieval agents or Just-In-Time information retrieval. For example the Watson [Budzik and Hammond, 2000] and Jimminy [Rhodes, 2000] systems suggest relevant information snippets when the user writes a document or an email.

Another approach to recognize the context based on user access-dependent distance between documents is described in [Chirita *et al.*, 2006b]. Clustering of documents accessed within the same time interval looks promising, but experimental evaluation is missing. Interesting work on automatic context detection is done in the CAAD system [Rattenbury and Canny, 2007], which uses a custom data mining algorithm for task detection. To summarize the above, we observe that activity detection is a machine learning problem which can be tackled using SVM, Naive Bayes, PLSI or custom classification algorithms. Based on the detected activities it is possible to improve desktop search ranking.

3.2.3 Desktop Search Ranking

There are several attempts to make desktop search activity-aware, with ranking methods mainly based on linkage-analysis algorithms, such as HITS and PageRank. One of these systems is Connections [Soules and Ganger, 2005], in which file traces are used to create a relationship graph. Several linkage-analysis-based ranking algorithms are applied and the final result includes both content-based and access-based results. Research similar to Connections was presented in [Chirita and Nejdl, 2006], in which authors experimented with different heuristics for the construction of relationship graphs. All initial studies are promising, but performance improvement is modest at the moment. The successor of Connections, Confluence [Gyllstrom *et al.*, 2007], combines applications in focus with original file traces. The most recent work from this research group is the SeeTrieve system [Gyllstrom and Soules, 2008]. The system associates documents with activity-based text snippets and use them at retrieval time. The algorithm is evaluated against the state-of-art Google Desktop [GD, 2011] search tool and demonstrated significantly better recall.

In a recent paper [Kim and Croft, 2010], the authors quantify the impact of type prediction in producing a merged ranking for desktop search and introduce a new prediction method that exploits type-specific metadata. They also try to predict a needed filetype by using discriminative learning models. For readers interested in comparison of commercial desktop search systems we recommend a survey of desktop search tools available from major market players[Lu *et al.*, 2007].

3.2.4 Desktop Search Evaluation

Experimental evaluation is a long-standing challenge in desktop search. A number of recent research papers used desktop data and/or activity logs for experimental evaluation. For example, in [Teevan *et al.*, 2005a], authors used indexed desktop resources (i.e., files, etc.) from 15 Microsoft employees of various professions with about 80 queries selected from their previous searches. In [Qiu and Cho, 2006a] Google search sessions of 10 computer science researchers have been logged for 6 months to gather a set of realistic search queries. Similarly, several papers from Yahoo [Kraft *et al.*, 2006a], Microsoft [Agichtein *et al.*, 2006a] and Google [Yang and Jeh, 2006a] presented approaches to mining their search engine logs for personalization. In other papers [Chirita *et al.*, 2006a] [Chirita *et al.*, 2006c] the temporary experimental settings were used, which made these experiments neither repeatable nor comparable.

We aim to provide a common desktop-specific dataset to the research community. A test collection that contains user activity logs, like the one we propose in this thesis, can help in pushing forward evaluation of desktop search systems.

The traditional Cranfield evaluation methodology, as adopted by TREC, cannot be directly applied to desktop search. It is highly complicated by privacy concerns for personal data, idiosyncratic user behavior, different levels of computer-literacy, variety of information tasks, and non-repeatability of the experiments. An interesting overview of problems related to an evaluation methodology for PIM can be found in [Kelly, 2006]. A method descrived in [Elsweiler and Ruthven, 2007] suggests a task-based search for PIM evaluation. It combines real user search tasks with a set of artificially constructed search tasks - the latter are inferred from user interviews and represent common real search tasks within organization. An evaluation dataset that needs to face privacy issues is the one provided by the MIREX initiative: a standardized dataset and evaluation framework to evaluate Music Information Retrieval systems and algorithms. The MIREX data sets cannot be redistributed due to copyright restrictions and then the organizers provide a service which allows "remote execution of black-box algorithms submitted by participants, and provides participants with real-time progress reports, debugging information, and evaluation results" [Jones *et al.*, 2007].

Another related dataset creation effort is the TREC-2005/2010 Enterprise Track [4]. Enterprise search considers a user who searches the data of an organization in order to complete some task. The most relevant analogy between the enterprise search and desktop search is the variety of items of which the collection is composed. For example, in the TREC-2006 Enterprise Track collection e-mails, cvs logs, Web pages, wiki pages, and personal home pages are available. But this dataset does not contain *personal documents* and *activity logs*, e.g. resource read and write time stamps, window switches, file renaming, etc., which are needed for desktop search evaluation.

A different approach to evaluate PIM systems is the one adopted in the NEPOMUK project[5] where user scenarios, designed observing activities of real users, are the base for the creation of artificial data which are used for the evaluation of the PIM tools developed within the project. But artificial data might also lead to misleading evaluation, as real-world data is usually much more diverse.

One of the most recent efforts in this area is described in [Kim and Croft, 2009]

[4] http://www.ins.cwi.nl/projects/trec-ent/

[5] http://nepomuk.semanticdesktop.org

with a proposal of automatic construction of pseudo-desktop collections, consisting of document gathering and query generation. This approach looks promising, while it still misses metadata such as the folder hierarchy, file creation date and usage logs. It is also an open problem to understand if the "queries in a test collection form an unbiased sample of a real search workload" [Rowlands *et al.*, 2007].

3.2.5 Desktop and Personalized Social Search

Personalizing the search process by considering the searcher's personal attributes and preferences while evaluating a query, is a great challenge that has been extensively studied in the information retrieval community. It is of great interest since user queries are in general very short and provide an incomplete specification of individual users' information needs.

Several works consider personalization using desktop data and external resources. For example, in [Teevan *et al.*, 2005b] the authors index desktop information and experiment with different representations of users, documents and queries for personalized web search. Chirita et al. [Chirita *et al.*, 2007b] explore personalized query expansion based on users' desktop information. Several approaches for personalized Web search are based on global interests using the *Open Directory Project (ODP)* categories [Liu *et al.*, 2004; Chirita *et al.*, 2005; Xu *et al.*, 2008]. In [Liu *et al.*, 2004] the authors map previously visited pages to *ODP* categories and use this mapping to build a user profile. Another work [Chirita *et al.*, 2005] proposes a personalized version of *PageRank*, in which a hand-picked set of preferred users' categories are applied for result re-ranking. Profiles created implicitly from user's bookmarks can be also used effectively for personalized search [Kim and Chan, 2005].

Recently, new approaches for adaptive personalization focus on the user task and the current activity context. There are several approaches trying to predict applicability of personalization while considering the current context of the user's task on query submission [Teevan *et al.*, 2008; Dou *et al.*, 2007; Luxenburger *et al.*, 2008], using either user context or click entropy. A study from [Broder, 2002] shows that the quality of search results can be harmed when personalizing unambiguous or navigational queries. In [White and Drucker, 2007], a study of over 2000 participants suggested that users can be also divided into navigators and explorers. Both categories of users might require different personalization of results, including different result presentation and user interface. An automatic identification of a user goal in web search is described in [Lee *et al.*, 2005], where goals include informational and navigational queries [Broder, 2002; Rose and Levinson, 2004].

So far, social search has not been fully addressed in conjunction with the desktop search problem. While modern desktop search applications allow to mix search results from the web and desktop (Google Desktop [GD, 2011]) or index information on network drives (Windows Search [WS, 2011], Autonomy IDOL Enterprise Desktop Search [AUT, 2010]), they do not search over the user's Web 2.0 data, as until now a major part of users' data was stored locally. Therefore, previous research in the area of "semantic desktop" is focused on extracting locally available metadata and storing it into a single RDF-based repository like in Haystack [Quan *et al.*, 2003] and Gnowsis [Sauermann, 2003] systems. A recent approach implemented in the Beagle++ system [Minack *et al.*, 2010] uses both desktop-located resources and external data proactively fetched from the Web.

Social search systems allow searching for resources of different types, such as URLs, people, tags and their connections and offer ranking algorithms which take into account the structure of a social network. Hotho et al. in [Hotho *et al.*, 2006b] developed ranking algorithms such as adapted PageRank and FolkRank which take network structure into account. Later, Bao et al. [Bao *et al.*, 2007] presented alternative algorithms called SocialSimRank and SocialPageRank. Personalized ranking using social factors was considered by [Bender *et al.*, 2008; Xu *et al.*, 2008]. These systems provide a personalized ranking, based on user's social network, which gives better search results.

3.3 The Design of a Desktop Search Dataset

The volume of data stored on a single hard drive and the amount of interactions with files, emails, skype and other instant messaging applications greatly increased in the last years. Many desktop search tools and systems for PIM were released recently by main search engine vendors. The variety of PIM projects calls for evaluation and comparison of proposed algorithms. As the functionality of many PIM systems stems from area of information retrieval, one can consider existing sound experimental methodologies, e.g. Cranfield methodology [Cleverdon, 1997] or a method for evaluation of interactive systems [Borland and Ingwersen, 1997]. The mainstream evaluation methodologies require an appropriate common test collection that is accepted by the community [Voorhees, 2001]. However, it is difficult to create a dataset with real personal information and testing algorithms on Web-based datasets can be misleading. Without a reliable dataset, it is difficult to make a choice between any ranking algorithms, and results from different research groups become non-repeatable and incomparable.

Currently, existing datasets came either from traditional digital libraries or from Web data. But they miss a lot of usual desktop data, like different file types, folder structure, personal emails, instant messages, etc. The desktop files are also different from Web pages, since they usually do not contain explicit hyperlinks between documents. On the other hand, a lot of work in PIM is related to personalization and the collections from digital libraries cannot provide personalized user profiles.

Desktop data tend to have a semi-structured representation, partially thanks to metadata annotation capabilities developed in state-of-the-art PIM systems like

Aduna Aperture [6] or Beagle++ [7]. For example, the address book contains different metadata fields for personal contacts, while email messages can be searched by date, sender or title. This information should be present in the dataset too. Morover, the user's information need on a desktop has a different focus than that on the Web. For example, people often seek for a previously known item on a desktop, which makes the historical data rather important. These desktop-specific features do not allow re-using existing datasets for PIM evaluation.

Highly personalized systems are designed using information about the current desktop content, but also take into account the current user's activities. It is very likely that users will highly benefit of "a system having knowledge of their specific tasks" [Catarci *et al.*, 2006]. A standard evaluation setup must incorporate and provide activity logs as well as data and metadata of the desktop items. As many desktop resources are accessed within some given activity context, one must be able to reconstruct these contexts in order to exploit them for information retrieval tasks, for example, using metadata annotations, file access timestamps, recently or frequently accessed folders, information about co-active items, etc. For this reason we need to include in a desktop evaluation collection history files (logs) of the activities performed by the desktop user. A dataset satisfying these requirements will allow all the desktop systems that make use of such information to be consistently evaluated and compared against each other.

The high privacy level of user files makes it challenging to create a customized dataset. While some people are willing to share information with their close friends and colleagues, they do not want to disclose it to outsiders. In this case, there is a way to keep information available only for a small number of people within a single research group. To collect the dataset one could use an indexing and logging application, which would collect static information from files and implicit data regarding user activities on a desktop. It should monitor user actions and record a variety of metadata in text logs.

In this section we present an approach for generating an activity-based desktop dataset. It is based on the Logging Framework application for collecting the desktop dataset, which includes activity logs and history of actions with each file or email. The design of our dataset provides a basis for evaluating special-purpose retrieval algorithms for different desktop search tasks. We present a dataset structure and ways for collecting the personal information. We describe a private test collection made of desktop data of about 20 users and outline recent research work carried out by our colleagues using our Logging Framework.

[6] http://www.aduna-software.com/technologies/aperture/
[7] http://beagle2.kbs.uni-hannover.de/

3.3.1 Desktop Dataset Requirements

Type of Information to Store

We propose to collect the desktop dataset data within a single research group. We consider a set popular file formats like TXT, HTML, PDF, DOC, XLS, PPT, MP3 (tags only), JPG, GIF, and BMP. Each group locally collects several desktop dumps, making use of logging tools for a number of applications like Acrobat Reader, MS Office family products, Internet Explorer, Mozilla Firefox and Thunderbird. We distinguish between *permanent information* and a *timeline information*. Permanent metadata could be extracted from files' indexable metadata fields during the one-pass indexing, and timeline metadata corresponds to particular user actions, which have to be continuously logged. The desired permanent and timeline information is listed in Table 3.1.

Permanent Metadata Information (indexing)	**Applied to**
URL	stored HTML files
Song metadata tags	MP3
Saved picture's URL and saving time*	Graphic files
Path annotation	All Files
Scientific publications	PDF Files
Bibliography data	BibTeX Files
Web cache	Web History
Emails and attachments	emails
Timeline information (logging)	
Time of being in focus	All applications
Time of being opened	All applications and files
Path of the file being edited	MS Office files and PDF
Being printed	Thunderbird, Firefox
Text selections from the clipboard	Text pieces within a file
Time of conversation (Instant Messengers)	Skype, MSN Messenger
Browsers actions: bookmark, clicked link, typed URL	Web Pages
Bookmarking actions (creations, modifications, deletions)	Firefox
Google Web search queries	Firefox, IE
IP address	User's desktop
Metadata of emails being in focus	Thunderbird, Outlook
Adding/editing an entry in calendar and tasks	Outlook

Table 3.1 Permanent and timeline information provided by indexing and logging operations.

In our setup, we considered desktop data of professional researchers, therefore, our dataset includes specific metadata like scientific publications or bibliography data. This information was considered useful in several studies and was required for experiments of our colleagues. The rest of information covers most frequent user activities and should be sufficient for majority of information needs. The list of information items could be extended in future, once new file formats and user interactions are introduced in modern desktop systems. The logging tools required for data collection are presented later in this section.

The proposed content of a desktop dataset could be collected within each research group planning desktop search evaluation. It would not be publicly distributed, so a good level of privacy would be preserved. As desktop collections gathered in different places would be similarly structured, the evaluation results - although they stem from different real data - are comparable and, as we define it, "*soft-repeatable*". Even when semantic information (e.g., RDF annotations, activities, etc.) is integrated as part of a search system, the traditional measures from information retrieval theory can and should still be applied when evaluating system performance. This allows the use of the same set of metrics in the evaluation of desktop systems, to make the results comparable among different systems.

Information Processing Tasks

One of the current issues is a consensus in the community on what set of tasks to be evaluated. Among possible information retrieval tasks we envision Ad Hoc retrieval, Folder Retrieval (i.e., ranking personal folders), and Known-Item Retrieval. The discussion is also open for Context Related Items Retrieval, both using example items or keyword queries, Information Filtering, Email Management and related tasks. It is also interesting what kind of advanced search criteria users need. As a starting point, we show some examples of simple search tasks.

Ad Hoc Retrieval Task. Ad hoc search is the classic type of text retrieval when the user believes that relevant information exists somewhere. Several documents can contain pieces of necessary data, but the user might not remember whether or where it has been stored, and might not be not sure which keywords are best to find them.

Known-Item Retrieval Task. Targeted or known-item search task is the most common for the desktop environment. Here the user wants to find a specific document on the desktop, but does not know where it is stored or what is its exact title. This document can be an email or a working paper. The task considers that the user has some knowledge about the context in which the document has been used before. Possible additional query fields are time period, location, and a topical description of the task in which scope the document had been used.

Folder Retrieval Task. Many users have their personal items topically organized in folders. At some point, they may search not for a specific document, but for a group of documents in order to use it later as a whole - browse them manually, reorganize or send to a colleague. The retrieval system should be able to estimate the relevance of folders and sub-folders using simple keyword queries.

Queries

As we aim at real world tasks and data, we want to reuse real queries from desktop users. As every desktop is a unique set of information, its user might be directly involved in both query development and relevance assessment. Therefore, desktop contributors should be ready to give queries selected from their everyday tasks. Their participation in relevance assessment solves the problem of subjective query evaluation, since users know best their information needs.

In this setting each query is designed for the collection of a single user. However, some more general scenarios can be designed as well, such as finding relevant documents in every considered desktop. One could envisage the test collection as partitioned in sub-collections that represent single desktops with their own queries and relevance assessments. This solution would be closely related to the MrX collection used in the TREC SPAM Track [8], which is formed by a set of emails of an unknown person. Following TREC collections traditions, a query can have the format presented in Table3.2.

Query field	Contents
<id>	KIS01
<query>	Pharos project deliverable June 2009
<metadataquery>	date:01.06.2009 topic:Pharos project type: deliverable
<taskdescription>	I am combining a new deliverable for the Pharos project.
<narrative>	I am looking for the Pharos project deliverable, I remember that the main contribution has been done in June 2009.

Table 3.2 Query format.

Compared to the standard TREC format, we included the <metadataquery> field, to enable the specification of semi-structured parameters like metadata field names, in order to narrow down the query. Since the desktop contributors must be able to assess pooled documents few months after the date of collecting the dataset, each query is supplemented with the task description and narrative. A user context also should be stored within the dataset. For example clicked and opened documents in the respective query session, to allow users to provide relevance judgments according to the actual context of the query. If users know their documents, the assessment phase should go faster than usual TREC assessments. For the task of known-item search, the assessments are quite easy, since only one document (or few copies) is considered relevant.

3.3.2 Logging Framework

We collect user-activity data using implicit feedback. This approach was exploited in [Claypool *et al.*, 2001] and proved to be a representative indication of user interests. We acquire activity data automatically by using logging software, which does not require explicit user input. User interaction with a desktop is being monitored without interrupting her workflow. The lack of direct user input is compensated by the amount and granularity of the automatically acquired data. The main advantage of this approach is that the context of accessing the resource or application is being logged. This information could be used to extract missing links between desktop objects.

[8] http://plg.uwaterloo.ca/~gvcormac/spam/

User actions are articulated through the interaction with different applications. In Windows operating system this interaction is expressed by handling windows, which are the visual representation of an application. For example, the window that is currently in focus, is the window that the user is currently looking at and working with. By observing user's actions on windows, we examine the actual activity that the user is performing on the desktop. However, one window can act on several resources. For example, all emails in one instance of a rich email client or several Web pages viewed in an Internet browser. In these cases, we extend the logging activity to monitor interaction with these resources.

First, we developed a logging framework for Microsoft Windows XP [Chernov *et al.*, 2008] and later adapted it to Windows Vista. We would like to thank Michal Kopycki for his contribution to implementation of the Logging Framework, the specific programming details could be found in his bachelor's thesis [Kopycki, 2008]. The general architecture of the logging application is shown on Fig.3.1. The main logging module is a system-wide logger, capable of observing *general user actions* like file opening and closing, application switches, accessed urls and active window title. However, a more detailed information can be extracted only on the application level. Therefore, we developed a separate plugin for *Microsoft Office* 2003 and 2007, which logs every access to email message, including properties like title, sender and sent time. The popular applications *Mozilla Thunderbird* and *Firefox* are logged using a modified version of the Dragontalk plugin [DT, 2007]. Each logging component maintains a separate log file, which is stored locally on a user PC. Moreover, we also log queries to the Google search engine as well as result pages and information about clicked urls. The logging framework is easily extensible.

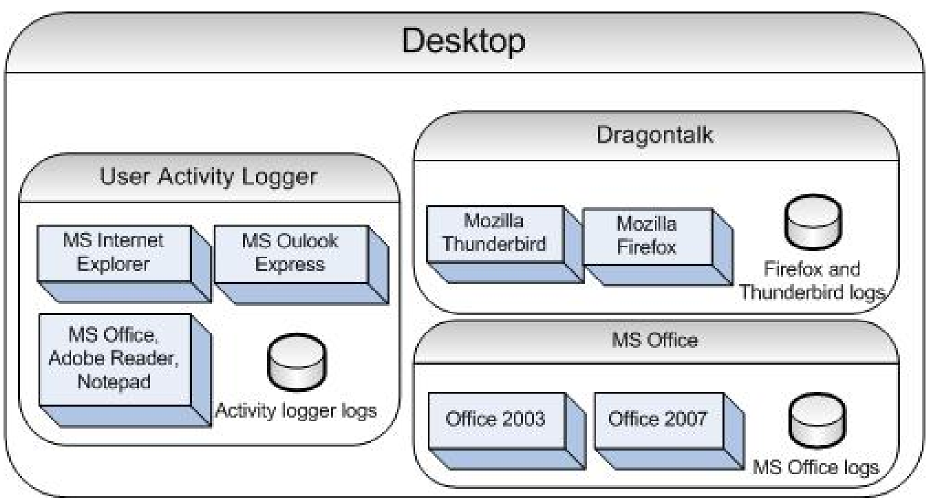


Figure 3.1 Components of the Activity Logging Framework

A special application is used for merging information from three logs into a single database. Before uploading the data to the database, all information is encrypted and only the log owner can decrypt it using a personal key. Using this approach, we can preserve user privacy, while still being able to perform statistical analysis on the activity data. When needed, we can ask a participant to decrypt a particular filename or url, if the user decides she is willing to do so. Users can also specify a list of urls that they do not want to log, disable query history function and easily stop and re-start the logging anytime. All these options are implemented in a user-friendly manner using a taskbar icon and a set of menus. The logging framework was installed on laptops of 21 users and has been running for several months. More details on this set of tools can be found in [Chernov *et al.*, 2008] and [Kopycki, 2008].

The in-depth information regarding user activities is gathered by extensions to the applications that we want to log. Such an extension, which is part of the application itself, has direct access to resources involved in user activities. The description of the resource enriches the description of an activity - and the other way round: the resource description is enriched by the actions that the user is performing on it. For example, the User Activity Logger receives a notification that Outlook 2003 is currently being used and the Outlook 2003 plug-in retrieves detailed information about emails being currently processed by the user. Another example: the Firefox plug-in indicates that since 5 minutes the user was looking at a particular Web page; however, based on data from the User Activity Logger, we know that the system is actually in idle time. This architecture is highly extensible. One can download our framework and write a customized plug-in to explore the user activity of interest.

3.3.3 Logging General User Activity

Our main contribution to logging utilities is the *User Activity Logger*. Once installed, it uses *Windows Hooks* to intercept every "activate", "create" and "destroy" window notification. The pop-up windows, invisible windows and dialog boxes are considered irrelevant and filtered out. For each notification, a generic activity description is being extracted. For some of the applications, the Logger acquires additional information that describes the resource displayed in the window. For example, for Word text editor or Adobe Acrobat Reader, the file path of the currently viewed file is stored; for Internet Explorer, the URL of the Web page currently viewed; for Outlook Express, the currently selected email message. Table 3.3 describes the information being logged by the User Activity Logger.

As the User Activity Logger covers the whole desktop, it is directly bounded to the system architecture. As an implication, it is not portable between operating systems. Currently, the Windows version of the logger prototype is available for download at the Personal Activity Track Web page [9]. To this end, we opened the development to those willing to participate via a *SourceForge* project [10]. Addressing larger groups of users requires porting the User Activity Logger to other platforms like Linux distributions.

[9] http://pas.kbs.uni-hannover.de/
[10] http://sourceforge.net/projects/activity-logger/

Generic information	**Applied to**
Operation type (created, activated, destroyed)	All applications
Timestamp	All applications
Unique window handle	All applications
Application exe name	All applications
Window caption	All applications
Resource specific Information	
File path to resource being viewed	MS Office products, Adobe Acrobat Reader, Notepad
URL	Internet Explorer
Sender, recipients, received date, sent date	Outlook Express

Table 3.3 Generic metadata collected by the logger

3.3.4 Logging Specific Email Applications

Collecting detailed resource information from the User Activity Logger level is possible for a limited number of applications. For other relevant applications, we developed or adapted existing plug-ins. The plug-ins store resource and activity information every time a notification has been triggered by the user. We have implemented such plug-ins for Outlook 2003 and Outlook 2007. By using Visual Studio Tools for Office technology [11], which allows to write extensions for MS Office Family products, we were able to collect in-depth email usage data. Data collected by Outlook plug-ins is described in Table 3.4.

Data description	**Applied to**
Operation type	Outlook, Thunderbird
Timestamp	Outlook, Thunderbird
Unique email ID	Outlook, Thunderbird
Path to the email in the email folder hierarchy	Outlook, Thunderbird
Subject	Outlook, Thunderbird
Sender (name and email adress)	Outlook, Thunderbird
Recipients (name and email adress)	Outlook, Thunderbird
Cc recipients (name and email adress)	Thunderbird
Bcc recipients (name and email adress)	Thunderbird
Address book entry	Thunderbird

Table 3.4 Email metadata fields collected by the Outlook 2003, 2007 and Thunderbird plug-ins

For applications from the Mozilla family, we have used an already existing solution and adapted it to our requirements. Dragontalk project [12] provides extensions to the Thunderbird rich email client and Firefox Internet browser. The extensions allow monitoring of user interaction with both applications. Our adaptation of Dragontalk included changing the outputting method, extending the functionality by supporting new notifications, and adding methods to preserve user privacy. See Table 3.4 for a description of the data collected from Thunderbird.

[11] http://msdn2.microsoft.com/en-us/office/aa905543.aspx
[12] http://dragontalk.opendfki.de/

3.3.5 Information Representation and Storage

Table 3.5 presents the full list of notifications that are currently supported by the framework. For each notification, additional data from Table 3.4 and 3.3 is extracted and stored.

Supported user actions	Supported Applications
General	
Window actions (create, activate, destroy)	All applications
Documents	
Document actions (open, activate, close)	MS Office, Adobe Acrobat Reader, Notepad, etc.
Web	
Navigate to URL (click, type in)	Internet Explorer, Firefox
Tab (create, change, close)	Internet Explorer, Firefox
Bookmark (create, modify, delete)	Firefox
Forward, backward, reload, home	Firefox
Print page	Firefox
Submit Web form	Firefox
Submit Google Web search query	Internet Explorer, Firefox
Email	
Email actions (select, sent)	Outlook 2003, 2007, Outlook Express, Thunderbird
Email actions (receive, reply, forward, delete, move, print)	Thunderbird
Address book entry (create, modify, delete)	Thunderbird
Email Folder (create, modify, delete)	Thunderbird
Instant Messengers	
Conversation (start, activate, finish)	Skype, MSN Messenger
System state	
Idle time (start, end)	System event
Hibernation (start, end)	System event
Framework state	
Logger actions (activate, deactivate)	User Activity Logger

Table 3.5 Types of notification supported by the Logging Framework

Collected data is stored in a simple human-readable format in text files located directly on user's computer. As different parts of the Logging Framework focus on user interaction with different resources, the format and granularity of output data differ as well. For example, a single notification intercepted by the User Activity Logger (e.g. Firefox window activated), may imply several notifications from the Dragontalk Firefox logger (switching between Web pages without leaving the Firefox window). For this reason, we decided to keep a separate log file for each component of the framework. As a result, in the current implementation, the user can have up to four log files (User Activity Logger, Thunderbird, Firefox, and Outlook 2003 and 2007). However, the simplicity of the format allows to parse it to any other format. In the scope of cooperation with the NEPOMUK project [13] we translated our output format into NEPOMUK Ontologies [14] by using a readable RDF syntax, called Notation3 [15]. The problems and limitations that were discovered during experiments

[13] http://nepomuk.semanticdesktop.org
[14] http://www.semanticdesktop.org/ontologies/
[15] http://www.w3.org/DesignIssues/Notation3

with the Logging Framework we discuss later in Section 3.4.3.

3.3.6 Privacy Management

Obviously, each logging utility introduces privacy issues. The collected data is very sensitive and exposes user interaction with the whole desktop. Our main consideration was to protect the data from unauthorized access. Because all the data is stored directly on the user's computer in plain text files in human-readable format, it is up to the user to decide to whom and in what form the data should be released.

In the Logging Framework we preserve the user's privacy by offering means to stop or pause the logging process. The user can pause the process or simply shut down the logging utility via a user-friendly menu (Figure 3.2).

Figure 3.2 Tray icon and menus provide control over the logging process

However, the goal of monitoring the user activity is to collect as much data as possible. Therefore, we introduced other means that only restrict the logging range without terminating the process itself. Figure 3.3 presents two dialog boxes that allow the user to specify which Web domains should be excluded from the logging process. Once specified, the utilities will ignore any notifications involving these resources.

3.4 Creating a Desktop Dataset

Here we describe the approach taken to build a personal information search test collection according to design presented in a previous section. The collection that we created is composed of data gathered from the PCs of 14 different users. The participant pool consists of PhD students, PostDocs and Professors in our research group. The data has been collected from the desktop contents present on the users' PCs in November 2006. For this reason the data and the activity logs collected are mainly referred to that year. Each data provider is allowed to use the entire collection for research experiments. We observed that only a subset of providers are actually experimenters but, in any case, all the providers had to sign a written agreement as they gain the access to the collection.

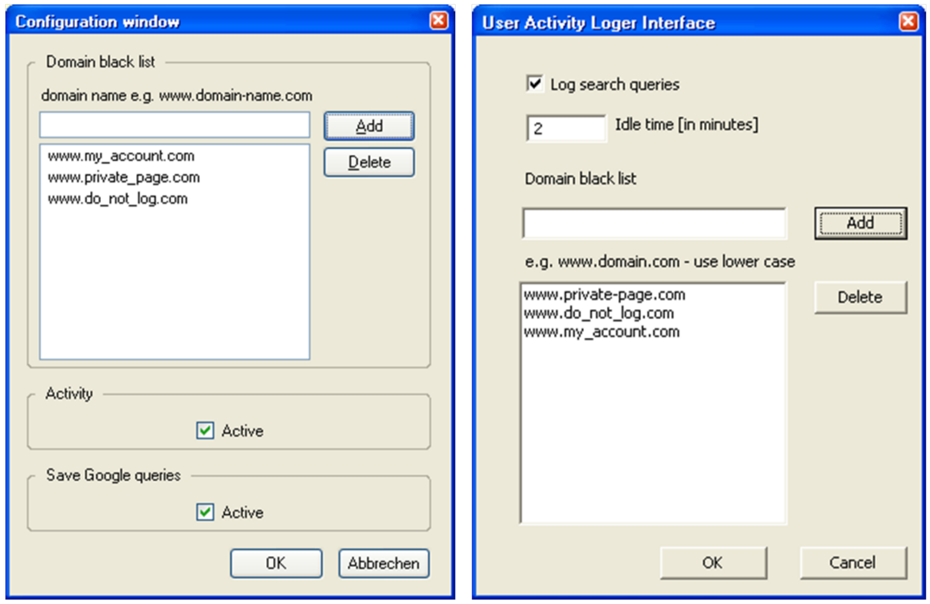


Figure 3.3 Menus restricting the range of the logging process by specifying pages that should be excluded from the logging process. Firefox (left) and Internet Explorer (right)

3.4.1 Collecting Desktop Dataset with Static Medatada

In 2006 we made our first attempt to create a desktop dataset. At this time, we did not have the logging tools ready, so this dataset did not include activity metadata information. The desktop items that we gathered from our 14 colleagues, include emails (sent and received), publications (saved from email attachments, saved from the Web, authored / co-authored), address books and calendar appointments. In order to face the privacy issues related to providing our personal data to other people, a written agreement has been signed by each of the 14 participants. A distribution of the desktop items collected from each user can be seen in table 3.6:

A total number of 48,068 desktop items or around 8Gb is collected as dumps of desktop data, including all kinds of documents, not just emails, publications, address books or calendars. On average we have 3,433 items per user. In order to emulate a standard test collection, all participants provided a set of queries that reflects typical activities they would perform on their desktops.

The query sets are composed as follows. Each user has been asked to provide two clear keyword queries, two ambiguous keyword queries, two metadata-only queries (e.g. “from:smith”) and two metadata and keyword queries (e.g. “information retrieval author:smith”). In total, 88 queries were collected from 11 users. The average query length was 1.77 keywords for the clear queries, 1.27 for the ambiguous queries, and 1.65 for the metadata queries. As expected, the ambiguous queries are shorter than the clear queries, which are in 73% of the case composed of a single term. These

User#	Emails	Publications	Addressbooks	Calendars
1	109	0	1	0
2	12456	0	0	0
3	4532	1054	1	1
4	834	237	0	0
5	3890	261	1	0
6	2013	112	0	0
7	218	28	0	0
8	222	95	1	0
9	0	274	1	1
10	1035	31	1	0
11	1116	157	1	0
12	1767	2799	0	0
13	1168	686	0	0
14	49	452	0	0
Total	29409	6186	7	2
Avg	2101	442	0.5	0.1

Table 3.6 Desktop items distribution over the users.

results are comparable to the average of 1.7 keywords, as reported in other larger scale studies (see for example [Dumais *et al.*, 2003a]).

In order to collect also ground truth data, we asked the data providers to manually assess the relevance of some search results. For every query and every system (we used 3 different ranking algorithms), each participant rated the top 5 output results on a Likert scale (from 0 to 4, with 4 being very relevant for the query and 0 without any connection to the query).

This dataset has been used in local projects like Beagle++[Minack *et al.*, 2010], but it had a limited utility for other experiments as activity metadata was missing. The next version of the dataset has been collected using our Logging Framework.

3.4.2 Collecting Desktop Dataset with Activity Medatada

Participants and Logging Procedure

The second version of the dataset we collected at L3S Research Center, over the period of one year and two months between October 2006 and December 2007. The user behaviour was recorded using our Logging Framework. The software did not require direct user intervention, excluding personal privacy management, in which users could manually specify a list of Web domains removed from logging and start/stop logging where necessary. Otherwise, the logging tools did not modify usual user behavior or application settings. The final metadata from the participants has been gathered in January 2008.

In the beginning we asked 25 people to participate in our experiment. While 4 people worried about privacy issues, we proceeded with 21 users. About 1/3 of our subjects are female, the average age is around 30. All participants are computer scientists with advanced user skills. The experiment has been performed on Windows

XP operating system. During the period of the data gathering the users lived in Germany, but originally they represent 11 different countries. They interacted a lot with their desktop, providing a large the amount the activity data. Obviously, such a dataset is skewed towards information workers rather than regular users.

All participants were instructed to install the Logging Framework on their computers. This process included installing the User Activity Logger and Dragontalk or Outlook plug-ins, depending on the software present on participants computer and her privacy concerns. Given the modular nature of our logging, the data capturing process was divided into four main subtasks, each responsible for covering different range of activities and applications. Although we were encouraging participants to install all the components of the framework, they could decide which module to install and use. Consequently, different activities received different coverage across the participants pool. With desktop activity logged for 21 participants, Web activity was registered for 20 participants and 15 participants covered Email activity. Additionally, 13 participants provided us with Web Search activity data. This discrepancy happend since some users did not install a particular component of the Logging Framework. Also, our logging software did not support some applications, in particular the Opera Web browser.

Each participant was using our logging utilities for some period of time between October 2006 and December 2007. The average number of days with at least one event logged equals 177 with a minimum of 40 and maximum of 406 active days. The average number of inactive days equals on average 47. Inactive days included days when participant did not use her computer at all or decided to disable the framework for any reason. From the whole pool of participants, one was using a stationary PC, 20 were using laptops. Some participants, took their computers home on a regular basis and used it for private purposes.

Data Representation

From the beginning of the study, our participants expressed their unwillingness to provide raw activity data, considering it sensitive. Following their requests, we added encryption capabilities to preserve user privacy and upload anonymized data to a central location for further analysis. Such workaround allows a cross-user analysis and enables changing the parameters of any experiments without asking the participant to re-run evaluation software every time. All information considered sensitive was encrypted using a symmetric-key encryption algorithm AES.

- Encrypted data: bookmark URI, bookmarks title, bookmarks folder, all URLs, anchor text, submitted web form, page title, search query, result snippet, email subject, sender, all recipients, attachment name, address book action, window caption, resource path;

- Plain data: event type, timestamp, search statistics, result click order, email

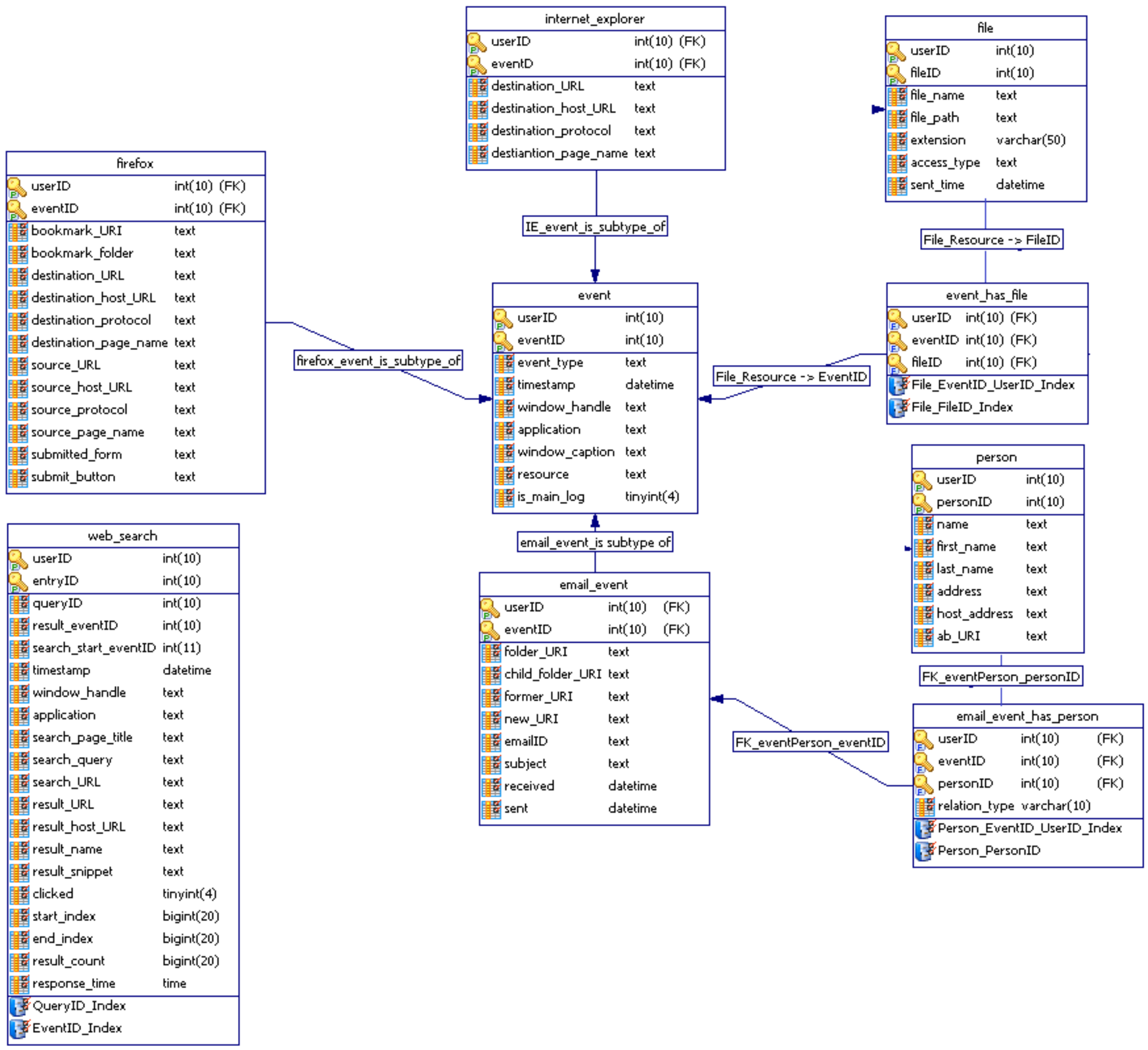


Figure 3.4 Database schema for storing activity logs

ID, received date, sent date, application name, windows handle.

While most of the information had to be encrypted in order to satisfy privacy concerns, we introduced specific encryption schemes to keep data useful for analysis. The goal of each scheme was to maximize the utility of encrypted data despite the fact that it is not human-readable. For example, each URI was split into URI scheme, host and dynamic part, each part was then encrypted separately. The email sender and recipient were split into name and email address, filename split into file name and extension, etc. This representation allows for statistical analysis, without disclosing specific user activity.

In order to proceed with the evaluation process, we uploaded collected user behavior data into relational database. The database schema is shown on Fig.3.4.

We observed that data coming from different applications and captured by different parts of the framework had diverse format and structure. To avoid faulty interpretation, we decided to keep the structure of the data in the database as close as possible the original logs. However, the nature of some data, for example many-to-many relationships between Email messages and sender/participant elements, required a specify structure to suit the relational database model. To satisfy this tradeoff, we restricted modifications of the original structure to conditions where it was necessary in order to meet demands of the database model.

3.4.3 Discovered Data Problems

During the evaluation, we encountered several problems connected with the nature of the data as well as the way our logging utilities acquired it. To prevent false interpretation, these issues should be considered.

Software techniques used for capturing user activity provided a diverse information with different levels of detail. For example, in comparison with Internet Explorer, the Dragontalk Firefox plug-in supported additional events like bookmarking actions, page printing or forward and backward navigation. Additionally, some events from the Internet Explorer log (e.g. page visited) corresponded to several events from the Firefox log. For example "page visited" in Internet Explorer would look like "link clicked" plus "page loaded" in Firefox. The difference in granularity of events has to be considered in any quantitative analysis of the dataset.

Since parts of the Logging Framework were developed independently, the actual meaning of some events differs. For example, the Outlook and Outlook Express "email selected" event was triggered when the content of the message has been presented to the user, which can be interpreted as a user reading the particular message. However, for the Dragontalk Thunderbird plug-in "email selected" event corresponds to selecting any message from the message list. If a user selects just one message to read, this would not cause a problem. But in case the user selects a group of messages to move them to another directory, all of them will be appended to the log. For this reason, not all "email selected" events corresponded to the actual reading activity.

The segmentation of the framework log files also caused some difficulties. Data from different log files had to be merged to represent the actual timeline-based workflow. If two events originating from different log files appeared within the same second, the order of the events can not be clearly indicated.

Because each logging utility can be managed independently from other utilities, activity coverage may vary. For example, if the Firefox plug-in is working, but the system-wide User Activity Logger is shut down, no idle time indication will be present. This would significantly influence the evaluation of information like Web page visit time.

During collection of the data, software instability issues were discovered, which had impact on the quality of the gathered data. For one participant the Google Web page was treated as a blacklist item, although the user did not specify this. Consequently, the users' Web search information has not been recorded, what in the end made part of her data not useful for the extraction of Web search sessions. Outlook automatic updates proved to the cause of a second serious software issue. During some major updates, the Outlook plug-in was automatically disabled by Outlooks TrustCenter component. The plug-in remained disabled, until it was manually switched back on. As there is no easy noticeable indication of the plug-in status, useful data can get lost during the time its disabled.

Issues discussed above, have to be considered while evaluating and interpreting data collected by the framework. Most of the encountered problems were directly related to the segmentation of the framework and this aspect is a crucial issue for future implementations. Creating stable, more standardized and unified utilities is the key for increasing quality of recorded data.

3.5 Desktop Dataset Description and Outcomes

Moving from the technical details of collecting the data, in this section we present results of quantitative analysis of the dataset. Starting with general statistics, we describe its characteristics and most important parts. Finally, we outline few research outcomes obtained by our colleagues using the information logged with the Logging Framework.

3.5.1 Dataset Description

General Statistics

At the end of the logging process we gathered data from 18 out of 21 participants. Data of one participant was lost due to a harddrive crash, 2 participants left L3S Research Center during the study. The logs were were anonymized, parsed, encrypted and uploaded to database. In total 2,828,706 events were logged. During the period of the study, participant spent on average 5 hours per day actively interacting with their desktops. However, this average is biased by days on which participants used their desktops less, like weekend or holidays. For each participant, an average of 157,150 events were logged, which translates to 902 events recorded per day. On average 108,502 "switchings" (558 per day) were observed. Switchings corresponf to user actions like creating, activating or destroying a window and toggling between applications. They also reflect different resources opened in a single application, for example, several pdf documents in a single instance of Adobe Reader. In that sense, users with a lower number of switchings (39% of participants) unveil moderate and concentrated interaction with the desktop. The 61% of participants with a greater number of switching, manifest more differentiated behavior.

General Application Usage

In total we identified 113 different logged applications. This number includes all application that were accessed at least once during the whole study. As expected, among most common applications are email clients, Web browsers, text editors and viewers.

To understand desktop activity coverage in our dataset, we assigned most common applications to seven categories based the common activity type. Figure 3.5 illustrates category distribution over the dataset. Web activity has the strongest representations in the dataset and evidently surpasses remaining categories. Email and text content interactions are uniformly represented. The relatively strong position of instant messengers is an interesting phenomena and should be examined in more detail. The minor media coverage could be explained by the fact that participants were mainly logged during their working hours.

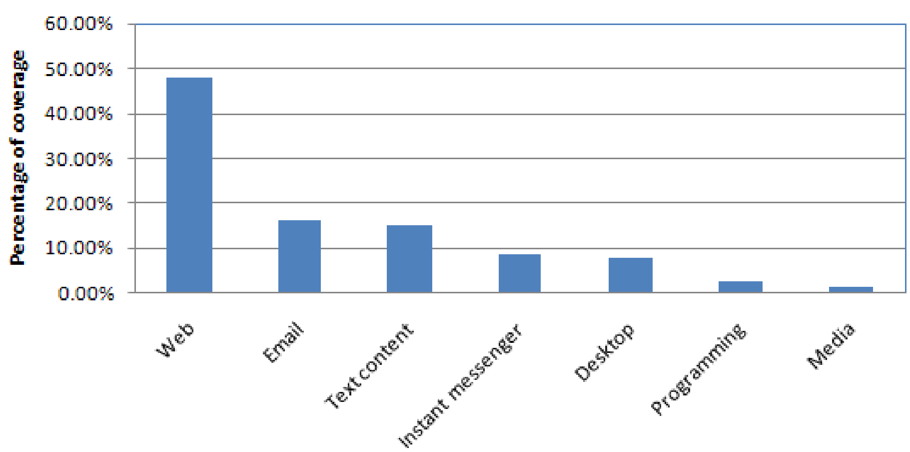


Figure 3.5 Activity types distribution based on access frequency

Web Usage

Web activity occupies a large part of our dataset, making it a valuable source of information on user Web behavior. In total 890,218 Web events were collected which makes an average of 48,028 per participant. The average of 302 Web events per participants active day, differs significantly from the 89 pages reported in [Her06], because of the number of additional Web events supported by the Logging Framework that were not included in the reported study.

During the study, each participant accessed on average 9,337 unique URLs from 1,361 distinct domains. The additional functionality of our Web browser plug-ins gave the possibility to collect Google Web search data, which after preprocessing enables reconstruction of GoogleWeb search sessions issued by the participants. In total 13 participants enabled this feature providing us with total of 17,395 queries. Finally, the Dragontalk plug-in collected additional bookmark and web form events from Firefox users. For 12 participants using this Web browser, a total of 5,251 bookmark events and 25,021 of web form events were stored.

Email Usage

Email activity, the second biggest part of the dataset includes 172,363 events with an average of 137 events per active day. Because the Email events included senders and recipients, 20,616 name-email address combinations were gathered. Therefore, each participant is connected via sender/participant relationship with an average of 1,374 pleople from 515 distinct domains. The dataset includes information on 39,412 distinct emails, on average 2,815 per participant.

File usage

The dataset includes information about 20,809 files with over 500 file formats. The most common file formats are PDF, DOC, TXT, MSG, HTML, XLS, JPG, PPT and GIF. While considering file monitoring, our main goal was to observe user interaction with files containing textual content. The framework was optimized to capture events triggered on those formats, support for other file formats was not intentional and data about them could be misleading.

During the study we have collected data from 13,092 text documents. Figure 3.6 presents the file format distribution based on frequency of access over the whole dataset.

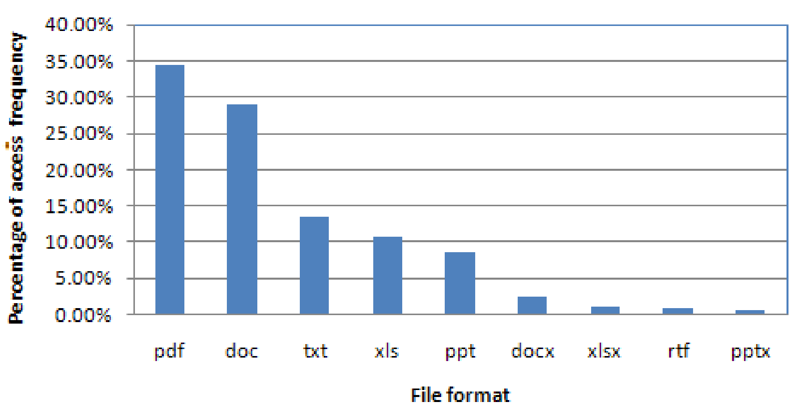


Figure 3.6 File format distribution based on access frequency

3.5.2 User Behaviour

In the scope of general characterization of the desktop dataset, we want to outline several interesting observations on user behavior. These are examples of practical value of the collected data.

First, to prove the quality of the data, we compared Web page visit times, with a study reported in [Herder, 2006]. We found out that our results match positioning the peak of the Web page visit time distribution between two and three seconds see Figure 3.7. It suggests that our dataset is representative with respect to Web surfing behaviour the users.

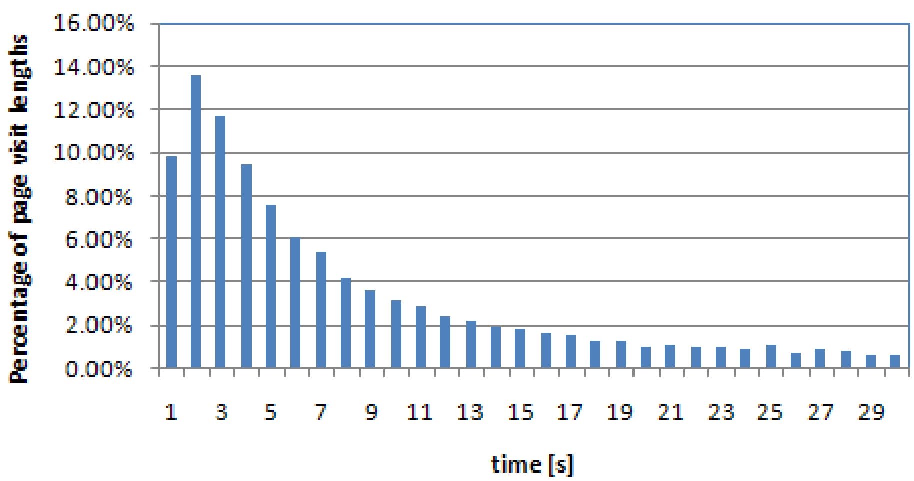


Figure 3.7 Distribution of Web page visit time

While analyzing Email activity of our participants we found out that the reaction time between receiving an email and reading it is very short. Most users could be classified as instant readers, accessing an email within two minutes from its arrival into the inbox. But one user was extremely different from the rest of participants, demonstrating uniformly distributed reaction time. Figure 3.8 presents an extremely instant reader, accessing over 50% of her messages within one minute from their arrival. Next Figure 3.9 illustrates reaction time distribution of an moderate reader. While analizng both cases, we found out that the moderate user was using Outlook Express, which lacks notification information on email arrival.

Another interesting aspect of user are here filing practices. We analyze it by examining the distribution of file access over the folder hierarchy. As mention before we are mostly interested in files representing text content, and this analysis was performed for those file formats. Figure 3.10 illustrates, how file access is distributed over the folder hierarchy.

According to [Alvarado *et al.*, 2003], a *filer* is a person who organizes information using a rigid structure, and a *piler* is someone who maintains a mostly unstructured

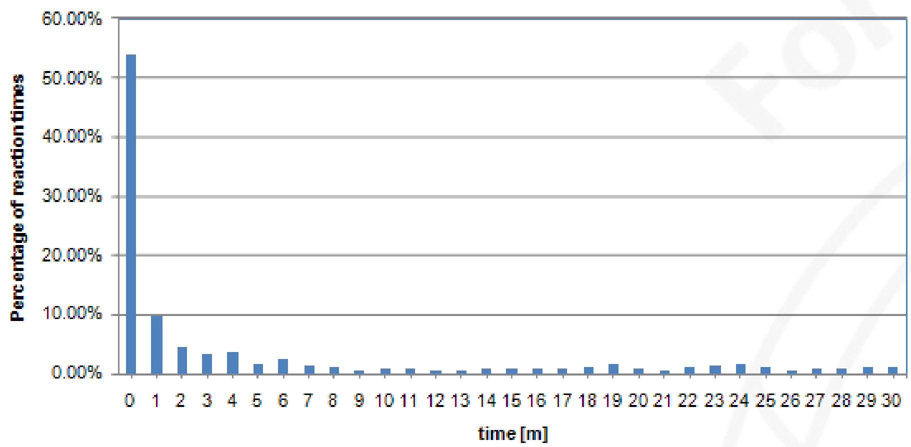


Figure 3.8 Distribution of reaction time to email arrival for instant reader

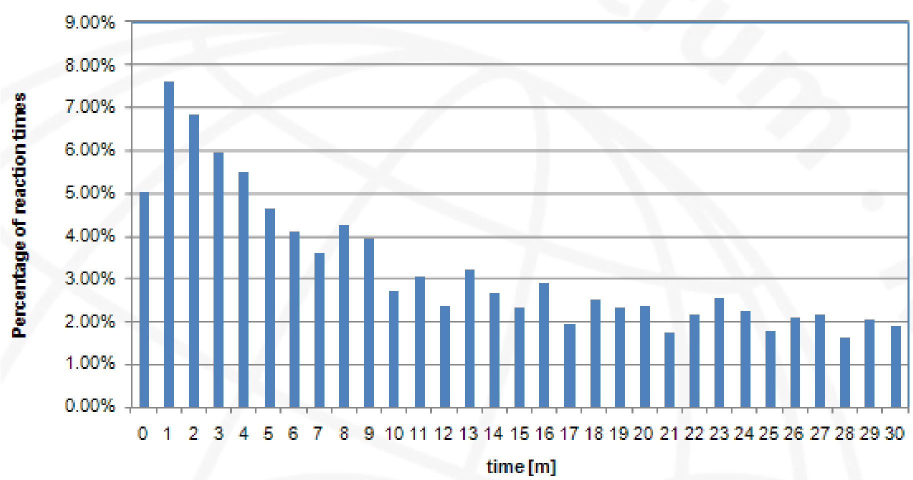


Figure 3.9 Distribution of reaction time to email arrival for moderate reader

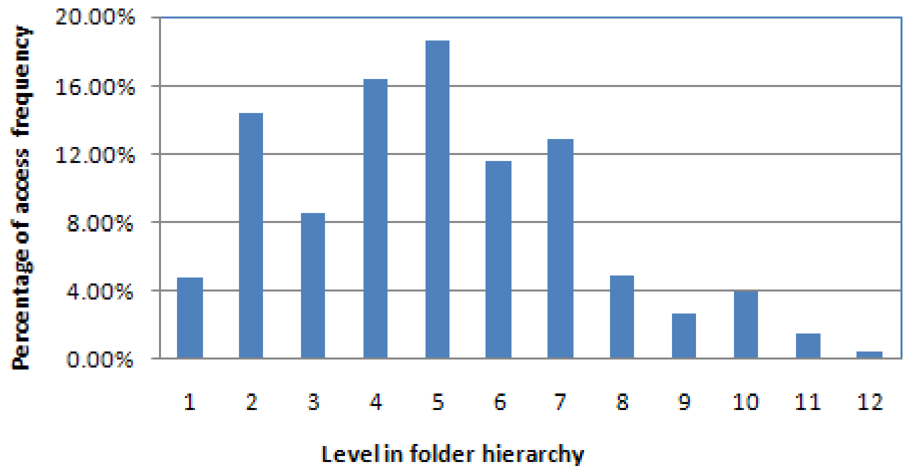


Figure 3.10 Distribution of file access events over the folder hierarchy

information organization. In the current file distribution level 5 corresponds to desktop metaphor and is typical for pilers, saving their documents right on a Windows desktop. Still, we observe filing as more popular practice, as significant portions of files are stored on several different levels of folder hierarchy.

This observation concludes our brief analysis of desktop activity data collected using the Logging Framework. This software provides rich user context information with minimum intervention from the users. With 18 participants this data represents a relatively large sample compared to other desktop studies. The amount of data and its quality make it a good source for experimental analysis of user behavior

However, the complex nature of the data could be examined in more detail. The activity information originates from various desktop activities, includes interaction with different information types and was collected by application providing different granularity of events in individual format.

3.5.3 Research Outcomes based on the Desktop Dataset

In the previous sections we described our approach to logging user interactions with a desktop and proposed possible usage of such data. We would like to stress usefullness of the Logging Framework and the data collected. For this reason, we provide here few examples of external research results achieved by our colleagues using our logging tools.

Using Desktop Dataset for Context Detection

One important research direction in desktop studies is a context detection. People tend to remember of information in terms of associations and context [Teevan *et al.*, 2004]. A task context could be combined from a set of related people, documents, tools and resources. More specifically, a desktop context is a collection of resources that the user uses to solve one task.

The [Costache *et al.*, 2010] used the Logging Framework to collect time-related similarity evidences. Besides traditional text similarity techniques it combines a variety of such evidences to identify the working context. In the proposed method authors focus on identifying file-to-context assignments, based on the content of the files on the desktop, as well as the time connections between them. They construct a Bayesian Network for modeling the evidences and their connections. A Bayesian Network is used to infer which files belong to which contexts. The aquired knowledge about the user's context can be used to support the user with the desktop resources and for a user's profile refinement.

Using Desktop Dataset for Search Ranking

The Logging Framework was meant not only for Context Detection, but also to help in actual desktop search. An approach which utilizes our framework is described in [Gaugaz *et al.*, 2008]. The authors propose to use activity based heuristics in order to rank desktop search results. The semantically related desktop items are connected by exploiting activity information about accesses to local resources. It investigates several approaches to translate this information into a desktop linkage structure, and suggests several algorithms to efficiently rank desktop items. The ranking results based on the user activity links outperformes regular TFxIDF ranking scores and the rankings generated using the schema based approaches.

3.6 Desktop Search Using Social Links

So far we experimented with contextual links extracted from user activity on a desktop. But with the availability of Web 2.0 platforms such as Flickr [16], YouTube [17] and Delicious [18], the personal space of information is no longer locked within a single desktop. Instead, it became increasingly distributed and selectively shared across various online applications.

Web 2.0 applications and social platforms are famous as convenient tools for sharing personal information. For example, a recent report [Heymann *et al.*, 2008] shows that around 115 million bookmarks were available on the Delicious social bookmarking site alone in 2008. In these settings it is important to provide a quick glance not only at the user's files available locally, but also include resources shared online by the user and her friends in search results. Whereas each single Web 2.0 application

[16] www.flickr.com
[17] www.youtube.com
[18] www.delicious.com

is specialized in a set of predefined tasks, users would expect a single search interface over the entire set of distributed personal knowledge resources.

We illustrate this problem with the following example: Alice organizes her trip to the SIGIR 2010 conference in Geneva. She needs to retrieve relevant resources stored on her desktop, such as a sample form to authorize the trip and trip-related email communication. Also, she would like to see hotel recommendations from her colleagues and multimedia resources related to places of interest in Geneva visited by her friends. Finally, she is interested in links and news shared by the other conference participants. This task would require performing search on her desktop and within each relevant social application such as Youtube, Flickr, and Delicious separately. Alternatively, Alice can use some search application to retrieve all these resources at once, see Figure 3.11.

Figure 3.11 Social Desktop Search

But majority of the existing desktop search applications do not support integrated search over shared social resources. A few applications such as Google Desktop Search [GD, 2011] try to combine web and desktop search results, but they do not support sufficient integration of search results obtained from user's accounts on social platforms. This lack of integration requires users to perform search in each social platform separately, which is a tedious task.

3.6.1 deskWeb 2.0 Application for Social Desktop Search

To address the problem above, we developed the deskWeb 2.0 search application. This integrated search environment gives a users a quick glance at the available personal resources both on the desktop and in social networks. The welcome screen of deskWeb 2.0 is shown on Figure 3.12

This application is built on top of Google Desktop Search and uses the currently active Web browser as a basis. It provides search results obtained from Google Desktop Search together with search results collected from various Web 2.0 applications.

Figure 3.12 deskWeb 2.0 welcome screen

Before using our software, a particular set of user accounts on supported platforms has to be specified. On Figure 3.13 we show a simple set of user accounts defined with a locally developed Interweb service.

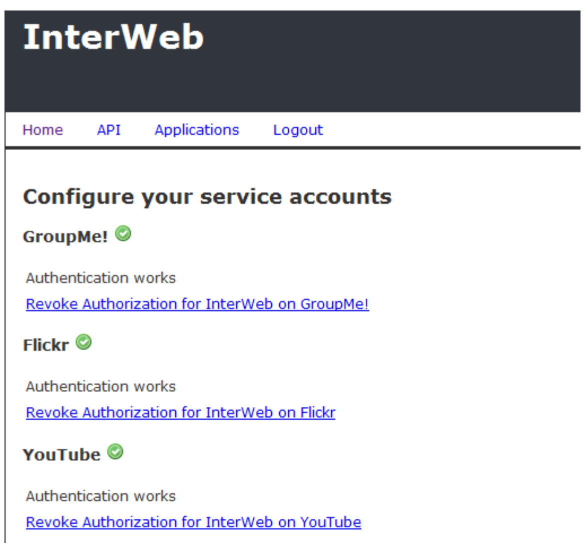


Figure 3.13 A sample set of user accounts specified using Interweb service

The user selects which platforms she uses from the list of social networks supported by Interweb. She provides credentials for access to these networks in order to automaticaly fetch search results later from these platforms. For example, on Figure 3.13 we see that user specified accounts for GroupMe!, Flickr and Youtube social networks. The search over social networks is implemented using publicly available APIs. It issues real-time queries to all platforms of interest and takes into account personal resources published by friends and peers of the current user.

When a large part of personal resources is distributed among heterogeneous social services, it becomes extremely difficult to provide satisfactory desktop search results. In this work we use the notion of *social desktop search* to describe the search process over data gathered from user's desktop and personal networks in Web 2.0 applications, such as social bookmarking systems, blogs, forums, social network sites (SNSs), and others. To integrate desktop and social search within deskWeb 2.0 we need to address two challenges discussed below.

Technical and Semantic Interoperability

Both technical and semantic interoperability is required for authentication, authorization, sharing and search services of the connected platforms. Currently, such functionality is partially supported by the Web 2.0 tools using platform-dependent APIs. Given the large number of available services and possible software updates such integration becomes an essential technical problem and a laborious engineering task. Moreover, as different systems focus on specific shared data types and support different syntax, it becomes important to address these differences at a query transformation step. Finally, resources within social networks frequently change and require efficient update propagation to guarantee up-to-date search results. We address the integration problem on a purely technical level and prepare a query for each service by applying service-specific heuristics.

Ranking and Aggregation of Search Results

Resources from different social platforms differ in their relevance, quality, and relation to the user significantly. Furthermore, users often share sequences of similar resources, such as photo series, such that search results can contain (near-) duplicates or similar resources in different formats. Following the ideas developed in social search [Carmel *et al.*, 2009], the relevance of resources should be influenced by the distance within the personal network, i.e. resources of closely connected peers should be ranked higher compared to resources gathered from friend-of-a-friend (FOAF). Moreover, even if all resources in a sequence have similar relevance to the query, aggregated search results should rather provide an overview over the available options. Our implementation of deskWeb 2.0 takes into account the distance within the personal network to facilitate top-k processing and presents a fixed number of results from each relevant platform.

One important task of deskWeb 2.0 is to provide an overview over the available results. To increase novelty of results and give the user a quick glance of the available resources, we restrict the total number of results obtained from each service as well as the number of results retrieved from a particular user.

Below we present a search algorithm over user's personal social network, supported by a small-scale user study to assess how different desktop search tasks benefit from integration of social search results.

3.6.2 deskWeb 2.0 Search Algorithm

To answer a query, deskWeb 2.0 gathers a user's personal resources as well as resources from the user's social network available through FOAF relationships. We model an integrated user's network as a tree, where each edge represents a friendship relationship and each node represents a user. The root node is the querying user. The children of the root node are the direct friends of the user from each connected platform. To transform a social network graph into a tree we apply an algorithm that selects the shortest paths within the graph. To retrieve up-to-date diverse search results, we implemented a query propagation algorithm presented in Algorithm 1.

Algorithm 1 Pseudocode for deskWeb 2.0 Search

```
GETTOPK(keywords, root, k, min_relevance, results){
    priorityQueue<relevance> queue;
    root.maxresults=root.relevance * k;
    queue.enqueue (root);
    WHILE (true)  {
         node = queue.dequeue();
         IF (node.relevance < min_relevance) break;
         node.results=node.query(keywords, node.maxresults);
         results.add(node.results);
         IF (results.size()>=k) break;
         FOR (friend: node.friends){
              friend.maxresults=node.maxresults-node.countresults;
              queue.enqueue(friend);
         }
    }
}
```

This algorithm traverses the tree in a breadth first manner. As a node can possibly contain either too many or too few results, the goal is to obtain a balanced result set giving the user an overview over the available results. To decide on the number of results to be retrieved from a node n, it weights k in top-k with the relevance of the node n. In case the node does not contain enough results for a query, it propagates the remaining number of results to the n's children.

The relevance of a resource in the integrated social network of deskWeb 2.0 depends not only on the content of the resource, but also on the global importance of its owner within the network and the strength of the relationship between the owner and the querying user. We calculate this relevance heuristically combining social and text-based factors in Equation 3.1:

$$S(r, n, q) = S(n, neighbourhood) * S(n, user) * S(r, q) \quad (3.1)$$

where S(r,n,q) is the similarity of the resource r from node n with respect to query q is the product of three items. First item is $S(n, neighbourhood)$ computed as similarity between the node n and its social network. Second item $S(n, user)$ is the similarity of the node n with respect to the querying user. A last item is $S(r, q)$ or similarity of the content of r to the query q. According to Equation 3.1, each node within user's network can be weighted independently of the query which supports efficient top-k query processing.

3.6.3 User Evaluation

To find out how users search personal resources using currently available tools we carried out a small questionnaire. We asked 22 graduate students from the Computer Sciences department of Leibniz Uinversity of Hannover to tell us which tool they use to find resources on their desktop. The majority of the Windows users (10 out of 12) use the native Windows Desktop Search tool; 3 participants use other tools, such as Spotlight or "find" command for Linux; 9 users do not use desktop search tools. As our current implementation relies on Google Desktop Search to retrieve local search results, we had to limit our user study to few people who installed it.

Research Questions and Evaluation Setup

In our evaluation of deskWeb 2.0 we focused on two research questions:

- **Question 1:** Does search over social services contribute to desktop search with respect to the relevance of results?
- **Question 2:** Which types of desktop search, such as location, people and general information finding, benefit from social search and which do not?

To answer these questions we performed a small user study. Our participants were five students from Computer Sciences department, who had Google Desktop Search tool installed. We selected four tasks for the users to perform, each including three search types. Each task required the user to retrieve certain information from the integrated environment of deskWeb 2.0:

- T1. Collect information for a business trip;
- T2. Prepare a tutorial on a topic of interest;
- T3. Organize a short-distance weekend trip with friends/family;
- T4. Organize a party.

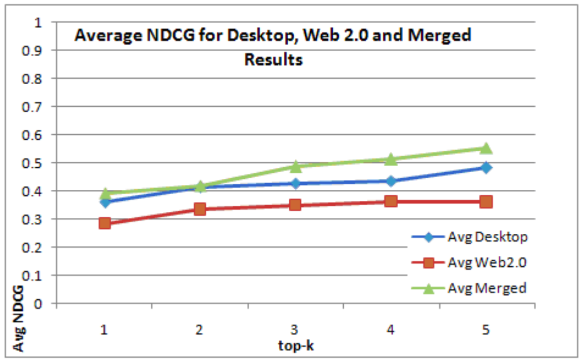


Figure 3.14 Average nDCG of all Tasks

Each task included the following search types:

1. A. Find contact details of a person;
2. B. Find location information;
3. C. Find general information.

For each task and search type, we asked users to issue a keyword query of their choice. For every query, we presented the user with two lists of results, one containing top-5 results from Google Desktop Search, and another list containing up to five results from each Web 2.0 service on which the user had an account. We asked users to rate the results on a 3-point scale as “relevant”, “less relevant”, or “non-relevant”. The users had one or two active accounts on social services supported by our prototype so quantities of desktop and social results were comparable.

Results

To answer Question 1, we computed the macro-averaged Normalized Discounted Cumulative Gain (nDCG) in three result lists: desktop, Web 2.0, and a merged list. To compute NDCG we ranked each list by TF-IDF scores. On Figure 2 we present the nDCG values for top-5 results averaged over the participants. As we can see from Figure 3.14, although absolute nDCG values of Web 2.0 results is lower than the values obtained by the desktop search, combination of desktop and social search results increases the gain of desktop search for all k>2 by about 6% on average.

We also report the results per each task T1 - T4 to see if there are any specific situations in which social search is useful. For readability reasons we split these results into two plots i.e. Figure 3.15 and Figure 3.16 and present only desktop and merged results. From the task-wise presentation we observe that tasks T1 and T3, both related to travelling preparations, only modestly benefit from merging with social search results. In contrast, task T2 about tutorial preparation shows stable and

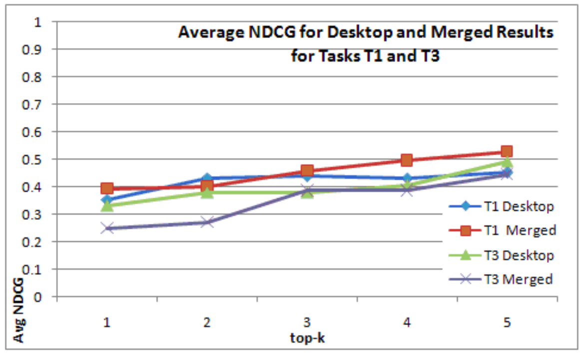


Figure 3.15 Average nDCG for Tasks T1 and T3

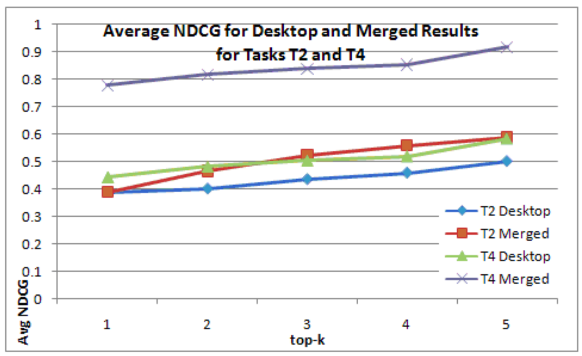


Figure 3.16 Average nDCG for Tasks T2 and T4

significant improvement over pure desktop search. Finally, party-planning task T4 shows about double nDCG improvement over regular desktop search. Therefore, we conclude that Question 1 could be answered positively and search over social services significantly complements relevance of the desktop search.

To answer Question 2, we considered nDCG in desktop and merged results for each type of search such as people (A), location (B) and general information (C) separately. Figure 3.17 presents nDCG results averaged over the users and search types. We observe that both people and general information search profit from the mixture of desktop and social search. nDCG for people search improved by about 17% and general information finding by 10%. On the contrary, nDCG of location search in the merged list decreased. Since nDCG of the social search results for location search was much lower than that of the desktop search their mixture did not provide any extra advantage. This result also explains the modest improvements in travelling-related tasks T1 and T3 where location search is important.

Our answer for Question 2 is that general information search benefits the most from social search, possibly due to the low desktop search effectiveness. Desktop search for people finding is already effective, but it is also significantly improved by the Web 2.0 results as provided by deskWeb 2.0. Location search degrades when using social search results, we assume that users did not have relevant information

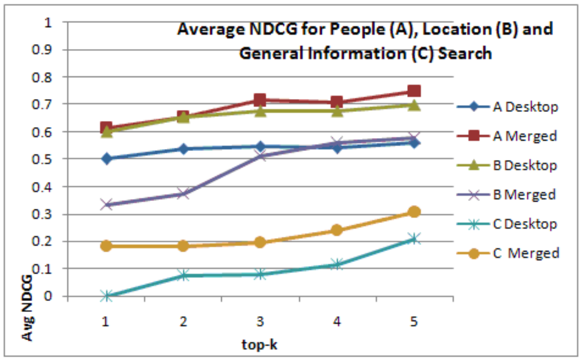


Figure 3.17 Average nDCG for three Types of Desktop Search

for this search type in the current networks.

3.7 Discussion

The first part of this chapter is devoted to usage of contextual links in desktop search applications. Considering desktop search evaluation, we presented the structure of an activity-based desktop dataset and defined its main requirements. Based on those guidelines, we developed and adapted a set of logging utilities that we called the Logging Framework. A distinctive feature of our approach is *activity metadata* collected from users, which can be exploited for improved desktop search.

In order to evaluate our implementation, we conducted a user study with about 20 users and collected a sample dataset within the L3S Research Center. Quantitative analysis of the dataset proved it to be a useful source of information for analysis of user behavior. In the meantime, the developed Logging Framework has been used by our colleagues in several research studies on context detection and desktop ranking.

Finally, we considered the problem of social desktop search. We presented the deskWeb 2.0 application integrating desktop search with social search over shared personal resources. A user study demonstrated that search over social resources increases overall search accuracy. It also suggests that people finding and search for general information benefit from such integration, while search for locations is more effective using desktop search alone.

In this chapter we have shown that desktop search could benefit from the rich activity metadata and corresponding contextual links. Such links have a high importance for the desktop domain, as information about social connections could be missing in this environment. But if social links become available too, we use them to further improve desktop search results.

We understand, that a large part of information resides outside the desktop. In the next chapter we consider the problem of enterprise search, including corporate social networks and blogging platforms. This type of search is different from desktop search, as context links are absent in such environment. But there is a lot of social information that could be collected from collaborative work in intranet. Therefore, a special focus is made on using available social connections to improve enterprise search.

4

Social Links in Enterprise Search

4.1 Personalization and Social Search

We have seen in the previous chapter how contextual and social links could improve the desktop search experience and support search over Web 2.0 applications. In this chapter we investigate the use of social links in enterprise setting and present our solution to personalized social search in enterprise. This chapter was written as a result of a research internship at IBM Haifa Research Lab and most of the current findings are also available in our joint paper[Carmel *et al.*, 2009].

4.1.1 Personalized Search

Personalizing the search process by considering the searcher's personal attributes and preferences while evaluating a query, is a great challenge that has been extensively studied in the information retrieval (IR) community, but still remains an interesting research problem [Belkin, 2008]. It is of great interest since user queries are in general very short and provide an incomplete specification of individual users' information needs. For example, searching for "IR" by an information retrieval student has a completely different meaning than searching by another who is interested in infra-red radiation.

Search personalization requires the capability of modeling the users' preferences and interests. This is usually done by tracking and aggregating users' interaction with the system. In general, such aggregation includes users' previous queries, click-through analysis, and even eye-tracking during the search session. Users' interactions are structured into a *user profile* that can be utilized during search. A user's profile is usually employed in two main scenarios, either through *personalized query expansion*, i.e., adding new terms to the query and re-weighting the original query terms based on the user's profile, or through *re-ranking* and *filtering* the search results while incorporating users' interests accordingly.

However the aggregation of user interactions comprises some difficulties. First, as we discussed in Chapter 2.1, many users consider user profiling as an activity which may violate their privacy. Users may feel uncomfortable with a system that accumulates their interactions and can potentially exploit that data for malicious actions such as spamming, phishing, or exposing it to the general public. Privacy issues are the main reason for new regulations enforced by many countries that put constraints on systems' sufferance to aggregate users' activity [Kobsa, 2007a].

Second, previous user interactions do not always provide a good indication of current needs. This is especially true for new users for whom only limited personal information exists, or when user preferences evolve over time. Moreover, the benefits that can be achieved through personalization vary accross queries. For some queries, different people may expect the same results, whereas for others different results are expected by individuals even for identical queries.

Finally, personalized search results make justifying the relevance of a specific result for a given query more difficult, as they are biased by query-independent personalized considerations. Some users may be confused when receiving different results for the same query due to the fact that their profile evolved during successive submissions.

4.1.2 Social Search

There are several alternative definitions of the concept *social search* [Hotho *et al.*, 2006b; Bender *et al.*, 2008; Amitay *et al.*, 2009]. It is used to describe several different aspects of search in Web 2.0 applications, finding a path between users in a social network, or to finding the set of users closest to a given user in the graph. In this chapter we use the notion of social search to describe the search process over "social" data gathered from Web 2.0 applications, such as social bookmarking systems, wikis, blogs, forums, social network sites (SNSs), and many others. Such a social search system represents different entity types (documents, persons, communities, tags) and their interrelations, and allows searching for all object types related to the user's query.

Social search provides an ideal testbed for personalization due to the explicit user interactions through Web 2.0 tools. A user profile that is derived from user feedback such as bookmarking, rating, commenting, and blogging, provides a very good indication of the user's interests. Furthermore, user profiles that are only based on explicit public social activity can be safely utilized without disrespecting the user's privacy[1]. Consequently, several previous works studied search personalization by profiling user interests based on public bookmarks aggregated from a social bookmarking system.

In addition, when the user's social network (SN) is available, the preferences of the user's related people can be utilized to assist in obtaining the user's preferences,

[1]For full transparency, Web 2.0 tools should better clarify to their users that any public social data provided by them can potentially be utilized by other social software applications.

assuming closely related people have similar interests. This is the main assumption behind collaborative filtering methods for recommendation systems, when user interests and preferences are predicted based on the preferences of "similar" persons. User similarity relationships are typically inferred through user feedback in the form of item rating. However, more recent approaches leverage implicit interest indicators [Claypool *et al.*, 2000], such as tags, views, or comments, as well as direct familiarity relationships [Kautz *et al.*, 1997], e.g., as reflected through connections in SNSs. We note that we refer to social networks in their broad definition, i.e., networks of people. Connected edges may represent any type of relationship, not only explicit familiarity [Guy *et al.*, 2008b].

4.1.3 Social Networks in Enterprise

In this chapter we study personalized social search in the enterprise based on the social relations of the searcher. We focus on re-ranking of search results by considering their relationships to users that belong to the searcher's social network. This approach is similar in spirit to the deskWeb 2.0 application, presented in a previous chapter, see Section 3.6.1. The assumption behind this personalization approach is that the preferences of other people, who are expected to have "similar" interests as the searcher, provide a good prediction for the searcher's preferences and can thus assist in revealing the search results that might subjectively satisfy the searcher's needs.

Personalized re-ranking of search results is done as follows: given a list of (non-personalized) results retrieved for the user's query, and a list of related users extracted from her social network, search results are re-ranked by considering their relationship strength with those users. Thus, documents that are strongly related to the user's related people are boosted accordingly.

To retrieve the user's social network, and the user-document relationship matrix, we use SaND [Ronen *et al.*, 2009], an enterprise social search system used in IBM[2]. For each user, SaND provides related people extracted through the user's SN. This is a ranked list of people, who relate to the user either through explicit familiarity connections (e.g., co-authorship of a wiki page or a connection within an SNS), or by some kind of similarity as reflected by their social activity (e.g., usage of the same tags or commenting on the same blog entry). Ranking of people is determined according to a weighting scheme that takes into account the overall related activity between two users [Guy *et al.*, 2008b]. In addition, SaND provides for each user all related documents (e.g., web-pages, blog entries), each associated with relationship strength to the user. A user may relate to a document through authorship, tagging or commenting, or by being mentioned in the page's content. The relative strength of each relationship type is determined by an appropriate weight [Guy *et al.*, 2008b].

[2] www.ibm.com

Using SaND, we would like to compare regular and personalized search that is based on user's social network. We are interested to see which particular type of social connections is more suitable for personalized ranking. In particular, we distinguish between connections between users who know each other in the real world, we call it *familiarity links*, and connections among people sharing common interests within an enterprise network, or *similarity connections*. To quantify a contribution of the social component, we include as a baseline the personalization built on user's profile from previously written documents and favorite Web pages and blogs.

We experiment with SN-based personalization considering three social network types: (1) *Familiarity-based* network, (2) *Similarity-based* network, and (3) *Overall* network that implies both relationship types. In addition to the user's SN, we consider the relevance of the search results to the user's topics of interest. These topics are approximated by a set of terms that are related to the user, including tags used by the user to bookmark documents, and tags used by others to bookmark that user [Farrell and Lau, 2006]. We assume that these related terms represent the user's interests, thus can be used to personalize the search results accordingly. We note that this assumption only holds for active taggers, or for users that were heavily tagged by others. Personalization is achieved by promoting search results that were tagged with these user's terms, either by the user or by others. We call this approach *Topic-based* personalization. As mentioned above, this approach has been extensively studied by previous works on personalized social search which construct users' profiles based on the tags they used for bookmarking [Kim and Chan, 2005; Noll and Meinel, 2007; Xu *et al.*, 2008]. We use it as a comparative baseline for an SN-based personalization approach for social search.

The different personalization approaches are evaluated by an off-line experiment and by a user study within the IBM organization. In the off-line study we follow the approach from [Xu *et al.*, 2008] and [Carman *et al.*, 2008], which evaluates search personalization as follows: given a user u who bookmarked a document d with the tag t, we assume that if u will search for t he will consider d relevant for t. Thus, any triplet (u, d, t) given by a social bookmarking system can be used as a personalized query for evaluation. The higher the rank of documents tagged by u with t, while simulating u searching for t, the better the personalization method is. The main drawback of this approach is that documents that were not tagged by u are considered irrelevant – a weak assumption that is not necessarily true. However, predicting u's tagging behavior indicates the system's personalization capability. We discuss the advantages and limitations of this evaluation approach in more detail in Section 4.4.

In the on-line study, we analyze a feedback from 240 employees exposed to the different personalization approaches studied in this work. Our main results are: (1) Personalized social search based on the user's SN significantly outperforms non-personalized search. A maximal improvement was achieved by the Overall social network which integrates familiarity and similarity relations. (2) As reflected by the user study, all three SN-based strategies significantly outperform the Topic-based strategy, which improves only slightly over non-personalized results. (3) The integration of related terms with related people in the user profile slightly improves the search results. (4) The off-line evaluation is consistent with the user study in confirming the superiority of SN-based personalization strategies, and the contribution of additional related terms to the SN-based user profile. However, several discrepancies between the two evaluation methods raise concerns about its reliability in ranking alternative personalization approaches.

The rest of the chapter is organized as follows. Section 4.2 reviews related work on search personalization in general and in particular for social search. Section 4.3 discusses the different personalization approaches we study in this work. Section 4.4 describes the off-line experiment and the on-line study, and the results we obtained. In Section 4.5 we discuss the obtained results.

4.2 Relevant Background

Our topic of interest is an intersection of Personalized Search and Social Search areas. The former deals with building user models based on previous user interactions saved in search engine logs. It aims to personalize ranking algorithms based on profiles of the users. The latter tries to improve the search and browsing experience, given access to user's social network, tags, bookmarks or comments. Social connections between users could be explicitly represented as links between friends or colleagues in popular social networks, or could be implicitly inferred when users co-author a document or comment on a blog. Below we review recent research activity in the areas of Personalized Search, Social Search and combination of both.

4.2.1 Personalized Search

In recent years many researchers utilize query log and click-through analysis for web search personalization. In [Qiu and Cho, 2006b], the authors combine a topic-sensitive version of *PageRank* [Haveliwala, 2002] with the history of user clicks data. Then, they automatically tailor it to a user to personalize the search results. Another work [Joachims *et al.*, 2005] studies clicks applicability as implicit relevance judgments. They show that user's clicks provide a reasonably accurate evidence to the user preferences. An approach from [Tan *et al.*, 2006] proposed a language modeling for mining query history. Their small-scale study demonstrated significant improvement of personalized web search with a history-based language model over regular search. The user modeling approach described in [Shen *et al.*, 2005] is based on a decision-theoretic framework to convert implicit feedback into a user profile that is used to re-rank search results.

In [Kraft *et al.*, 2006b] a context is mined from the web logs in a form of text vectors. This context is used for query rewriting and evaluated on several major search engines. As introduced in [Agichtein *et al.*, 2006b], alternative user modeling method could be used to apply a set of rules to a query log. While user models are usually targeted at search personalization, they could also be applied for personalized information filtering, as was shown in [Yang and Jeh, 2006b] who analyze click history for the identification of regular users' interests. Recent work on "groupization" [Teevan *et al.*, 2009] shows that combining implicit user profiles from several related users has a positive impact on personalization effectiveness.

In addition to regular web log data, several works consider personalization using desktop data and external resources. For example, in [Teevan *et al.*, 2005b] the authors index desktop information and experiment with different representations of users, documents and queries for personalized web search. Another research group [Chirita *et al.*, 2007b] explore personalized query expansion based on users' desktop information. Several approaches for personalized Web search are based on global interests using the *Open Directory Project (ODP)* categories [Liu *et al.*, 2004; Chirita *et al.*, 2005]. In [Liu *et al.*, 2004] the authors map previously visited pages to *ODP* categories and use this mapping to build a user profile. Another work [Chirita *et al.*, 2005] proposes a personalized version of *PageRank*, in which a hand-picked set of preferred users' categories are applied for result re-ranking. Profiles created implicitly from user's bookmarks can be also used effectively for personalized search [Kim and Chan, 2005].

There are several algorithms trying to predict applicability of personalization while considering the current context of the user's task on query submission [Teevan *et al.*, 2008; Dou *et al.*, 2007; Luxenburger *et al.*, 2008], using either user context or click entropy. A study from [Broder, 2002] shows that quality of search results can be harmed when personalizing unambiguous or navigational queries. In [White and Drucker, 2007], a study over about 2000 participants suggested that users can be also divided into navigators and explorers. Both categories of users might require different personalization of results, including different result presentation and user interface. An automatic identification of a user goal in web search is described in [Lee *et al.*, 2005], where goals include informational and navigational queries [Broder, 2002; Rose and Levinson, 2004].

4.2.2 Social Search

The amount of social data is rapidly growing and has become a main focus of research on social search. Recent work [Heymann *et al.*, 2008] reports that in 2008 around 115 million bookmarks were available on the Delisious social bookmarking site. In [Wu *et al.*, 2006] authors show how Delisious data and its tags can be used for semantic annotations of web pages. In addition, as shown in [Chirita *et al.*, 2007a], tags could be automatically created on-the-fly in a personalized manner, using both page content and desktop data of a user. Following the language modeling approach, a theoretically sound generative model for social annotations is presented in [Zhou *et al.*, 2008].

A formal model for folksonomies and ranking algorithms called *Adapted PageRank* and *FolkRank* are defined in [Hotho *et al.*, 2006b]. *FolkRank* is used for the generation of personalized rankings of entities within the *folksonomy* and for the recommendation of tags, users and resources. Lately, the work presented in [Bao *et al.*, 2007] proposed two alternative algorithms, *SocialSimRank* and *SocialPageRank*. Both are based on social annotations and corresponding connections between pages, annotations and users. A comparative evaluation study of these algorithms and a few novel algorithms are described in [Abel *et al.*, 2008]. A similar page popularity measure *SBRank* was proposed in [Yanbe *et al.*, 2007], it is proportional to a number of existing social bookmarks.

4.2.3 Personalized Social Search

Several approaches for directly or indirectly employing users' social relations for personalization exist. A re-ranking method presented in [Noll and Meinel, 2007] is based on users' tag profiles which are derived from her bookmarks in Delisious. The tags of each search result on the site are matched against the user's profile. The problem of automatic user profile generation is addressed in [Au-Yeung *et al.*, 2008]. The authors investigate how accurate user profiles can be built from Delicious data. They extract *personomy* for each of Delisious users and cluster the bookmarks based on a *modularity* measure. Both approaches above successfully use social networks for building user profiles, however, they do not study direct application of social links for personalization.

Another approach [Bender *et al.*, 2008] directly exploits social relations by combining semantic and social factors in the ranking. The users, tags and documents are represented as nodes in a "friendship graph", in which edges are extracted from relationships like links, content, tagging and rating. Ranking is based on *UserRank*, an algorithm derived from the PageRank computation on the friendship graph. A document receives an extra "friendship" score when tagged by a user's "friend". Similarly to this approach, our method personalizes the score of a document for a specific user if it has been bookmarked, authored or commented by people related to the user, or tagged with terms related to her. We further analyze the value of personalization according to different relationship types, in particular familiarity and similarity.

It has been shown in [Xu *et al.*, 2008] that personalization approach could use social relations indirectly. The *Topic Adjusting* algorithm is built on top of *Open Directory Project*, *Dogear* and Delicious data [Millen *et al.*, 2006]. Users' interests are inferred using the topics of their tagged pages. The relationship weights in the user-page matrix are defined based on the number of user annotations assigned to a page. This work has some similarity with our approach, however, our personalization method explicitly uses familiarity and similarity scores to model direct and indirect relations between users. For evaluating the *Topic Adjusting* algorithm, authors introduce a new method for automatic evaluation of personalized search, in which the user's tags are used as queries and all documents bookmarked by this user with that tag are considered relevant. More details on automatic evaluation of personalized search based on social bookmarking data can be found in [Carman *et al.*, 2008]. We adopt a similar evaluation approach for our experiments and complement it with a large-scale user study.

Overall, the works discussed above show high potential of social links for personalization. However, they do not consider contribution of different types of social connections. Also, the experimental evaluation of the relevant methods rely on simulated experimental setup, which validity has not been widely accepted so far. In this chapter we both experiment with different types of social links and compare results of a simulated evaluation with outcomes of a user study.

4.3 Personalized Search System

In this section we describe the social search platform used for our study, and the types of user profiles we experimented with. We start with an introduction of Lotus Connections and SaND, two working IBM systems, which we used as a search platform for our personalization experiments.

4.3.1 Lotus Connections and SaND

IBM Lotus Connections (LC)[3] is a social software application suite for organizations that was introduced in 2007. It contains (as of version 2.0) five social software applications: profiles – of all employees, a social bookmarking system, a blogging service, a communities service, and activities (not discussed in this work). In our study we experimented with LC tools as used by IBM employees. Dogear[Millen *et al.*, 2006], LC's social bookmarking application, allows users to store and tag their favorite web pages. Over 90% of the bookmarks are public (visible to all other users) and about half are intranet pages, while the other half are external internet pages. Dogear includes 743,239 public bookmarks with 1,943,464 tags by 17,390 users. Blog Central [Huh *et al.*, 2007], LC's blogging system, has 16,337 blogs, 144,263 blog entries, with 69,947 users. LC's communities service contains over 2,100 online communities, each with shared resources and discussions, with a total of over 50,000 members.

Social Networks and Discovery (SaND) [Ronen *et al.*, 2009], is an aggregation tool for information discovery and analysis over the social data gathered from all LC's applications. It leverages complex relationships between content, people and tags, and its integrated index supports a combination of content-based analysis and people-based analysis. SaND provides several social aggregation services including social search, item recommendations, people recommendations, finding social paths between people, and additional social network services. SaND provides social search

[3]http://www-01.ibm.com/software/lotus/products/connections/

over the social data using a unified approach [Amitay *et al.*, 2009] in which all system entities (documents, persons, groups, tags) are searchable and retrievable. As part of its analysis, SaND builds an entity-entity relationship matrix that maps a given entity to all related entities, weighted according to their relationship strength. The entity-entity relationship strength is composed of two types of relations:

- **Direct relations**: Figure 4.1 shows all direct relations between entities that are modeled by SaND. For example, a user is directly related to: (1) a document: as an author, a tagger, or a commenter; (2) another person: as a tagger of, or tagged by that person, as a friend as stated in several SNSs that exist in the enterprise, or through the enterprise's organizational chart (direct manager/employee); (3) a tag: when used by the user for bookmarking, or when used by others to tag that user; (4) a group: as a member or an owner. Other direct relations and their corresponding relative weights are shown in the figure.

- **Indirect relations**: Two entities are indirectly related if both are directly related to the same entity[4]. For example, two users are indirectly related if both are related to another user, e.g. if both have the same manager, or if both tagged the same document.

The overall relationship strength between two entities is determined by a linear combination of their direct and indirect relationship strengths. More details on score calculation and implementation issues are described in previous work on social network aggregation and social search [Guy *et al.*, 2008b; Amitay *et al.*, 2009].

4.3.2 User Profile Types

We experimented with three types of social networks for personalization. Each network maps a user to a list of related users weighted according to their relationship strength.

Familiarity SN

Familiarity between two individuals is considered according to indicators that they know each other [Guy *et al.*, 2008b]. A direct familiarity relation exist if both persons are marked as friends in one of the enterprise SNSs, or when one is the direct-manger/employee of the other. In addition a person is familiar with those she tagged, but not vice versa. Indirect familiarity relations are defined when the two persons are both authors of the same paper, patent, or wiki-page, or when both have a common manager (team members).

[4] Currently, only indirect relations of level two are considered, i.e., two-length path in the entity graph. However, the model can easily be extended to support indirect relations of any level.

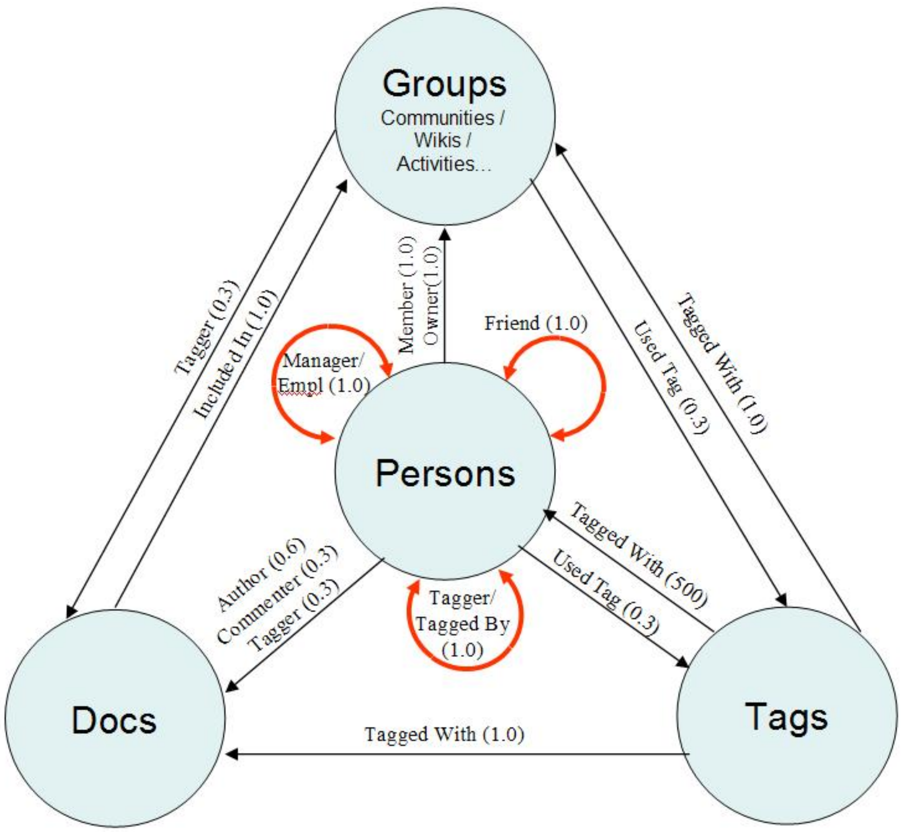


Figure 4.1 Direct relations between entities modeled by SaND. The relative relationship strengths appear on the graph's edges. Familiarity relations are colored red (bolded).

In order to extract the user's Familiarity network, we use SaND to extract all the user's related people and to filter out all non-familiar people which do not obey the above constraints. In addition, the relationship strength between the two is modified to be based on familiar relations only. More details on the familiarity relationships and the calculation of the familiarity score can be found in [Guy *et al.*, 2008a].

Similarity SN

Similarity between two individuals is considered according to common activity in the context of LC's social software: co-usage of the same tag; co-tagging of the same document; co-membership of the same community, or co-commenting on the same blog entry. Similarly to the familiarity case, in order to extract the user's Similarity network, we use SaND to extract all related people and retrieve (and re-weight) only people which obey the above constraints.

Overall SN

Besides the Familiarity and Similarity networks, we also examine the user's Overall social network, which contain all related persons according to the full relationship model.

One of our research tasks is to measure the relative effectiveness of the social-based personalization compared to existing approaches. For this reason, we also consider a topic-based personalization using text- and tag-based user profiles, similar to the recent approaches in personalized social search. This fourth personalization algorithm serves as a baseline for performance evaluation of different types of social connections.

Topic-Based

The user's topics of interests are represented by a set of terms that are closely related to the user. Directly related terms are tags used by the user to tag documents and other people, and tags used by others to tag that user. Indirectly related terms are those that are related to the user through other entities (e.g. all tags of a document bookmarked by the user). The user's top related terms retrieved by SaND serve as the user's Topic-based profile.

Equation 4.1 shows the score of tag t for user u, where $ief(t)$ is a the *inverse entity frequency* score of a tag which is inversely related to the tag's frequency (similar to terms *idf*), $w(t, u)$ is the number of times u tagged and were tagged by t, and $w(e, u)$ is the relationship strength between entity e and u.

$$S(t,u) = ief(t) \cdot (w(t,u) + \sum_{e \in U \cup D} w(t,e) \cdot w(e,u)) \tag{4.1}$$

For more details how tags are scored by SaND in relation to a specific user see [Amitay *et al.*, 2009].

4.3.3 Personalizing the Search

A user profile is constructed on the fly when a person logs into the system. For a user u, SaND retrieves $N(u)$ – the ranked list of users related to u, and $T(u)$ – the ranked list of related terms. These two lists are then used as the user profile to personalize the search results for all user's queries during the search session.

Given the user profile, $P(u) = (N(u), T(u))$, the search results are re-ranked as follows:

$$\begin{aligned} S_p(q,e|P(u)) = \quad & \alpha S_{np}(q,e) + (1-\alpha)[\beta \sum_{v \in N(u)} w(u,v) \cdot w(v,e) \\ & +(1-\beta) \sum_{t \in T(u)} w(u,t) \cdot w(t,e)] \end{aligned} \tag{4.2}$$

$S_p(q, e|P(u))$ is the personalized score of entity e to query q given the profile of user u. $S_{np}(q, e)$ is the non-personalized SaND score of e to q. Since we only re-weight the search results, only entities with positive score are considered. $w(u, v)$ and $w(u, t)$ are the relationship strength of user v and term t to u, as given by the user profile. Similarly, $w(v, e)$ and $w(t, e)$ are the relationship strength between v and t to entity e, as given by SaND.

Thus, an entity is first scored by SaND according to its non-personalized scoring mechanism, and then the entity score is modified according to its relationship strength with users and terms in the user profile.

The equation has several parameters that control the amount of personalization. First $N(u)$ is determined according to the SN type used for personalization (Familiarity, Similarity, Overall). Second, the number of users and terms in the profile are configurable. Third, the parameter α controls the relative weight of the personalization score compared to the original non-personalized score, and β controls the relative weight between people and terms for personalization.

In the next section we describe several experiments we conducted with some of these controllable parameters. We test our main hypothesis that familiarity of similarity connections between users could be directly exploited for personalizing search results. We identify which connection types are more suitable for personalization and compare them to a non-personalized search and topic-based personalization baselines.

4.4 Evaluation

In this section we describe the experimental methodology used to evaluate the SN-based personalization approach, the results of an off-line study using a bookmark-based evaluation, and a user study we conducted in IBM.

4.4.1 Evaluation Methodologies for Personalized Social Search

Evaluating personalized search is a great challenge since relevance judgments can only be assessed by the searchers themselves – only the users can subjectively judge whether a specific result answers their personal need. Therefore, existing IR evaluation benchmarks based on judged queries, each associated with a set of relevant results objectively assessed by experts, cannot be utilized for personalized search evaluation.

Existing evaluation approaches for personalized search are often based on a user study, where participants are asked to judge the search results for their personal queries in a personal manner, thus alternative personalization techniques can be comparatively analyzed. However, appropriate user studies with a reasonable number of participants are very expensive to accomplish, therefore, such studies are uncommon and often limited to a small and a biased sample. Alternatively, users' implicit feedback such as clicking on a specific result (while un-clicking other results), can be interpreted as personal relevance judgment. Clicks, however, are not necessarily the best indicators for user satisfaction with results - clicking on a result does not necessarily mean it is relevant, while un-clicking does not always imply irrelevance. Furthermore, such evaluation is only feasible for a live system with enough users who use it on a regular basis.

Social search applications provide richer sources for user feedback that can be used for regular personalized search evaluation. User feedback such as rating, tagging, and commenting, indicates the user's interest in a specific document. Recently, several works utilized data from *Delicious* to evaluate personalized search methods [Xu *et al.*, 2008; Carman *et al.*, 2008]. In this approach, any bookmark (u, d, t) which represents a user u who bookmarked a document d by a tag t, can be used as a test query for personalized search evaluation. The main assumption behind is that any document tagged by u with t (including d) is considered relevant for the personalized query (u, t) (i.e. u submits the query t).

Therefore, the bookmark triplets (u, d, t) extracted from a social bookmarking system provide an almost unlimited source of personalized test queries to be used for personalized search evaluation. Given the bookmark (u, d, t), a personalized search system is evaluated according to its ability to highly rank the corresponding documents. A good personalization policy is expected to differentiate between two similar tested queries (u_1, d_1, t) and (u_2, d_2, t), promoting d_1 while serving (u_1, t), and d_2 for the query (u_2, t).

There is a delicate issue with bookmark-based evaluation. The search system is already "aware of" the association between d and t, as realized by u, hence this information can be exploited for over tuning. For example, given the query (u, d, t), a personalization approach that retrieves only the documents tagged by u with t will inevitably outperforms other personalization alternatives, since any other document is considered irrelevant. However, this "over-tunned" personalization policy is restricted to queries that were previously used as tags by the user, hence it will totally fail for other personalized queries. This limitation cannot be disclosed by the bookmark-based evaluation methodology.

In order to eliminate the dependency between personalization and evaluation, and to simulate the personal query (u, d, t) with no prior knowledge on the user's association between t and d, we have to mask u bookmarking of d. Masking is done as follows: for each personal query (u, d, t), we first "hide" that bookmark from the search system before handling the query (u, t). The system is instructed as this specific bookmarking has never happened – d content is not enriched by the tag t (unless d was tagged with t by others), t is taken out from the user profile (unless t relations with u is derived from other resources) and u's relations with other entities that are based on this bookmark are modified accordingly. This masking guarantees that personalization is evaluated without any prior knowledge on u relations with d and t.

Note that personalized methods that better predict their users' interests, as reflected by their tagging activity, will be favored by that evaluation methodology. This is definitely one of the main characteristics that are expected from a personalized search system, hence such evaluation can successfully prioritize alternative personalization strategies. However, the bookmark-based evaluation approach still suffers form the incompleteness problem – not all documents tagged by u with t are relevant for u while searching for t, and not all documents not tagged by u with t are necessarily irrelevant. This limitation is partially handled by the huge amount of personalized queries available for evaluation. But we believe that conclusions based on such evaluation should be supported by alternative evaluation methods - an approach that was taken by us in this work. We first evaluate and tune our personalized social search system with the bookmark-based evaluation, using *Dogear's* bookmarks as personalized queries, and confirm our findings with an extensive user study based on 240 participants that subjectively judge the results for their 577 personal queries. To the best of our knowledge, this is the first study that (1) eliminates the dependency between personalization and evaluation that inherently exists in bookmark-based evaluation; (2) validates the bookmark-based evaluation methodology for personalized search by comparing its findings with the results of an independent user study.

4.4.2 Experimental Setup

In our evaluation we wanted to see how personalization based on different types of social links compares to existing topic-based personalization. A tuning of coefficients of Equation 4.2 was not considered as a main goal and we fixed them so that amount of personalization is equally balanced with regular search results. The weights corresponding to relationship strength were provided by SaND and calculated as dot products.

We experimented with several personalization methods that are based on the user's social network, and on the set of the user's topics. A user profile, $P(u) = (N(u), T(u))$, is based on $N(u)$, a ranked list of the user's related people, as given by the user's SN, and $T(u)$, a ranked list of the user's related terms. The user profile is constructed on the fly while the user logs into system and is used to personalize (re-rank) the search results by Equation 4.2 throughout the search session[5].

4.4.3 Off-line Study

In the off-line study we used Dogear's bookmarks as personal queries. For each personalization method, we randomly selected 2000 bookmarks, and for each bookmark (u, d, t) we masked its existence from the search index and the user profile, to completely hide the relations between u, d, and t. Then, t was submitted as a query to

[5] In the off-line study, since a bookmark is hidden prior to handling the personal query, we reconstruct the user profile after bookmark masking and before query submission, to guarantee that the user profile has no dependency on the tested bookmark.

SaND and 1000 results (documents) were retrieved. Other retrieved entities such as persons and tags were ignored, as they are not suitable for evaluation by the off-line approach. The search results were re-ranked using u's profile, and were evaluated by measuring average-precision (AP) and reciprocal rank (RR), while considering all documents tagged by u with t as relevant answers. After completion, the hidden bookmark was returned to the collection before processing the next tested bookmark.

Note that due to the masking process, d will be retrieved for t only when t appears in the original content of d , or when d was associated with t by others. The personalization methods differ in the way they re-rank d. SN-based personalization methods will advance d when it is related to at least one person in u's social network. Topic-based personalization will boost d if tagged by at least one of the terms related to u.

Off-line Study - Main Results

Table 4.1 shows the mean-AP (MAP) and mean-RR (MRR) results for the configurations we experimented with, setting $\alpha = \beta = 0.5$. The top rows show the results of SN-based personalization with top-5 related people and with no related terms. The bottom rows show the results with top-5 people and top-5 terms.

User Profile		MAP	MRR
Non-Personalized		0.156	0.187
No Terms	Familiarity-SN	0.389	0.444
	Similarity-SN	**0.423**	**0.476**
	Overall-SN	0.388	0.442
With Terms	Topic-based	0.426	0.475
	Familiarity-SN	0.412	0.461
	Similarity-SN	**0.452**	**0.510**
	Overall-SN	0.410	0.461

Table 4.1 Bookmark-based evaluation of personalized social search. User profile is based on the top-5 related people and top-5 related terms.

There are several interesting insights from these results. First, all personalization methods significantly outperform non-personalized search (one-tailed unpaired t-test, $p < 0.001$). The MAP of the Similarity-based strategy is almost three times higher than that of non-personalized search.

Second, the Similarity-SN significantly outperforms (one-tailed unpaired t-test, $p < 0.001$) the Familiarity and the Overall networks, and maybe surprisingly, the Overall-SN is slightly inferior (almost identical) to the Familiarity-SN. This indicates that similarity relations better predict the user' preferences than familiarity relations. We do not have a good explanation for the inferiority of the Overall network, especially because this result contrasts the results of the user study discussed in the following.

We hypothesize that better integration of the similarity and familiarity relations by SaND might result in better performance of the Overall network.

Third, Topic-based personalization with no SN data improves the search significantly, and outperforms the Familiarity and the Overall SN. Integrating the user's related terms with the related people improves the search performance of all network types. The best result achieved while integrating the top-5 similar people with the top-5 related terms. In the following we further experiment with that integration task.

User Profile Size

The size of the user profile is determined by the lists $N(u)$ and $T(u)$. These lists boost the search results through their relationship strengths with those related people and terms. There is a risk that adding too many people or terms to the user profile may personalize too much, disregarding new relevant items that have not been discovered yet by the user's community. Therefore, finding an "optimal" user profile size is an important factor that significantly affects personalization effectiveness.

The size of $N(u)$ is controlled by two parameters, max_N, which sets the maximum number of (top scored) related people in the profile, and θ_N which determines a threshold on the relationship score. This threshold guarantees that only closely related people will be part of the user profile. Therefore, "socially active" users will have max_N related people in their profile, while others may have much less. Similarly, max_T and θ_T determine the number of terms in the user profile.

We experimented with max_N and max_T, while fixing the θ values to 0.0 (i.e., each user has max_N people and max_T terms in the profile, unless SaND retrieves fewer related people or terms for that user). Figure 4.2 shows the MAP for the different SN types, averaged over 2000 personal queries, while fixing the number of related people to 5 and varying the number of related terms. Similarly, in Figure 4.3 we fix the number of terms to 5 and vary the number of people in the profile.

According to Figure 4.2, the maximum performance is achieved while adding 4-6 related terms to the the user profile, improving the MAP by 4-5% for all network types. Adding too many terms degrades the performance, even lower than with no terms at all, probably due to overstated personalization. This means that several user's topic of interest are considered at once and their summary shifts a focus of personalization to wrong direction in vector space model. The effect is similar to *topic drift*, which happens when user's interests are given too high priority and results are biased so, that initial query is no longer important.

Figure 4.3 shows the performance of adding related people to a user profile with 5 related terms. We can see improvement only while adding similar people to the profile. Maximum improvement is achieved with 3 similar people ($MAP = 0.457$), then the performance is dropped for additional people. In contrast, familiar people constantly harm the search performance while added to the profile.

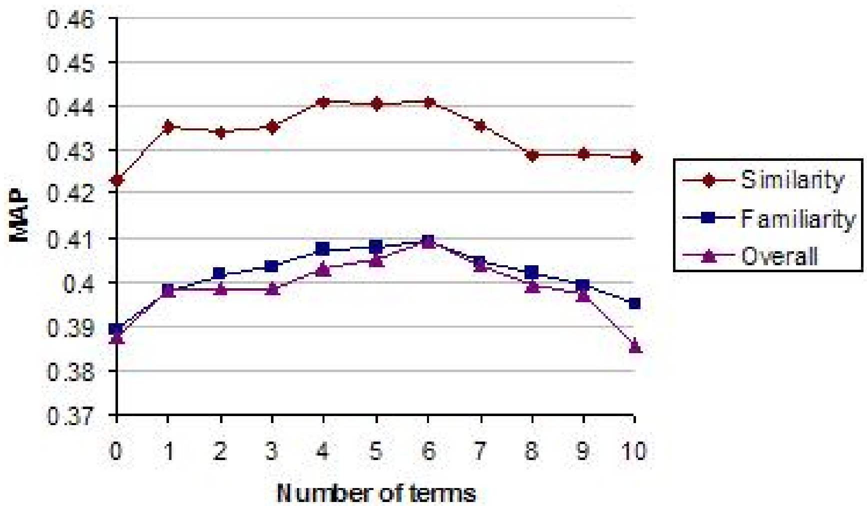


Figure 4.2 MAP for the different SN types, averaged over 2000 personal queries, for different number of related terms while fixing the number of related people to 5.

These results suggest that according to the bookmark-based evaluation, an "optimal" user profile should be based on a few similar people and a few related terms. However, we note that these results fit an "average user" while it is quite clear that user profiles should be subjectively adapted according to the user's personal characteristics. One might be interested also with the θ values, adapting the optimal profile size for each user in a personal manner.

The off-line bookmark-based evaluation can be easily applied in very large scales, without any user intervention, hence it can be efficiently used for tuning the system parameters, and to efficiently examine alternative personalization strategies. However, due to the limitations of this approach, conclusions based on that evaluation should be validated by applying complementary evaluation methods.

4.4.4 User Study

To complement and validate the results of the off-line evaluation, we ran a user study in IBM, asking participants to assess the search results for their queries in a personal manner. Each participant was asked to assess two personal queries and was given the opportunity to evaluate more queries, as much as she likes. In order to simulate personal queries, for which the user has personal information needs, we recommended the participant a set of tags she was tagged with in IBM's people tagging application [Farrell and Lau, 2006], to be submitted as personal queries, assuming that such tags represent interesting topics, or at least familiar to the user. The participant was asked

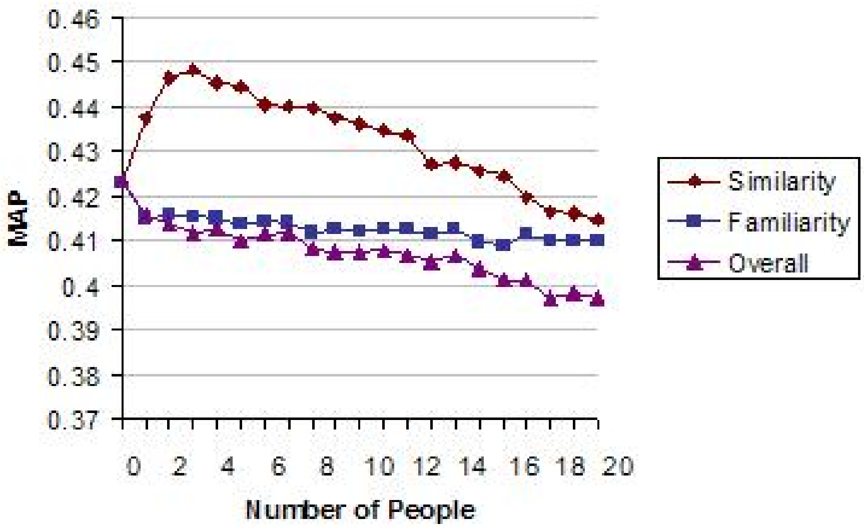


Figure 4.3 MAP for the different SN types, averaged over 2000 personal queries, for different number of related people while fixing the number of related terms to 5.

to select two terms out of the recommended tags, or alternatively submit their own (personal) queries for assessment. After completion, participants were encouraged to comment on the search experience with the system.

Experimental Setup

We considered users who had at least 30 people in both their Similarity and Familiarity networks, and at least 30 related terms. We note that this sample does not represent the entire population of employees, but rather active users of the LC system, who are the target population for our search system. We sent a link to the study with a request for participation to a random sample of 645 of these users and got a response from 240, who judged 577 personal queries (91% of the queries were personal terms suggested by us while the rest were original queries selected by the participants). Our study participants originated from 28 countries, spanning over the globe and over all IBM divisions.

Each participant of the study was first classified randomly into one of eight classes, each associated with a different personalization strategy; the eight strategies we experimented with are shown in Table 4.2. Participants were not aware of the personalization type selected for them.

For each study participant, the user profile was set according to the strategy of the class she was associated with, and for each user query, the search results retrieved by SaND, which were re-ranked according to the corresponding user profile, included 10 top relevant pages and 10 top relevant people, each judged by the user as non-relevant, relevant, or highly relevant. Figure 4.4 shows the entrance page users obtained while taking the study, including the terms suggested as personal queries, and the results to be judged after the query was issued. Looking at the figure, please note that most terms suggested as personal queries for this specific user are ambiguous and can be interpreted in several ways. For example, the subjective meaning of "pasta" for this user is probably a code-name of a research project and not a noodle type.

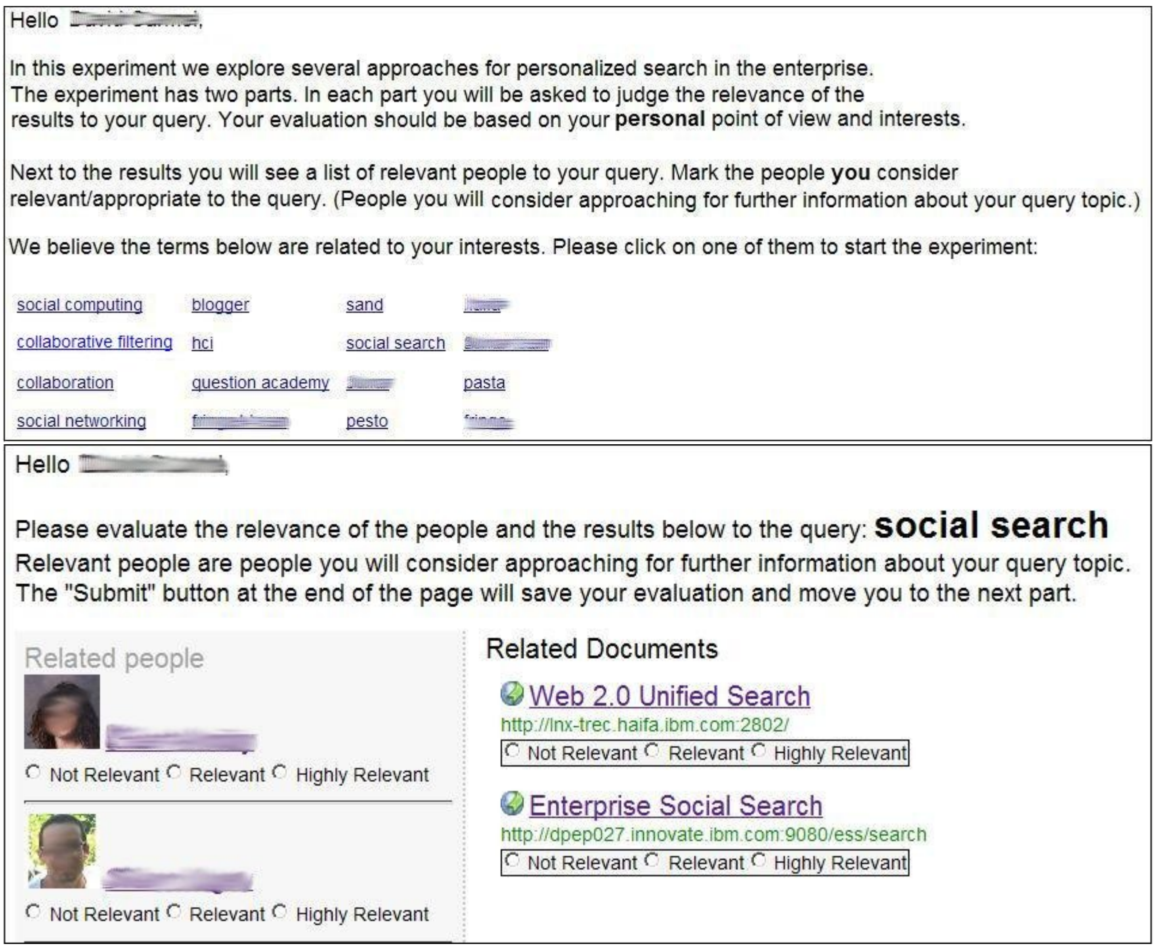


Figure 4.4 User study pages. Top: the entrance page with instructions and the personal terms suggested for querying. Bottom: a snapshot of the results page.

Most of the comments we got were very positive. One participant wrote: " *[...] thanks for the opportunity to try out the research project! The results were quite interesting, I found content on the topic I didn't know to exist [...]*". Another one wrote: "*[...] I am eager to see the evaluation metrics from these experiments. Such an outcome motivates to pay a lot more attention to social factors in all personalized applications [...]*".

Results

The quality of search results was measured by the normalized discount cumulative gain (NDCG) and by precision at 10 (P@10), averaged over the set of judged queries, for each of the classes. For DCG calculation we used gains (0,1,2) for the three relevance levels respectively, and the discount function used was $-log(rank+1)$. Normalization (NDCG) was done by dividing the DCG value with an ideal DCG value calculated as all results are highly relevant. For P@10 calculation, we considered any positive judgment as relevant.

Table 4.2 shows the precision of the search results, as measured by NDCG@10 and P@10, for the eight personalization strategies. The general high satisfaction from the social search system is reflected by the high NDCG@10 (> 0.5) and P@10 (> 0.6) achieved in all classes.

User Profile		Judged Queries	NDCG @10	Delta (%)	P@10
Non-Personalized		79	0.511	–	0.61
No Terms	Familiarity	71	0.560	9.7	0.68
	Similarity	78	0.550	7.6	0.68
	Overall	69	**0.597**	**16.9**	**0.73**
With Terms	Topic-based	81	0.518	1.4	0.64
	Familiarity	68	0.561	9.9	0.69
	Similarity	69	0.565	10.7	0.71
	Overall	62	**0.581**	**13.8**	**0.72**

Table 4.2 User study: The precision of the search results of the personalized search strategies, measured by NDCG@10 and P@10. The Delta column shows the improvement in NDCG@10 over non-personalized search.

The main outcomes of the study are: (1) As in the off-line study, all personalization methods outperform the non-personalization strategy. These differences are found to be significant for all strategies except for the Topic-based one (one-tailed unpaired t-test, $p < 0.05$). (2) A maximal improvement was achieved by the Overall network, 16.9% improvement in NDCG@10 without terms and 13.8% with terms. (3) The Similarity network outperforms the Familiarity network with and without terms, and both significantly outperform the Topic-based strategy. (4) Related terms slightly improve search effectiveness when applied alone (1.4%), and when added to the Similarity and the Familiarity SN, in agreement with the off-line study; however, they decrease the performance of the Overall network. This result indicates that optimal integration between SN and personal terms should be further studied for each of the networks separately, as currently system parameters are commonly set to all user profile types.

There are several substantial differences between the two evaluation methods. Both methods confirm the significant contribution of personalization for social search, and the superiority of using similar people over familiar people in the user profile. However, the Overall network, the "shining star" of the user study, performs the worst according to the off-line study. Similarly, the topic-based strategy, with marginal contribution in the study, perform very well in the off-line study. In Section 4.5 we discuss possible reasons for these discrepancies and whether the conclusions derived from the bookmark-based evaluation have any value at all.

Personalized People Search

Table 4.3 shows the distribution of the relevant people retrieved by the different SN based strategies (accumulating all positive judgments as relevant). On average, people retrieved by the Familiarity network were judged as more relevant for the user queries compared to to other networks and to the non-personalized search. We can clearly see an increase in the percent of relevant people while moving from non-personalized, to Overall, Similarity, Familiarity, respectively. However, this result is likely to be affected by the natural bias of users to people they are familiar with.

User Profile	Relevant people (%)
Non-Personalized	47.8
Familiarity	**55.8**
Similarity	52.2
Overall	51.8

Table 4.3 The relevance distribution of retrieved people for the different SN types.

Several participants mentioned the difficulty in judging the relevance of people to their query, mostly because of unfamiliarity. Someone wrote " *"...It would be good to include more information on the people that are shown on the results, like their Job Role/Title. This would help to identify on a first look their relevance or not."*.

Actually, participants had the opportunity to open the home-page created by SaND for each retrieved person, viewing his role, communities, list of publications, blogs, and more. However, it seems that judging unfamiliar people's relevancy is more difficult than judging unfamiliar documents' relevancy. Indeed, 21% of the retrieved people were not judged by the participants, relative to 9% only of retrieved pages.

4.5 Discussion

In this chapter we investigated personalized social search based on the user's social relations. We studied the effectiveness of several social network types for personalization, and evaluated their contribution by an off-line study and by a user study within IBM. Our results showed that according to both evaluations, social network based personalization significantly outperforms non-personalized social search. In addition, as reflected in our user study, all three SN-based strategies significantly outperform the Topic-based strategy, which improves only slightly over non-personalized results. In particular, personalizing the search by the Overall social network, which incorporates similarity and familiarity relations, improves the search precision by 16.9%. These findings clearly demonstrate that the user's social network should be taken into account as a productive mean for search personalization.

The bookmark-based evaluation for search personalization has the advantage that it can be easily applied in very large scales, without any user intervention. To validate its outcomes we compared the results we got from the off-line study with those of the user study. Our results show that there are several substantial discrepancies between the two evaluation methods. In particular, according to the off-line study, the Overall network is inferior to the Similarity and Familiarity networks, and to the Topic-based strategy, while in contrast, according to the user study, the Overall network performs the best.

These disagreements are not unexpected – there are several differences between the two evaluation approaches. In the off-line study, participants were randomly selected from all Dogear users, while in the on-line study we focused on heavy users of the LC system. In addition, off-line queries were based on the user's tags while on-line queries were based on the tags the users were tagged with. Furthermore, the bookmark-based evaluation method predicts the user's bookmarking activity while the on-line study measures directly the users' personalized relevance judgments. As a result, the off-line approach discriminates against authored or commented documents, and biases tagged documents, while this discrimination does not exist in the user study.

The extreme success of the Similarity network in the off-line study, in contrast to its comparable performance with other networks in the user study, can be explained by the fact that social activity of similar people better predicts the user's social activity than the activity of familiar people. This also interprets the difference in performance of the topic-based strategy, which performs reasonably well in the off-line study while exhibiting inferior effectiveness compared to SNs in the user study. It seems that similar related people and related terms are strongly associated with the tested bookmark's document, therefore, in the off-line run, this document is advanced even after bookmark masking. In contrast, according to the study results, interesting/relevant documents are associated with similar and familiar related people much more than with related terms.

The disagreements between the bookmark-based evaluation and the user study put a question-mark on its reliability for personalized search evaluation, especially for ranking different personalization approaches. Considering the study results as ground truth, some of the "conclusions" derived from the off-line evaluation were proved to be wrong. However, we believe that it might have some benefits, mostly for parameter tuning while fixing the personalization strategy. For example, finding a good combination of related people and terms in the user profile, or searching for appropriate α and β for Equation 4.2. In any way, it seems that conclusions based on bookmark-based evaluation should better be confirmed by an independent evaluation method.

In this chapter we mostly focused on document retrieval, while abandoning other retrievable entities such as people, tags, and groups. Our user study evaluated people search quality, and indeed showed the superiority of Familiarity network over other networks for personalized people search. However, this results should be confirmed as many retrieved people were not judged due to participants' unfamiliarity (20%). In addition, we assume that familiar people were favored by participants in their judgments over non-familiar ones. Therefore, reducing that bias is needed in order to objectively evaluate personalized people search. However, we believe that people search will benefit from emphasizing familiar people for the searcher, as these people are the most reasonable sources of additional information, as expected from people search results.

As previous work showed, not all queries should be personalized [Teevan *et al.*, 2008]. We hypothesize that this is also true for personalized social search. Here we simulated personal queries with tags used for bookmarking by the user, in the off-line study, and with tags the user was tagged with, in the user study. In both cases these types of personal queries are limited and do not cover the whole spectrum of possible personal queries, but rather a subset that is likely to benefit from personalization and which can be judged by the methods in use.

Our personalization approach is simple and is based on re-ranking the search results based on their relationship strength with the user's related people and topics. The high effectiveness of this approach for social search implies that the social relations used for personalization, as derived from the user's social network, are highly reliable in predicting user interests and preferences. This claim holds for enterprise social data, as shown in this work.

We would like to point out the difference between desktop environment, which is a rich source of contextual links, and a enterprise area, in which we have limited context information. This discrepancy biased our research in corporate search area towards social connections. Once the enterprise contextual data become available for analysis, it would be interesting to characterize its value for information retrieval tasks.

In this chapter we investigated how the high quality social data that is available today for individuals in the enterprise, allows the identification of social relations that can be utilized for search personalization and for other applications. The question whether social data out of the firewall, typically with lower quality, can be used effectively for social search enhancement, would be considered in the next chapter.

5 Social Links in Social Networks on the Web

In the prevous chapter we have shown how social links could be used for personalized search on an enterprise level. But it does not accurately represent the large scale of the Web, in which the volume of data is larger and grows faster compared to enterprise networks. Given the rising popularity of social networks on the Web, it is interesting to investigate the potential of social links with respect to information retrieval tasks within such platforms. In this chapter we consider applications based on social links for large public social networks including Flickr [1] and Twitter [2].

5.1 Information Retrieval Tasks in Large Social Networks

Modern social networks share many common features like tagging, rating and commenting capabilities, user groups functionality, etc. However, each of them has its special focus. For example, Flickr is one of the first large-scale social networks for photo sharing. It allows its users to upload photos and describe them with tags, title and additional comments. Its users create social groups to share photos related to particular topics, or build friendship networks to exchange recent pictures with friends. Similarly, Youtube [3] is a famous place for sharing video content. It is also equipped with social functionality, where the videos can be rated and commented by all users. Twitter is one of the most well-known microblogging network at the moment for information exchange. People post there short text messages to personal message streams (microblogs), including URLs, to display their current thoughts, comment on news or disseminate information. Its users can tag each of their posts with keywords and all posts with the same tag can be displayed. Most of these platforms contain

[1] www.flickr.com
[2] www.twitter.com
[3] www.youtube.com

dozens of millions of users and billions of information items. The social links in such platforms create a potential for a variety of applications, that would reduce user's workload and retrieve only relevant content.

One example of asearch task on Flickr is *landmark finding*. When a user plans to travel to a new destination she often uses different social services to look for photos of representative sights and landmarks near her point of interest. Such textual search on multimedia resources is using tags and descriptions assigned by other people. The problem is that users in practice often do not know in advance what are the main sights in the city of interest and cannot formulate precise search queries. They could only provide a city name, which is not enough to filter out thousands of irrelevant photos tagged with such a name. Therefore, a search system should be equipped with a service, which helps to decide which landmarks photos to return for a given city name. The task of automatic selection of representative landmarks we call the *landmark finding problem* [Abbasi *et al.*, 2009b].

Some working prototypes for landmark finding are available online, for example, in the World Explorer [Ahern *et al.*, 2007] application [4] a user can enter a location name and browse through the landmark related tags and the corresponding photos. The system has a reasonable performance, but it only works with geo-tagged photos (supplied with geographical coordinates). The problem is that many interesting places around the world are still represented by photos without geo-tags and their landmarks cannot be found using World Explorer. Instead of using geo-tags, we propose to exploit the tagging features and social Flickr groups to train a classifier with minimum efforts.

Another related problem is that the photos alone do not provide a complete overview. A traveller might need some background information regarding the landmarks, video guides, recommendations from fellow travellers, etc. Since this information is spread across different social platforms, one has to search for landmarks on each service separately. Here we present a combined solution to this problem, a new mobile application GuideMe!, that retrieves landmark-related resources from various Web 2.0 platforms [Zerr *et al.*, 2011].

A second focus of this chapter is on *links recommendation* in Twitter. Besides search, users might be also interested in a recommendation functionality, especially for microblogging platforms. For example, there are plenty of different activities popular among Twitter users, including chatting, posting random thoughts and information sharing [Naaman *et al.*, 2010]. Recent studies confirm the eagerness of Twitter users to share links by finding URLs in 22% of Twitter messages [Boyd *et al.*, 2010]. However, the consumption of these recommendations by users is not yet well studied. At the same time, the risk of severe information overload of Twitter users increases with every new message stream (i.e. a *friend*) that they desire to read. The automatic real-time ranking of recommendations posted in these streams daily and hourly should

[4] http://tagmaps.research.yahoo.com/worldexplorer.php

provide a great help for Twitter users. We study the user preferences on URLs that were explicitly or implicitly recommended to Twitter users by their information providers. We analyze cases when users clearly express their interest in a certain URL by posting it to their own message streams, considering that the same URL was posted earlier by their friends within a short time window. The presented approach allows to recommend interesting link using information about user social interactions with other users.

The rest of this chapter is structured as follows: in Section 5.2 we review related work on tags analysis, landmark finding and links recommendation. Next, in Section 5.3 we introduce our algorithm for landmark finding. We present the mobile application, which uses our algorithm, in Section 5.4. Two methods for the real-time links recommendation on Twitter we discuss in Section 5.5. Finally, a discussion on obtained results and presented methods is included in Section 5.6.

5.2 Relevant Background

5.2.1 Landmark Finding

While Flickr photo sharing service becomes so popular, a lot of research has been done recently on this data. We distinguish few research directions and in particular, summary extraction and view representation for locations. The available algorithms used both purely content-based techniques and methods combining contextual information and content of the photos.

In one of the first works [Jaffe *et al.*, 2006], a three-steps approach has been proposed for creating photo summaries. Within the first step, authors partition geo-tagged photos into a hierarchy of clusters and later each cluster receives a score in the second step. In the third step, a ranking of all photos in the data is produced, using recursive ranking of the sub-clusters at each level, from the leaf clusters and to the root. The clustering is a fixed one-time computation step, but the ranking can be re-evaluated, allowing users to specify personal preferences towards social, temporal, spatial or other available features. Later, this clustering algorithm was changed to the K-Means algorithm [Kennedy *et al.*, 2007] and analysis of image visual features analysis has been added. The additional step with extraction of image color, texture and interest points allowed to select photos of the same landmark from different positions and the perceived quality of photo summaries has been improved.

A method developed in [Kennedy and Naaman, 2008] follows a similar combination of context- and content-based features. There landmarks are identified with analysis of the distribution patterns of the tags in the dataset. The representative photos for landmarks are found using canonical views. Using various image processing methods, the landmark images are clustered into visually similar groups, and are linked to each other if they contain the same landmark. The most representative pictures for each of these views are selected based on a link structure. The pure content-based method for iconic image ranking can be found in [Berg and Forsyth, 2007]. Photos labeled with a particular theme should be ranked with respect to a visual category. This algorithm consists of a learning step to locate the main subject inside the pictures, which is used later to classify the remaining photos. These test images are ranked according to shape and appearance similarity of 5 hand-labeled images per category. Three ranking algorithms are compared: random ranking inside categories, ranking using similarity over the whole image, and ranking using similarity of the segmented objects from the pictures. A user study shows that the superiority of the ranking with segmentation.

Some experts in image annotation examine the synergy of location information with image based media in [Toyama *et al.*, 2003]. They propose solutions for acquiring the location metadata. A list of six methods for gathering location tags for image media is proposed: (1) by manual entry, (2) from the camera, (3) from a separate location-aware device, (4) from a digital calendar, (5) from the surrounding text and (6) by association with other digital documents with known location tags. Complementary to [Toyama *et al.*, 2003], in al. [Davis *et al.*, 2004] it is suggested to enhance photos with metadata. They consider activities, objects, people and location for the metadata sources.

The ontology-creation from Flickr tags is a topic of several recent articles. These methods infer semantics of tags [Abbasi *et al.*, 2007; Schmitz, 2006] and time usage distributions [Rattenbury *et al.*,]. Following ideas from a burst detection algorithm, the authors apply a family of Spatial Scan methods for attribution of tag semantics and achieve relatively good recall and precision rates. However, compared to our approach this work relies only on photos with geotags. With the idea to organize Flickr photos, another approach [Naaman *et al.*,] develops the PhotoCompass system. It uses both location and time data to build a collection of personal photos. The photos are structured into event- and location-based hierarchies, in which location names are used to label obtained clusters. By comparing the pictures' geospatial coordinates against external geographical datasets a set of state, city or park names is created for labeling each pair latitude and longitude values. Next, several heuristics are imployed to select between one and three terms for each cluster caption. This is similar to our work in the sense that they also use external sources of information for inferring location names. Later, the same research group presented the World Explorer application [Ahern *et al.*, 2007], in which authors also create summaries of sights by first clustering images based on geographic location information and then score tag representativeness of each tag in the cluster.

We consider the similar problem of generating a summary of landmarks, but in absense of geo-spatial information. It is reported in [Ahern *et al.*, 2007] that some regions like San Francisco had enough geo-tagged photos. However, many other regions had sparse geo-tagged data. While the majority of pictures do not have manually specified geographic location, in Section 5.3 we propose a landmark finding algorithm based solely on plain tag data from Flickr, enriched with social links information like Flickr groups.

5.2.2 Recommendations in Social Systems

The recent work presented [Chen *et al.*, 2010] studies content recommendation on Twitter. For recommendations they investigated three dimensions: topic interest models, content sources and social voting. The combination of these dimensions with different parameters has been evaluated in 12 versions. Real Twitter users evaluated the system and the most effective algorithm doubled the percentage of interesting content compared to a baseline.

A tag recommendation strategy is described in [Sigurbjörnsson and van Zwol, 2008a]. First, the authors analyse a representative snapshot of Flickr and present a characterisation of tags. Based on this analysis, several tag recommendation strategies are evaluated. They aim to help the user during the photo annotation by recommending tags that can be added to the photo. The empirical evaluation results demonstrate that algorithm is reasonably effective. Another type of items for recommendation is news stories. In [Phelan *et al.*, 2009] it is shown an approach to news recommendation that uses Twitter activity for ranking news stories from a user's RSS feeds. A preliminary evaluation shows effectiveness of this method.

In [Amatriain *et al.*, 2009] an algorithm for recommending items to users is using expert opinions. For a nearest neighbors algorithm they limit a set of considered neighbors to a set of expert neighbors from an independent dataset, whose opinions are weighted according to their similarity to the user. The approach is evaluated using the Netflix dataset and a user-study with positive results. As our ideas are close to time-critical collaborative filtering, we considered work presented in [Ding *et al.*, 2006]. The authors suggested that users' recent ratings reflect her preferences more than older ratings. To reflect time-based topic drift, they propose a recency-based collaborative filtering algorithm, which weights items based on their expected accuracy on the future preferences. Its new similarity function produces more accurate similarity scores and prediction precision is significantly improved.

The influence of social connections on users behaviour has been studied in [Anagnostopoulos *et al.*, 2008]. This question is important in a sense that modern recommender algorithms imply that user's behaviour is in a way similar to behavior of her "friends' in a social network. Their evaluation on Flickr data suggests that while there is significant social correlation in tagging behavior on this system, this correlation cannot be attributed to social influence. When thinking of social links we imply some similarity between users with tight connections. Some earlier works considered similar users for collaborative filtering. In [Wang *et al.*, 2006] the memory-based collaborative filtering problem is formulated as a generative probabilistic framework. The novelty of this approach is that recommendations are based on similar users who rate similar items. This model is more robust to data sparsity give better recommendations. In another paper [Bell *et al.*, 2007] authors present algorithm for predicting user ratings of items. They take into account user's interaction in the neighborhood, resulting in improved estimation quality. The experiments on the Netflix dataset demonstrate the effectiveness of this method. But microblogging is different in terms of user behaviour and available data, so these approaches need to be adopted for the Twitter.

While recommender systems achieved good results in many domains, the problem of recommending links on Twitter has not been fully addressed so far. The most relevant prototype [Chen *et al.*, 2010] does not make a particular stress on recommending links and social dimension is presented in a simplest form of social voting. In Section 5.5, we present our algorithms for link recommendation and show their effectiveness with experiments on the Twitter data.

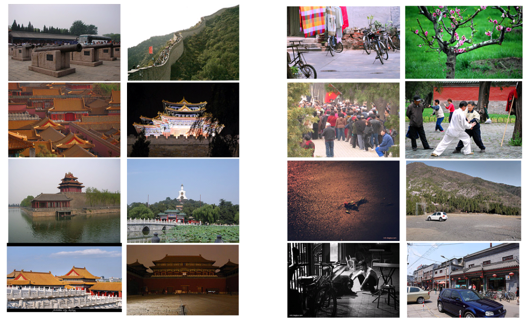
5.3 An Approach for Finding Landmarks

As digital photography and social photo-sharing services continue to grow, the tasks of effective photo search is getting more and more attention from research. For example, Flickr is already beyond simple photo sharing and it becomes much more like a social network, particularly for people who like traveling. The millions of photos from all over the world can help users in picking the most interesting places to visit. However, it becomes difficult to get a comprehensive overview of a city having also hundreds of family photos and party pictures mixed all together in the same picture sets.

Consider a scenario, in which a user is thinking about going to Beijing and searches for Flickr photos tagged with "Beijing". She wants to see a concise and representative view of Beijing with a few photos related to both famous and relatively unknown landmarks, like in Figure 5.1(a). Instead, she receives many non-relevant photos as the ones presented on Figure. 5.1(b). These photos contain the tag "Beijing", but do not give a good overview of the local landmarks any tourist should see. To address this scenario we propose a method for automatic landmark selection for each location of interest. We refer to this task as the *landmark finding* problem.

Some of the previous work focusing on creating landmark photo summaries from Flickr data [Rattenbury *et al.*, ; Jaffe *et al.*, 2006; Kennedy and Naaman, 2008; Ahern *et al.*, 2007], rely on geo-tagged pictures for solving this problem, but the majority of Flickr photos still miss geographical coordinates data. The usage of simple text labels or tags for photo annotation is much more widespread among users. Therefore we choose to use these annotations for helping us in solving this task. We apply a Support Vector Machine (SVM) binary classifier to identify landmark-related photos. Getting better classification results requires good training data. We solve the problem of training the classifier by exploiting the feature of social groups available on Flickr [5]. These groups mostly contain images which are related to the group

[5]http://www.flickr.com/groups



(a) Landmark photos **(b) Non-Landmark photos**

Figure 5.1 Landmark (a) and Non-Landmark (b) photos containing tag “Beijing”.

theme. We found that using information available in Flickr groups provides very good training data for the classifiers. After classifying the images as landmarks, we extract and rank the representative tags from landmark images and use these tags to create city sights summaries.

The landmark finding problem is a hard task in two ways. The content-based image analysis has very limited capabilities to solve this problem in general, given that photos are taken in different light and weather conditions, from different viewpoints and angles. The text-based or tag-based methods are much more appropriate for this task, but they do not have extra information if a tag represents a landmark or family photo taken in a city. We propose to obtain this extra information from social groups in which users are involved in. Nowadays Flickr is enriched with specific photo groups related to landmarks, cars and other types of objects and themes, which can be used to distinguish the main topic of the photo. To our best knowledge, our solution is a first method which solves landmark finding problem based on photo communities information.

Within our approach we identify two main parts. First, we exploit tags and social Flickr groups to train a classifier to identify landmark photos and tags. The method requires minimum human intervention, one only has to give links to relevant Flickr groups and the system automatically trains a classifier based on the data retrieved from Flickr groups. Second part of the method ranks all suggested relevant tags by their representativeness of a landmark. It is important that our approach is generalizable for other problems like car finding, mobile phone finding, etc. Current method is limited to users’ tags and social Flickr groups and does not make use of low level image features or geo-tags. In our user study, we show that our approach outperforms World Explorer even on geo-tagged photos.

5.3.1 Formal Model

The goal of our landmark finding application has to automatically create a summary of Flickr photos, giving a comprehensive overview of landmarks at some location of interest. We will decompose this task into several sub-problems, as presented in Figure 5.2.

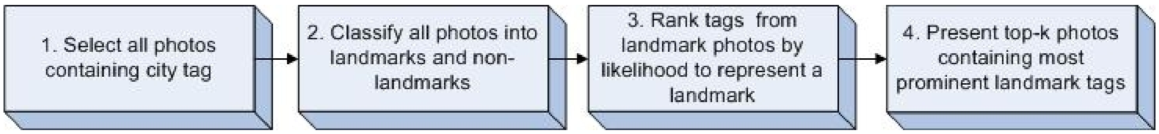


Figure 5.2 Decomposition of Landmark Finding Problem

The first step consists of selecting a set of photos related to a particular city. Since we do not consider geo-tagged photos, we rely on the simple heuristic of having the city name as a tag associated with a photo. This way we may miss some relevant photos, but for our task it is not a problem, since we still get a lot more photos than we need for a summary generation. In the second step, all collected photos are automatically classified as either landmarks or non-landmarks. It is important to understand that at this point, we do not have a summary of city landmarks. We have just a list of pictures with or without landmarks, according to the classifier. What we want to achieve is a list of names representing city landmarks and based on these names create a comprehensive city landmark summary. In the third step, tags of the photos classified as landmarks are ranked according to their likelihood of representing city sights. Once a ranking score is available for all tags in the set, in the fourth step we select top-k most representative tags. For each of these k tags we retrieve the best Flickr photo which has as tags both the name of the city, as well as the landmark tag. We consider that photos should not contain several representative tags at once, since we aim at showing a single landmark with each photo.

For returning the set of top-k Flickr pictures satisfying the conditions described above we make use of the Flickr API[6] for tag-based search and sort the pictures by relevance. As the steps 1 and 4 presented in Figure 5.2 are quite simple, we will not discuss them further. Rather we would focus on the sub-problems of classification and tag ranking.

Before diving into the details of the landmark finding problem and understanding the algorithms we need to introduce a number of formalizations and definitions. We define the collaborative tagging system S of users, tags, and resources, and relationship between users, tags, and resources as a quadruple

$$S = (U, T, R, Y) \tag{5.1}$$

where U represents the set of users, T represents the set of tags, R represents the set of resources and $Y \subseteq U \times T \times R$ is ternary relation over U, T and R, if a user $u \in U$

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

uses tag $t \in T$ to tag a resource $r \in R$, then there is a relation $(u, t, r) \in Y$. This assignment is called tag assignment [Hotho *et al.*, 2006a].

Tag Frequency (TF)

In information retrieval, normalization techniques like Term Frequency Normalization have been used. Similarly, like Term Frequency normalization is used to prevent bias towards longer documents, we use the same idea to prevent bias of results towards resources having a lot of tags. We define the number of times that a tag t appears with a resource r as frequency of the tag t with resource r, $f_r(t)$:

$$f_r(t) = \mid \{(u, t, r) \in Y, u \in U\} \mid \tag{5.2}$$

In Flickr, as a *Narrow Folksonomy* system, a resource can be tagged with a keyword only once [Wal, 2005] and because of this restriction, the function $f_r(t)$ will always return either 1 or 0. In order to reduce the effect of photos having a large number of tags, we normalize the frequencies of tags by dividing the number of occurrences of a tag in a resource by the total number of tags assigned to that resource. Normalized tag frequency $TF_r(t)$ of a tag t in a resource r is thus computed as follows:

$$TF_r(t) = \frac{f_r(t)}{\sum f_r(t')}, (u, t, r) \in Y, (u, t', r) \in Y, t' \in T, u \in U, \tag{5.3}$$

Inverse Resource Frequency (IRF)

Inverse Resource Frequency (like Inverse Document Frequency in information retrieval) is used to reduce the influence of very popular tags like "sky" or "wedding". *IRF* of a tag t is computed by dividing the total number of resources by the number of resources that have tag t and taking its *log* for smoothing:

$$IRF(t) = log\left(\frac{\mid R \mid}{\mid \{(t, r), u \in U, r \in R, (u, t, r) \in Y\} \mid}\right) \tag{5.4}$$

Inverse User Frequency (IUF)

Similar to *IRF*, we define Inverse User Frequency (IUF). *IUF* identifies the general importance of the tag based on the number of users that used that tag. If a tag is used by many users it has a low *IUF* value and respectively if it is used by few users, it has high *IUF* value. We formally define *IUF* as follows:

$$IUF(t) = log\left(\frac{\mid U \mid}{\mid \{(u, t), u \in U, r \in R, (u, t, r) \in Y\} \mid}\right) \tag{5.5}$$

5.3.2 Our Algorithm

Step 1: Selection of City Photos

The first step is to select a set of photos related to a particular city. Since we do not consider GPS-enriched photos, we rely on a simple heuristic of having city tag name associated with a photo. This trick works well in practice, since city names are commonly assigned by Flickr users, even if more detailed tags are not provided. Another possibility would be to pick a social Flickr group with a city name and consider its content as relevant.

Step 2: Landmark Classification

From the set of pictures containing a city tag, we want to select photos representing landmarks. For this task, we make use of a Support Vector Machine (SVM) binary classifier [Vapnik, 1999]. SVM is a state-of-the-art method for text classification, so its performance is expected to be comparable with other machine learning algorithms. In general it is possible to apply any other classifier, and in next sections we will show results obtained with a Naive Bayes classification algorithm. Here we make use of SVMLight implementation [Joachims, 2002].

Normalization

For every picture, we create a feature vector based on the tags which were used to annotate it and the SVM classifier assigns each photo to "landmark" or "non-landmark" categories. We assign weights to the tags in the feature vectors based on the usage of tags among resources and users, presented below. Formally we define a feature vector for a photo r as following:

$$F(r) = [w(t_1, r), w(t_2, r), \cdots, w(t_{|T|}, r)], u \in U, t \in T \tag{5.6}$$

where $w(t, r)$ can be defined using one of the following normalization methods:

$$tr(t, r) = \begin{cases} 1 \text{ if } (u, t, r) \in Y, u \in U; \\ 0 \text{ otherwise.} \end{cases} \tag{5.7}$$

$$tf(t, r) = TF_r(t) \tag{5.8}$$

$$tf\_irf(t, r) = TF_r(t) \cdot IRF(t) \tag{5.9}$$

$$tf\_iuf(t, r) = TF_r(t) \cdot IUF(t) \tag{5.10}$$

$$tf\_irf\_iuf(t, r) = TF_r(t) \cdot IRF(t) \cdot IUF(t) \tag{5.11}$$

For example, if the tag set is $T = \{t_1, t_2, t_3, t_4\}$ and a photo r is tagged with tags t_1, t_3, t_4, then its simple feature vector without any normalization (eq. 5.7) is represented as $[1, 0, 1, 1]$, where 0 at position two represents that this photo has not been tagged with the tag t_2 and 1s represent that each of the tags t_1, t_3, t_4 has been used once to tag r.

The reason to use different normalization methods is to select the most appropriate weighting scheme for normalizing the features. Besides simple boolean weighting *tr* (Eq. 5.7), we consider *tf* normalization (Eq. 5.8), which gives lower feature weights to the training examples having a lot of tags. But *tf* normalization does not take into account the effect of number of users and resources having a particular tag. To boost tags with higher discriminative power, we assume that if a tag is used quite often among all the resources, it must be given a lower weight than the tags used infrequently across different resources. To incorporate the effect of the number of users and resources sharing a tag, we correct weight using number of resources having a particular tag (Eq. 5.9), number of users who assigned a tag (Eq. 5.10), and both (Eq. 5.11). In the next sections we show experiments with these normalization schemes for initializing the feature vectors corresponding to the pictures for which we then train the SVM binary classifier to separate them into landmarks and non-landmarks photos.

Training the Classifier

One of the main challenges for SVM or any machine learning technique is to create a good training set. Once a model is created based on the labeled data from the training set, the SVM can classify unseen examples based on the model. Our hypothesis is that some of the Flickr groups like "Landmarks around the world" can serve as positive examples, while arbitrary general groups, like "Birds" or "Airplanes" represent negative examples. Both positive and negative example sets are used for training the SVM and the learned model is used to classify new photos. For each feature vector corresponding to a photo, the SVM classifier returns a decision value. Positive decision value corresponds to "landmark" and negative value to "non-landmark". These decision values returned by the classifier are then re-used for ranking tags in next step.

Usage of the Flickr groups as training data can be extended for any arbitrary photo classification task beyond the landmark finding problem. If a relevant group of photos exists on Flickr, one can use it as a training data to find more photos on the same topic within Flickr. For example "CAR [directory]" or "Mobile Phones" groups can be helpful for finding thousands of car and mobile photos.

Step 3: Measuring the Representativeness of Tags

Once we have selected a set of city photos and filtered only landmark-related ones, the third step consists of ranking all tags by how well they represent landmarks. What we would like to achieve is a ranked set of tags representing landmarks specific to a particular city. For example, one can intuitively mark the tag "sky" as a poor evidence of landmark, "bridge" is somehow better and "goldenbridge" is the most promising one. However, we need to be able to generalize this over the whole set of Flickr tags for finding the most probable tags as being landmark annotations. Several intuitions for discovering the most representative tags were presented in [Ahern *et al.*, 2007]. We consider the global and local tag properties. Global properties consider the complete dataset, while local properties are related to the tags representing landmarks of a particular city.

The global properties are seen when looking at the whole dataset. For example, we would like to give low score to common tags. The assumption is that representative landmark tags appear few times in landmark photos, but not very common among rest of the collection. Let us consider R the set of all photos (both landmark and non-landmark related ones), and T the associated set of tags. Supporting this first assumption, we compute the *Inverse Resource Frequency* (Eq. 5.4) of the considered tag. If a tag is frequently used to tag photos in the dataset, it has a low $IRF_{R,T}(t)$[7] value and vice versa. Similarly, if a tag is globally very common amongst users, it must be scored low. This is achieved by computing *Inverse User Frequency*, $IUF_{R,T}(t)$ (Eq. 5.5).

After defining global scoring factors, we come to local measures computed on part of the collection with landmark photos only. When considering the dataset containing only pictures associated to a particular city and classified as landmarks, our assumption is that common tags should be scored high. Let us represent the set of landmark-related photos selected for a city as R_c and the corresponding tag set as T_c. If a tag is common among the photos for a particular city, probably this tag represents some feature of the city, e.g. some museum, or an old and famous building. Let $nrt_c(t)$ be a number of times a tag t appears within landmark photos for a city c. Then we can compute normalized *City Tag Frequency*, $CTF(t)$, as follows (Eq.5.12):

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

Similarly, if a tag is used frequently by users, then it is probably a feature of the city. Let $nut_c(t)$ be the number of users using a tag t for the landmark photos for a city c. We compute the normalized *City User Tag Frequency*, $CUTF$, using (Eq.5.13):

$$CUTF(t) = \frac{nut_c(t)}{MAX(nut_c(t'))}, t, t' \in T_c \tag{5.13}$$

However, it is often the case that a frequent tag for a city represents a different name of the city, or the country in which the city lies. Therefore we ignore those tags which are found in more than 20% of the landmark pictures corresponding to a city.

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

[7] Computation is relative to R and T

tag t.

$$CONF(t) = log\left(\sum_{r \in R_t} d_r\right) \tag{5.14}$$

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

$$SCORE(t) = IRF_{R,T}(t) \cdot IUF_{R,T}(t) \cdot CTF(t) \cdot CUTF(t) \cdot CONF(t), t \in T_c \tag{5.15}$$

Step 4: Ranking Photos for Landmarks Summary

Once we have for a city a set of tags, ranked based on their representativeness scores, we need to retrieve the most relevant photos depicting these landmark tags. From the previous step, for a city c we have top-k tags, $TK_c = \{t_{1_c}, ..., t_{k_c}\}$, scored based on how well they represent the city's landmarks (by using the formula presented in Equation 5.15). In the last step, for each of these k tags we retrieve the the best Flickr photo which has as tags both the name of the city as well as the landmark tag. We consider that photos should not contain several representative tags at once, since we aim at showing a single landmark with each photo. When several photos contain the tag we randomly choose one out of this set.

For returning the best Flickr pictures satisfying the conditions described above we make use of the Java Flickr API[8] and use the sorting by 'relevance' Flickr scheme for ranking the pictures. We do not use the text search feature, as in Flickr standard photo search is based on Title, Description, Notes and Tags. We only rely on tag-based search.

5.3.3 Experimental Datasets

In the following we present the datasets used in our experiments: a ground truth set of pictures (DS_1), a training (DS_2) and a test (DS_3) dataset. We proceed by discussing each of them in detail.

Ground Truth Data (DS_1).

The ground truth dataset consists of 640 pictures, namely 200 images that depict city landmarks (like historical buildings, monuments, and parks etc.) and 440 non-landmarks images (people, flowers, animals, and objects, etc.). Images representing landmarks were collected manually from different Flickr groups related to travel and landmarks, while all of the non-landmark images were selected from general groups. Figures 5.3 and 5.4 show a sample of landmark- and respectively non-landmark images.

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

Figure 5.3 Sample ground truth data showing landmarks

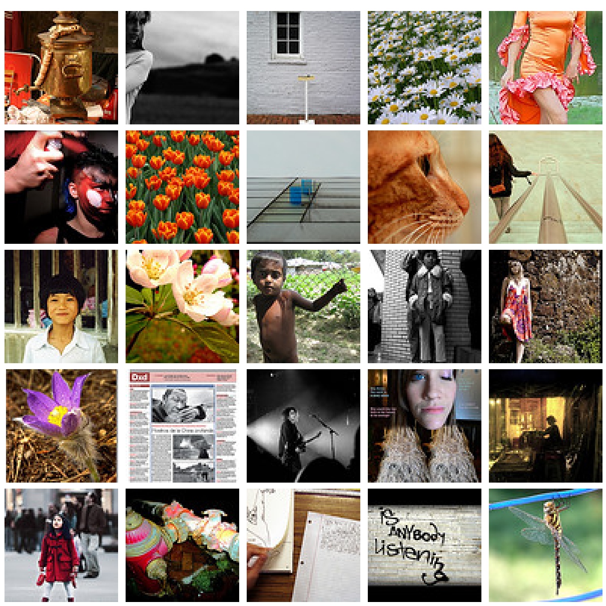


Figure 5.4 Sample ground truth data showing non-landmarks

This dataset was used to find an optimal weighting scheme from Equations (5.7)-(5.11) to estimate the feature vector defined in Equation (5.6). Detailed results of this analysis are presented in Section 5.3.4.

Training Data (DS_2).

To train the classifier we downloaded 430282 photos uploaded by 57581 different users to the Flickr groups, several examples of these groups are listed in Table 5.1. This dataset contained 14729 positive examples (related to landmark groups) and 415553 negative examples (related to general groups). None of these 430282 photos were used for the test data set. All cities used for evaluation are diversely represented in the training data. For examples, the cities Leeds, Yokohama, and Mexico City do not contain any positive example in the training data. For 15 other cities, training data contains less than 8 positive training examples per city.

Negative Groups:	Positive Groups:
Airplanes (15529695@N00)	Landmarks (13772150@N00)
Birds (10469051@N00)	Landmarks (16906560@N00)
Cars (16305888@N00)	Landmarks (45202403@N00)
Boats (393574@N22)	Landmarks (54505019@N00)

Table 5.1 Examples of Flickr groups used for training the classifier. Each group can be accessed using URL like http://www.flickr.com/groups/13772150@N00.

Test Data (DS_3).

Finally, the third dataset consists of pictures corresponding to 50 randomly picked cities (for which World Explorer [Ahern *et al.*, 2007] has at least 10 landmark tags), 60% European ones and the rest of 40% representing Asian, North-, South- American and Australian cities. We downloaded 4,000 to 5,000 photos/city, so that in total we gathered 232,265 photos, uploaded by 32,409 different users. Pictures from dataset DS_3 were used for testing the classifier, after a model was learned based on DS_2. None of the photos in DS_3 was used for training the classifier.

5.3.4 Experiments

Tags' Influence on Classification Accuracy

With the first set of experiments we focused on identifying how different tags influence the accuracy of the classification algorithm. We used the SVM implementation proposed in [Joachims, 2006], as this implementation uses a cutting plane algorithm for training the classifier in $O(|T||R|)$ for linear classification. Thus, this allows us to use SVM on a large scale.

For our exploration, we used the ground truth dataset DS_1 and we applied the SVM classifier on this data using different percentages of data as training and test sets.

Tag Counts	Frequent Tags	User Counts	User-Popular Tags
15	architecture	14	architecture
14	travel	11	travel
5	asia	5	asia
5	digitaltribes	4	monument
5	dt	3	ancient
5	markoneil	3	church
4	church	3	clouds
4	india	3	goldstaraward
4	marble	3	heritage
4	monument	3	india
3	2004	3	marble
3	agra	3	masjid
3	ancient	3	mosque
3	baroque	3	night
3	boston	3	sky

Table 5.2 Top-15 tags from correctly classified landmarks, ordered by number of occurrences (left) and ordered by number of users (right)

As expected, the accuracy of the classifier increases as more data is used in training and the best results we obtain when proportion of positive and negative landmark examples are of comparable size. The first question was if it is useful to normalize tags frequency with a number of photographers who used them. We considered the result from SVM classification in which 50% of the positive and negative data was used for training and the other half for testing. The classification accuracy was about 84%.

For the correctly classified set of landmark examples, we collect the tags within this set, ordering them by plain number of occurrences and taking top-15 most frequent tags as the most representative tags for landmarks. Similarly, we also build a second set of 15 top tags scored by the number of users who applied them at least once. Table 5.2 presents the resulted top-15 tags, on the left side the most frequent tags and on the right side tags used by most number of users. One can see that less meaningful tags like "dt" or "2004" appearing in the left column are filtered out in the right column, which suggests that tag ordering by number of users provides a better evidence for landmarks compared to plain tag frequency. The other examples we explored also support this observation.

In recent research [Sigurbjörnsson and van Zwol, 2008b] it was found that different classes of tags have different utility for the users. For example, location tags are considered two times more useful than tags referring to persons. We were interested to see which groups of tags have positive and negative influence on the classification accuracy. Thus, based on visual exploration of classification results under different parameters we qualitatively studied the influence of different groups of tags:

Neutral tags. We observed that color-related tags like "blue", "green", "yellow", etc. do not boost classification. The tags containing digits also of limited usefulness in landmark classification, they are quite popular among both positive and negative examples. The number of tags per photo is also a weak classification feature, both negative and positive examples contain photos with small and large number of tags. The language of tags also does not have significant influence on classification.

Positive tags. Country names and religion-related tags like "buddhisttemple", "hinduism" or "church" usually represent a strong evidence for a landmark photo. Obviously, architecture tags like "pyramid", "ruin" or "castle" appear a lot more often in landmark photos.

Negative tags. Person tags, like "me", "girl" or "people" are good indicators of non-landmark photos. Among other frequent non-landmark tags we found tag groups like clothes, "jeans" or "kilt", and body parts, "eyes" or "beard".

Given the above mentioned findings it would be useful for future work to first pre-process the sets of tags occurring along with the pictures we aim to classify as landmark or non-landmarks, so that only those types of tags with positive and negative influence are considered. For training the classifier with positive examples we would increase the weights of tags coming from the positive set of annotations, whereas for the negative examples we would bias tags identified as negative ones.

Small Scale Evaluation for Selecting Normalization Schema for Feature Vectors

In the present section we proceed by presenting the evaluation of this approach. We are interested in how effective we can classify photos into landmarks and non-landmarks pictures (step 2). To select the best normalization schema out of (Eq.5.7-5.11), we evaluate their accuracy on the small dataset DS_1. Having all pictures in this set manually labeled as landmarks or non-landmarks, it is then straightforward to assess the accuracy of the classifier using each of the schemes. The SVM was trained on a subset of the ground truth dataset DS_1, using 80% as training data and rest as test data. Note that all normalizations are computed relatively to the collection DS_1.

In Figure 5.5 for the five normalization schemes we present Precision, Recall and F-measure values. Given the information overload which users nowadays have to face, we do not consider Recall a very important parameter for our algorithms. We are rather interested in improving the Precision values – i.e. providing users only relevant results.

All normalization methods have precision values above 80%: simple *tr* performes worst with 85.53%, followed by *tf* with 88.61%. The best performing schemes are *tf_irf* and *tf_iuf* with 92.65% and 92.54% respectively. Interestingly, the influence of users's tag usage (see Equation 5.5) is lower than that of the resources tagged with a particular tag. We can observe on the plot that normalization by *iuf* or *irf* increases precision, which supports our earlier observation from Section 5.2. However, combination of both factors does not improve classification further. We think that the normalization is effective only until a reasonable limit, while the current dataset is too small to make strong quantitative comparison. Therefore for the large scale evaluation we will focus on the *tf_irf* normalization scheme.

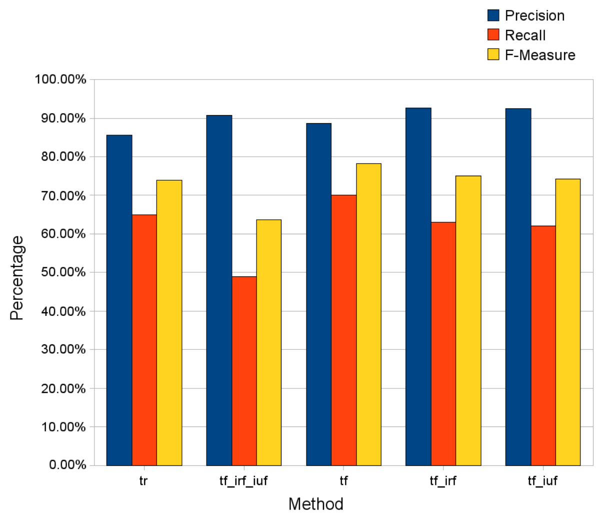


Figure 5.5 Classification results on ground truth data (DS_1) using different normalization methods

Large Scale Evaluation of Landmark Finding Accuracy

Here we test how effective our algorithm performs overall for creating summaries of city landmarks (outcome after step 4). For being able to generalize the good classification results presented in Section 5.3.4 we need to experiment also on bigger datasets. Therefore we focused on measuring the classification accuracy in the following setup: the SVM classifier was trained on the dataset DS_2 (see Section 5.3.3). After the model was learned, we tested the classifier on the dataset DS_3 containing in total 232265 pictures corresponding to 50 cities. The evaluation of this classification was then done through a user survey. We present the details of this analysis below.

Tag- vs. Geo-Tag-Based Classification

Our final goal is to evaluate the quality of the city landmarks we identify with our proposed algorithms. For this, we want to compare our results against results produced by existing systems trying to solve the same problem. For the purpose of this evaluation we chose World Explorer to compare against and since this system uses as input for its algorithms Flickr pictures with GPS data – i.e. richer input data than in our case – our aim is to obtain at least comparable quality. In the following we describe in detail the setup and methodology we used for this evaluation and the results we obtained.

As already mentioned, for evaluating the performance of our algorithms, we chose to compare our results against results returned by World Explorer. Our goal was to evaluate the landmark summaries created for a list of 50 cities (the same cities used to create the testing dataset DS_3 presented in Section 5.3.3). Most of the cities are European, since we expect our users to be more familiar with them.

Since World Explorer needs as input geographical information instead of city names, for all 50 cities we collected their associated GPS coordinates from the World Gazetteer database[9]. For retrieving tags representing city landmarks, we then made use of the World Explorer's Web API[10]. For each city, one needs to specify two pairs of GPS coordinates defining a rectangular area inside the city (bottom - left and top - down corners), for which the World Explorer web service should return the corresponding landmark-related tags. While World Explorer has 16 different zoom levels, we concentrate only on the single city-level zoom. The most relevant level choice is an interesting question, but for now we consider it to be level 5, which is mentioned as "city" zoom level in [Ahern *et al.*, 2007].

Given the fact that for the cities we have selected, we only had one single pair of GPS coordinates - basically specifying the city center - we had to define two additional pairs, such that the resulting rectangle had its' center coinciding with the city's. We assumed that most of the city landmarks are located around the city center. We experimented with two sizes of the rectangles' sides: 10x10 km or 5x5 km. However, the case where the rectangle sides were 10 km produced the best results.

Based on our problem statement we would like to evaluate each of four stages of data processing and overall effectiveness of our approach. On the first stage we select a set of city-related photos with simple heuristic that photo must have a city name tag associated to it. We observed that Flickr system provides us with thousands of photos for each city tag name, which mostly do not represent a landmark. The user evaluation details for other three steps we present below.

Evaluation Setup

Based on the initial results obtained on the datasets DS_1 we chose for this evaluation the best normalization scheme for the tags, *tf_irf*. This was the most robust and effective for the classification of results. Thus, the tag representativeness was measured according to Eq. 5.15 and this combination of parameters we will call below "TG-SVM method" (Tag - Group - SVM Method).

For the comparison between photo summaries produced from World Explorer tags and our algorithm we experiment on the set of 50 cities. For each of them World Explorer produced at least 10 tags and we discarded all tags above top-30 (in order to keep a reasonable set of pictures to be manually evaluated by users). For each tag belonging to a particular city we issued a tag query of the form "cityname tagname" using the Flickr API. From each query we picked the top-1 returned photo. If this photo was already returned by some other tag for the same city we picked the next top photo and so on, to avoid having duplicates in the photo summaries.

For the evaluation setup we recruited 20 volunteers among our colleagues, who are expert users and familiar with photo sharing and search services. Each user was asked to evaluate two result sets for 10 randomly selected cities out of 50, and the selection

[9] http://www.world-gazetteer.com

[10] http://developer.yahoo.com/yrb/tagmaps/

process picked each city so that by the end of the experiment it was evaluated by at least 4 users. Two photo summaries were combined from preview photo versions and mixed on a single screen, with one result set created using our algorithm and one coming from the World Explorer API. The users did not know which system produced each photo, as the photos from the two systems were randomly interleaved. Each photo was supplied with a title and a single landmark tag produced by either World Explorer or by our algorithm and used to retrieve this photo. A radio button was placed near each photo, where users selected between "landmark", "non-landmark", and "don't know" options. The users were asked to judge if a photo is a landmark or not, in total producing between 400 and 500 judgments per user. The experiment took about 30 minutes per user.

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

Evaluation Results

We observed quite different users' assessments patterns, some are considering lots of photos as landmarks and some accepting only few of them. First, we measured the performance of two algorithms for each city separately. Each city was assessed by 4 users, so we applied majority vote with "landmark" judgment is equal to 1, "non-landmark" to -1 and "don't know" is equal to 0. If sum of 4 judgments was greater than 0 we considered photo a landmark.

In Table 5.3 we present micro-average precision for each city, where our algorithm outperformed World Explorer on 30 out of 50 cities or in 60% of the cases. For evaluation we had to take only cities with sufficiently large number of geo-tagged photos to allow comparison with World Explorer. But our algorithm did not consider geo-spatial information, so it also works for cities for which geo-tagged photos are not available. The results show that we can effectively find landmarks in absence of geo-spatial information.

One problem with city-wise averaging is that each user had different bias in judgments. However, each user expressed the same bias towards both algorithms, so comparison of overall algorithm performance averaged across judgments of a single user is fair. In Table 5.4 we present the results from each user using macro-average precision, when all photos marked by users as landmarks are normalized by the total number of photos returned by an algorithm. Out of 20 users, 16 preferred our algorithm, 3 considered World Explorer-based results better and in one case the algorithms performed equally well. We obtained 12% improvement in precision with our method over World Explorer. We performed a paired t-test over the two outputs and calculated that precision improvement of our algorithm is statistically significant at confidence level $\alpha = 0.001$.

These results support our hypothesis that landmark finding based on photo classi-

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

Table 5.3 Micro-Average Precision for 50 Cities

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

Table 5.4 Macro-Average Precision for 20 Users

fication can replace geo-tagging based methods in situations where geo-spatial information is not available. They also show that our algorithm significantly outperforms state-of-art algorithms for landmark search. There was no particular tuning of the representativeness score as defined by (Eq. 5.15). Estimating the best combination of these parameters might give additional boost to results' quality.

Our approach for finding landmark could be easily combined with external content-based methods for retrieving sights of interest. We have shown that social links alone, in particular Flickr social groups, could be effectively used to improve user's search experience. Besides landmarks, one can apply this method for a similar search task, targeted on different type of entities. The only requirement is to have the relevant social groups available, which would provide a basis for a photo classification. Next, we present a practical application that uses our method for finding landmarks.

5.4 Mobile Application for Landmark Finding

While landmark finding in a Flickr setting definitely helps a traveller to see the locations of interest, it provides a limited information. A tourist might need some background information regarding the landmarks, video guides, recommendations from fellow travellers, etc. Fortunately, such information could be found on various Web 2.0 applications. These services are a rich source of multimedia resources and accompanying metadata, that describe sights, events, whether conditions, traffic situations and other relevant objects along users route. But this information is spread across different social platforms, one has to search for landmarks on each service separately.

Consider example, in which a user is thinking about going to Hanover and searching for resources tagged with "Hanover". She wants to see a concise and representative view of the city with a few photos and videos related to famous landmarks. She might also need a couple of web pages with a historical overview and travel tips. Such information need could be satisfied with results from different Web 2.0 services, with photos retrieved from Flickr, videos from Youtube and bookmarks from Delicious, like in Figure 5.6.

This scenario requires several searches on different social platforms, which might be particularly inconvenient if a user is already travelling and looking for the interesting sights using her mobile phone. It is also challenging to filter landmark-related information on each service, as this problem is currently addressed only by photo sharing providers. To address this issue we developed a mobile application GuideMe! [Zerr *et al.*, 2011] that retrieves landmark-related resources from various Web 2.0 platforms. First, GuideMe! extracts representative tags from landmark pictures according to the algorithm which we described in the Section5.3. The landmark-classified tags are used to query a set of social sources like Flickr, Youtube, Delicious, Slideshare, etc. Finally, the retrieved results are fused and ranked according to their relevance, popularity, rating and number of comments.

Figure 5.6 Landmark resources for "Hanover".

5.4.1 System Architecture and Mobile Client

This section describes the three main components of the GuideMe! system architecture presented in Figure 5.7. We present the GUI of the mobile client, followed by the landmark extractor web service and finally the InterWeb web service — a mashup network which integrates a number of tools like Flickr, YouTube, Slideshare, etc.

The GuideMe! application can be installed on any mobile device with the Android [11] operating system. The graphical user interface consists of the explorer page and the settings page, see Figure 5.8. The explorer page contains a search field and a list of results is displayed after successful search query execution. At the settings page the user can define how many resources should be displayed in the result list and select their preferred data types. The available data types are:

- Videos (from YouTube, Vimeo and Ipernity)
- Pictures (from Flickr and Ipernity)
- Presentations (from Slideshare)
- Bookmarks (Delicious)

Each result can be previewed at the original web page using the integrated browser. GuideMe! uses an internet connection over WLAN or GSM, dependent on their availability, in order to interact with Web services that wrap other application components.

[11] http://en.wikipedia.org/wiki/Android_(operating_system)

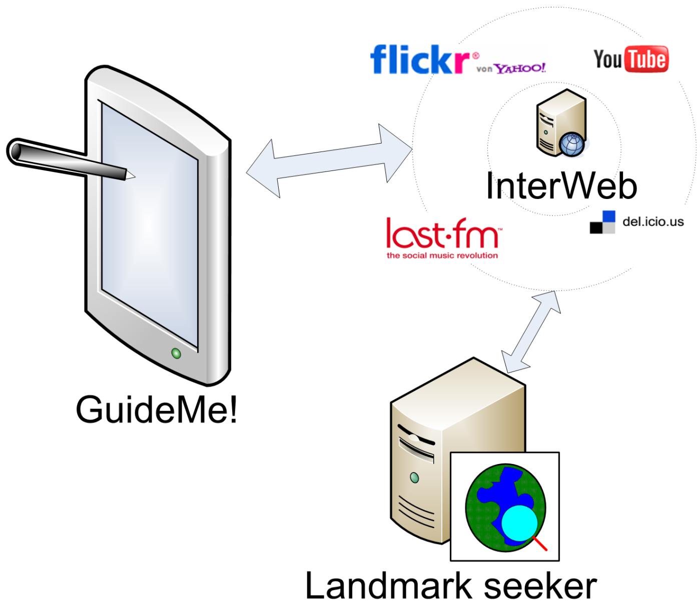


Figure 5.7 GuideMe! System architecture

Landmark Seeker Webservice

The Landmark Seeker is a SOAP[12] based web service that provides an interface with the method *getTopKcityTags* with the mandatory (String) parameter *cityName*. The response is an XML-formatted list of sightseeing labels corresponding to the given *cityName*. For example the query *hanover* could return the following list: "herrenhausen", "nordlb", "rathaus", "cityhall", and "marktkirche". The web service implementation is Java based (JDK 6) by using the Apache CXF[13] Framework. Its WSDL is available online[14].

InterWeb

InterWeb is a web service which integrates a number of different Web 2.0 tools like YouTube, Flickr, Ipernity, Slideshare, and Delicious. Most of the Web 2.0 applications and their orchestrations focus on finding resources related to user information needs. Portals like iGoogle and Netvibes also help to locate information distributed across different information sources. However, such portals typically provide no facilities for integration or merging of information obtained from these sources without providing a way of actually linking them together.

InterWeb provides a rich set of functions and a seamless overview of the entire set of distributed Web 2.0 resources. In this manner InterWeb serves as a Meta-Web

[12] http://www.w3.org/TR/soap/
[13] http://cxf.apache.org/
[14] http://pharos.l3s.uni-hannover.de:9966/landmarkseeker?wsdl

Figure 5.8 GuideMe! Graphical user interface

2.0 service. It provides a uniform interface for basic functionalities such as search. InterWeb is a PHP based implementation and is available online[15]

5.4.2 Landmark Extraction and Ranking

To provide users with a set of landmark resources we first need to identify the tags associated with landmarks. Our method for extracting landmark information from Web 2.0 exploits tags and social groups from *Flickr*. The full details of the landmark tags extraction are presented in the Section5.3 and here we provide a quick summary of our approach. One important difference to the original method is the change of classifier. While before we used the SVM-Light package, here we chose the *NaiveBayesMultinomial* classifier from WEKA[16] as it builds models faster and delivers comparable classification results. The final scoring function has been adapted to better fit the new, though, similar dataset and demo scenario, e.g. with respect to time constraints (i.e. the number of pictures queried for from Flickr). The representativeness score of each tag for a city c is computed as follows:

$$SCORE(t) = IRF_{R,T}(t) \cdot IUF_{R,T}(t) \cdot CTF(t), t \in T_c \tag{5.16}$$

Therefore, each city is assigned a ranking of landmark-related tags and this information is stored in a database. All expensive computations are performed offline and do not affect system performance.

[15] http://athena.l3s.uni-hannover.de:8000
[16] http://www.cs.waikato.ac.nz/~ml/weka/index.html

When calling the web service, first a database with previously learned city tags is queried for the given city name. If there are city tags – already learned according to the steps described above – in the database, the top-k city tags will be returned in descending order. If no city tags have been learned for the city yet, a search request with the name will be sent to the Flickr API. This request returns all relevant photos that have been tagged with the given city name. Then the classification and tag ranking procedures presented above are applied on the returned set of photos. The top 20 city tags will be saved to the database and the top 5 of them will be send to the client.

Using the set of landmark tags returned by the seeker service, InterWeb combines them into a query and executes it on available Web 2.0 services. In order to provide a representative and diverse overview of sights in Hanover on a small display, the results have to be ordered before presenting them to the user. The list of seightseeing results returned by InterWeb is first divided into sub-lists, one for each landmark. We sort each of the lists by relevance to the corresponding landmark and the top-ranked result from each list is displayed to a user. If some result is relevant to several landmarks it is copied to each corresponding list, see Fig. 5.9.

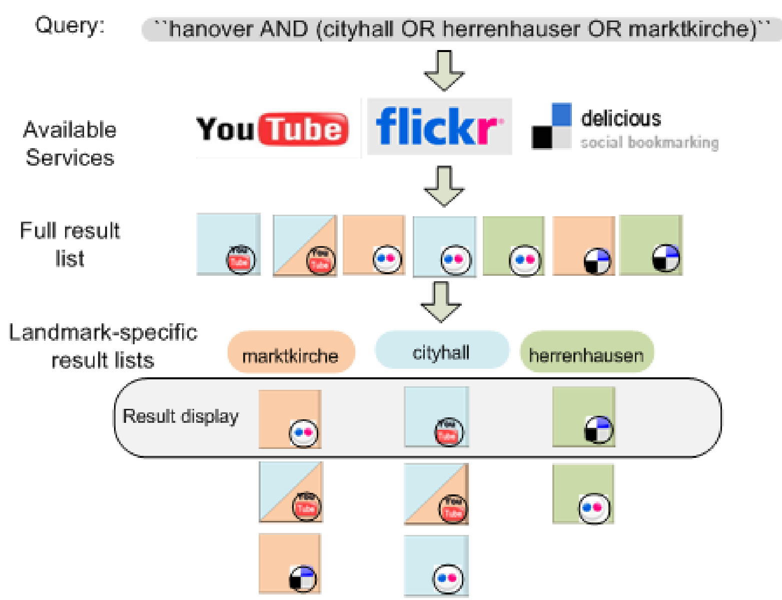


Figure 5.9 Collecting and Ranking Landmark Resources

The relevance depends both on the content of the resource and its popularity at the source service. The popularity can be derived from the number of views, comments and ratings of the resource. Thus the relevance can be computed as a weighted sum of these factors. Let us define $Sim(r, t)$ as the textual similarity between resource r's description and tag t, V_r as the number of views of resource r, C_r as the number of comments assigned to r, R_r as a rating assigned by users, and P_r as the position of r in results ranking returned by a particular social service. We rank all landmark resources according to their relevance to a landmark using the Equation 5.17:

$$RELEVANCE(r) = \alpha \cdot Sim(r,t) + \beta \cdot V_r + \gamma \cdot C_r + \; + \delta \cdot R_r + \epsilon \cdot P_r, \qquad (5.17)$$

where α, β, γ, δ and ϵ are coefficients used for tuning the system.

While working with the system, the user selects the preferred data types such as images, videos, or presentations at the options page and saves these settings. She types the search terms into the text field at the exploration page and starts the execution. The results are shown as a ranked list and the user selects the resource of her interest and can view it using the integrated browser. The demonstration application package can be downloaded from our page[17] and can be installed on any mobile device with android operating system or an android emulator available for PC.

This system for federated search of Web 2.0 resources related to these landmarks provides a user with a representative and diverse overview for sightseeing. An open question is what is the best type of resource for a particular landmark. Some static objects like buildings or paintings look good on photos, while objects like church bells call for a video representation. It might be interesting to explore how well different types of resources are suited to visualize specific types of sights.

In the section above we demonstrated how Flickr social groups could be used in the working application for landmark search. This system shows that our method for landmark finding is efficient enough to be implemented in a mobile setting. In addition, we smoothly replaced SVM classification with the Naive Bayes algorithm and proved that our solution does not depend on a particular machine learning method, but rather shows general way to incorporate social links into advanced information retrieval applications.

5.5 Recommending URLs on Twitter

In this section we consider a problem of link recommendation in Twitter. The goal of this new research focus is twofold. First, we would like to explore a social dimension in a different information retrieval problem coming from the recommender systems area. Second, it is important to consider another dataset and ensure that social links could be effectively used in different types of social networks.

The Twitter microblogging platform is becoming one of the most popular social media services, where people intensively interact with each other by posting directed and undirected short texts to personal message streams. Twitter users like to chat, to post thoughts and share information. In about 22% of Twitter messages users include some URL [Boyd *et al.*, 2010], while there is no reports on how much of these links are really checked out by other users. This large quanity of hyperlinks pose a danger

[17] http://l3s.de/~zerr/guideme/

of severe information overload of Twitter users. We hypothesise that automatic real-time ranking of recommendations posted in these streams should provide a great help for Twitter users.

In this section we study the user preferences on URLs that were explicitly or implicitly recommended to Twitter users by their information providers. We would like to cases when users explicitly express their interest in a certain URL by posting it to their own message streams, considering that the same URL was posted earlier by their friends within a short time window. Our approach allows to recommend interesting link using information about user social interactions with other users.

5.5.1 Twitter Dataset

In order to get a representative set of socially active Twitter users, we crawled the directory JustTweetIt (accessed in November 2009) where users register and assign themselves to categories describing their specialty (e.g. "lawyer") or interest (e.g. "gaming"). We selected ∼17,000 users and fetched their Twitter profiles using the Twitter API. In order to experiment only with users which we believe belong to the majority at Twitter, we removed all very popular users with more than 1000 followers. At the same time, in order to make the task of social recommendation realistic, we also considered only users with at least 10 and at most 300 friends.

In order to acquire a sample of user accounts representing real people using Twitter for personal needs and not organizations, we removed all profiles not containing one of the following phrases in their 'bio" part (short self-description): "i am", "i like", "i love", "my name is", and their variations. Finally, we crawled for all tweets with URLs posted by these users using API of the social search engine Topsy[18], which decodes URLs in case they are shortened. Since Topsy returns only up to 500 recent URLs per user, we selected only users with from 50 to 450 URLs, for whose we were sure to have their entire sharing history. Finally, we had 267 users appropriate for our experiments, who posted ∼15,000 unique URLs during the period from January 2008 till January 2010. For these users we also downloaded their lists of friends (∼29,000 unique users), URLs posted by their friends (∼600,000 unique URLs) and up to 3,200 posts (due to Twitter API limitations).

5.5.2 Prediction of Retweeting Behavior Using Social Links

We considered a real-world situation in which a user "follows" dozens of peers. They post a large number of tweets and many of these posts contain URLs interesting to the user. We want to recommend her to look at the most interesting URLs which she might like. As a natural notion of relevance we consider re-posting behavior. If the user posted a link which was previously published by one of the users she knows,

[18] otter.topsy.com

this link was a relevant candidate to be recommended. In our analysis we distinguish two types of relationships: by "friends" we call all users who are explicitly "followed" by the user; by "communicators" we call a subset of users' "friends", who have also received at least one personal message from the user (they are recepients mentioned in the metadata of posts that we downloaded).

To identify the best link recommendation strategy we studied a proportion of relevant link candidates within time spans of different size. Each pair of a user and a relevant link represents an implicit "query". All links posted by the user's friends within a specified time span before the relevant link posting is a set of candidate links. It is always just one relevant link we are looking for, but it could be posted by different friends independently, so it could be two or more identical relevant recommendation candidates in a pool.

In such a setup $Precision$ can be computed as $\frac{1}{CandidatePoolSize}$ and $Recall$ is defined as binary function, having 1 if relevant url appears in a candidate pool and 0 otherwise. This definition of $Precision$ is similar to $Precision@K$, as its value depends on the number of retrieved documents. Ideally, we want a candidate pool with a single relevant link and $Precision$ describes how effective a strategy of limiting a candidate pool size. On the other hand, while removing candidates from a pool we can delete relevant links, so $Recall$ shows if query could be answered after shrinking the pool.

For experiment we picked 16% of available 15764 queries (user-url pairs) which had at least one relevant url within past 24 hours. Our observation is that large number of Twitter links are relatively rare and cannot be recommended based on friendship connections. We consider only situations where such recommendation is possible given a one-day snapshot, taking into account highly dynamic nature of the Twitter information flow. We analyze $Recall$ and $Precision$ metrics macro-averaged across 2546 queries of 267 different users (see Figure reffig:chart1). The metrics are computed on time spans starting at query posting time and going back by one hour, two hours, etc. We considered candidate pools computed on friends and communicators separately.

As expected, average Recall increases with the size of candidates set, while $Precision$ drops down due to the large number of irrelevant links. Link recommendation functionality should be biased towards increased $Precision$. Most interestingly, our results suggest that prediction could be made solely based on communicators links, so it is a simple and effective way to increase recommendation effectiveness.

Once we select a set of links to be recommended, we need to rank links within each selected candidate pool and recommend only a few top-ranked urls. Our first strategy is to rank links by their popularity among user's friends or communicators. $Popularity$ measure is the number of friends (or communicators) who posted this link within a specified time period. This method requires only information collected from the user's friends in the last 24 hours and is suitable for new Twitter users. If the user already has some links posted in the past, we propose to use an alternative algorithm in which all links are ranked based on $JaccardSimilarity$ between the user

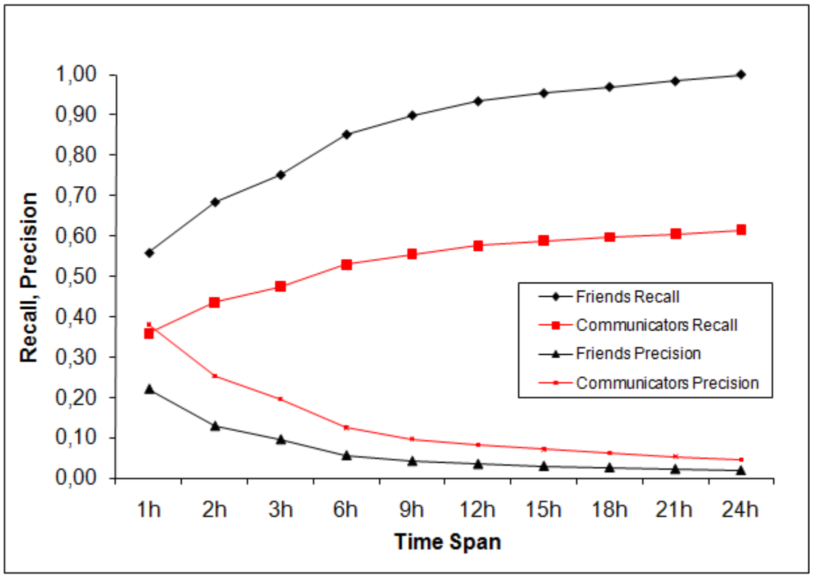


Figure 5.10 Recall and Precision for Pools in 24H

and an information provider.

The similarity is defined as $J = \frac{A \cup B}{A \cap B}$, where A is a set of links posted by user and B is a set of links posted by some of her friends. The links in the current pool (last 24 hours) are excluded from $JaccardSimilarity$ computation to distinguish between test and training sets.

We tested both ranking strategies on sets of friends and sets of communicators. For the evaluation we used Mean Reciprocal Rank (MRR) and Success at 1st Rank (S@1). Evaluation results are presented in Figure 5.11. We report selected evaluation metrics for the following link recommendation strategies: *Popularity* method computed on friends (FP) and communicators (CP) and $JaccardSimilarity$ algorithm also applied to friends (FJ) and communicators (CJ).

We observe that both ranking strategies are effective, while Jaccard Similarity is significantly better in all cases (paired t-test, p ¡0.05). The best S@1 we obtain with pool of links posted by communicators in the last hour, here we recommend the relevant link at the first rank in 35% of the cases. Our experiments confirm that ranking based on *Popularity* is an effective link recommendation method. In addition, if some training data is available from previous user activity, we can outperform this baseline using $JaccardSimilarity$ ranking.

Considering this novel problem of link recommendation within microblogging networks, our analysis shows that social interconnections can be used for recommending interesting urls to a user. The two proposed link recommendation strategies are using different types of social information. Our experimental evaluation on Twitter data has shown that both information about user's connections and their type could be used for effective link recommendations. However, we realize that our preliminary

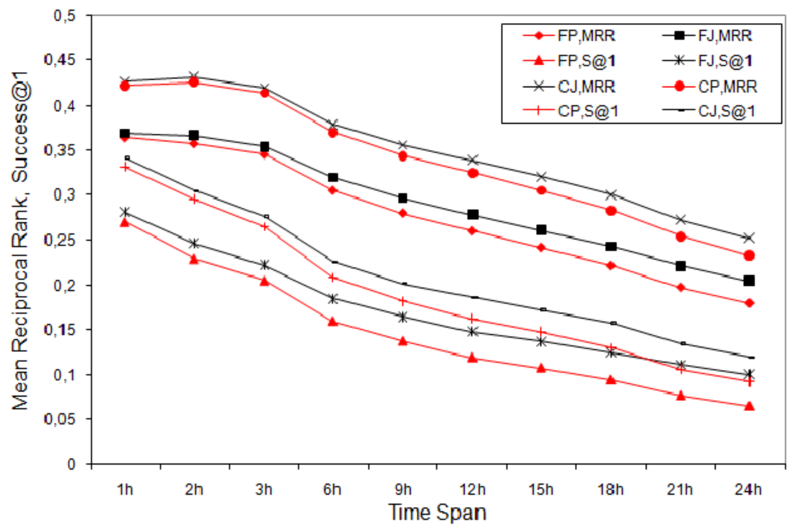


Figure 5.11 MRR, S@1 for Popularity and Jaccard

experiments do not represent a strong evaluation and have to be extended later. We rather suggest that current results look promising and usage of the social links for recommendation algorithms on Twitter should be investigated in more detail.

The next step might be to apply advanced collaborative filtering methods to this problem and explore more social dimensions like user's global popularity and social activity level. Another related research dimension is analysis of users' content preferences with respect to the link domain types.

5.6 Discussion

Social links have a high potential as means to improve user experience in social networks. In this chapter we considered two scenarios, in which social links support information retrieval tasks. First, we show usefulness of explicit user groups for the landmark finding task. Second, we introduce real-time link recommendation based on user's social connections.

In the first part of this chapter we addressed the problem of identifying landmark pictures using Flickr user groups information. Our algorithm exploits Flickr tags and groups information, without relying on GPS coordinates. For finding relevant landmark-related tags we apply an SVM classifier for which the training data – both positive and negative examples – is extracted from thematical Flickr groups. The positive examples are chosen from traveling and landmark related groups, while the negative examples come from groups with generic photographical interests. Our results show that the two-class SVM classifier effectively finds landmark photos based on Flickr groups training data, and is able to recognize landmarks which are not explicitly included in the training set. User evaluation results demonstrate that our method outperforms state-of-the-art system relying on GPS information for solving landmark finding task. The algorithm we described in the present paper could be generalized to help identifying not only city landmarks but also other topical photos, such as "animals", "flowers" or "cars".

Later, we consider Web 2.0 resources and metadata for answering the diverse and complex information needs. We continue the landmark finding scenario, in which we identify and extract landmark information from multiple social platforms and compile a representative summary for a given city. Using Flickr groups and user's social links again, we present a mobile search interface which retrieves landmark resources from sites like Flickr, YouTube, Delicious, etc. and fuses them. Our algorithm provides an efficient extraction of landmarks from Web 2.0 sources. Our GuideMe! system for federated search of Web 2.0 resources collects and ranks landmarks, providing a user with a representative and diverse overview for sightseeing.

In the last part of this chapter we studied the novel problem of link recommendation within microblogging networks. We analyzed how social interconnections can be used for recommending interesting URLs to a user. We proposed two link recommendation strategies utilizing different types of social information. Our experimental evaluation on Twitter data has shown that both information about user's connections and their type could be used for effective link recommendations.

In the next chapter we review the main findings of this thesis and discuss our results. We summarize our research on social and contextual links applied to desktop, enterprise and Web search domains. We conclude our work with several interesting open problems, which can be considered for future investigation.

6 Conclusions

The rapid development of information space poses great challenges for the information retrieval community. Researchers and developers continuously strive to provide more efficient and accurate algorithms, to identify what users want to see and how to find this information. Until recently, the two corner stones of such algorithms were text and hyperlinks. In this thesis we studied social and contextual links as a third dimension for ranking, applied to desktop, enterprise and social networks on the Web. We explored different ways to personalize user search and recommendation experience, presented several novel algorithms and demonstrated achieved improvement in several user studies. This chapter summarizes our research contributions in the three domains mentioned above and discusses issues open for future development.

6.1 Summary of Contributions

Our main focus in this work was to apply social and contextual links for ranking and recommendation within several information management domains. In Chapter 2 we provided a survey of information retrieval foundations, relevant to our topic. We characterized the information domains of interest and reviewed existing issues and solutions. This way we built a common ground for understanding the problems and approaches presented in the following chapters.

In Chapter 3 we considered how contextual and social links could improve desktop search experience. We elaborated on the need for logging the desktop activity data and creating a common collection for desktop search evaluation. First, we described the design of such a dataset and defined a set of events and applications for logging. Second, we developed the necessary logging tools and collected a sample activity-based desktop dataset. We presented characteristics of the dataset and analyzed user behavior.

We also considered the problem of personal resources being stored across the Web 2.0 platforms, non-searchable by regular desktop tools. We tackled this problem in the deskWeb 2.0 application, which combines results found on the desktop with results from social networks like Flickr, YouTube, and Delicious, and presents the user with an overview of the available resources. We supported the effectiveness of this method with a user study.

In the Chapter 4 we investigated the use of social links in enterprise setting and presented our solution to personalized social search in enterprise. Our algorithm uses the user's social relations and search results are re-ranked according to connections with individuals in the user's social network. We studied the effectiveness of several social network types for personalization: (1) *familiarity-based* network of people related to the user through explicit familiarity connection; (2) *similarity-based* network of people "similar" to the user as reflected by their social activity; (3) *overall* network that provides both relationship types. For comparison we also experimented with topic-based personalization that is based on the user's related terms, aggregated from several social applications.

We evaluated the contribution of the different personalization strategies by an off-line experiment and by a user study within the IBM organization. In the off-line study we applied bookmark-based evaluation, suggested recently, that exploits data gathered from a social bookmarking system to evaluate personalized retrieval. In the on-line study we analyzed the feedback of 250 employees exposed to the alternative personalization approaches. We have shown that both in the off-line experiment and in the user study social network based personalization significantly outperforms topic-based personalization and non-personalized social search.

Chapter 5 presents our contributions to search and recommendation in social networks on the Web. First, we proposed a new method to identify landmark photos using tags and social Flickr groups. It applies an SVM classifier for which the training data is extracted from thematical Flickr groups. In contrast to similar modern systems, our approach is also applicable when GPS-coordinates for photos are not available. The presented user study shows that the proposed method outperforms state-of-the-art systems for landmark finding.

Next, we considered a wider set of Web 2.0 applications for the same task. We presented a mobile search application *GuideMe!*, which retrieves landmark resources from social sites like Flickr, YouTube, or Delicious, and supports the user with the overview of the available resources. Our contributions include a modified algorithm using Naive Bayes classification for efficient extraction of landmarks from Web 2.0 sources, a system for federated search of Web 2.0 resources related to these landmarks and a ranking strategy to provide a user with a representative and diverse overview for sightseeing.

Later in this chapter, we introduced a novel problem of recommending links to the users of microblogging platforms. We studied URL recommendation based on two types of social connections and proposed two algorithms for links recommendation. The evaluation on the Twitter data has shown that recommendation based on social information alone achieves high accuracy level.

6.2 Open Problems

In this thesis we proposed a number of algorithms and applications, but this set of solutions is by no means complete. There is always a room for improvement, both for upgraded versions of provided methods and completely new approaches. Here we discuss several related research directions that we consider interesting for future investigation. We split them according to the structure in which relevant problems appear in the thesis.

6.2.1 Desktop Domain

Many of the desktop search problems have a purely technical nature, since they are related to the operating system, particular applications and version upgrades. But some issues represent conceptual design limitations and could be analyzed in more detail. For example, the experiments with activity logs revealed the problem of data freshness. Once the data is collected and encrypted, it starts to lose its relevance, as users restructure their data, install new applications and, ultimately, become unaccessible due to moving to other locations. It is interesting to modify the data collecting procedure so, that the desktop dataset becomes more independent from the users' future behaviour and keeps its value over a longer period. This leads to related problems of privacy preservation mechanism. While currently used encryption gives participants a high level of trust, it severely limits dataset usage opportunities. It would be worthwhile to study additional methods for data anonymization, which provide more data for future analysis. A desktop usage analysis could also be extended with additional dependencies between files, emails and user tasks.

We also might be able to improve integration of desktop search and personal resources on social platforms. It is interesting to see how search results retrieved from different places should be better diversified. This problem suggests some optimal results merging strategy, which would re-rank results down the ranking, given that some relevant information already appears in top results.

6.2.2 Enterprise Domain

As previous work showed, not all queries should be personalized. In our experiments we simulated personal queries with tags used for bookmarking by the user, in the off-line experiment, and with tags the user was tagged with, in the user study. In both cases these types of personal queries are limited and do not cover the whole spectrum of possible personal queries, but rather a subset that is likely to benefit from personalization and which can be judged by the methods in use. A relevant research question would be how to select between queries which should and should not be personalized in real time? Another research problem left out for future experiments is tuning of the coefficients used in similarity equation.

6.2.3 Social Networks Domain

There are many interesting research problems left open for the landmark finding algorithm. One could study parameter estimation for measuring tag representativeness scores, as well as experiment with other classifiers and employ feature selection methods. Additionally, tags can be enriched with their corresponding semantic classes according to the WordNet lexicon to further improve our algorithm. Alternatively, it is worthwhile to test the generality of our algorithm not only on city landmarks but also on other topical photos, such as cars, mobile phones, etc.

Another question is what is the best type of resource for a particular landmark. Some static objects, like buildings or paintings, look good on photos, while objects like church bells call for a video representation. One could explore how well different types of resources are suited to visualize specific types of sights.

Regarding the link recommendation algorithm, it might be interesting to apply advanced collaborative filtering methods to this problem and explore more social dimensions like a user's global popularity and social activity level. In addition, it would be beneficial to analyze users' content preferences and adjust recommendations based on the link domain types.

6.3 Final Remarks

We believe that social and contextual information are crucial factors for the future information retrieval. A particular type of necessary information depends on the nature of search task and properties of the environment. Each search domain determines which social or contextual links could be obtained. In this thesis we focused on application of social and activity data for various purposes. Still, much of research in this area is going on and continuously extends a work presented in here.

As we are living in a digital world, the effectiveness of search engines will continue to play a big role. We think that research on information retrieval will expand, while social and contextual links will help people to find pearls of useful information in an ocean of irrelevant data.

A

Curriculum Vitae

Sergey Chernov, born on September 17 1980, in Volgograd, Russian Federation.

May 2005 - present	PhD student at L3S Research Center, Leibniz University of Hannover, Germany
Aug 2008 - Nov 2008	Research intern at IBM Haifa Research Center, Haifa, Israel
Oct 2003 - Feb 2005	Master Studies in Computer Science, Saarland University, (under the fellowship from the International Max Planck Research School for Computer Science)
2001 - 2003	Master Studies in Computer Science, Volgograd State Technical University, Volgograd, Russian Federation
1997 - 2001	Bachelor Studies in Computer Science, Volgograd State Technical University, Volgograd, Russian Federation

Bibliography

[Abbasi *et al.*, 2007] Rabeeh Abbasi, Steffen Staab, and Philipp Cimiano. Organizing resources on tagging systems using t-org. In *In proceedings of Workshop on Bridging the Gap between Semantic Web and Web 2.0 at ESWC 2007*, Innsbruck, Austria, June 2007.

[Abbasi *et al.*, 2009a] Rabeeh Abbasi, Sergey Chernov, Wolfgang Nejdl, Raluca Paiu, and Steffen Staab. Exploiting flickr tags and groups for finding landmark photos. In Mohand Boughanem, Catherine Berrut, Josiane Mothe, and Chantal Soulé-Dupuy, editors, *ECIR*, volume 5478 of *Lecture Notes in Computer Science*, pages 654–661. Springer, 2009.

[Abbasi *et al.*, 2009b] Rabeeh Abbasi, Sergey Chernov, Wolfgang Nejdl, Raluca Paiu, and Steffen Staab. Exploiting flickr tags and groups for finding landmark photos. In *Proc. Advances in Information Retrieval, 31th European Conference on IR Research (ECIR 2009)*, pages 654–661, Toulouse, France, 2009.

[Abel *et al.*, 2008] Fabian Abel, Nicola Henze, and Daniel Krause. Ranking in folksonomy systems: can context help? In *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*, pages 1429–1430, New York, NY, USA, 2008. ACM.

[Agichtein *et al.*, 2006a] Eugene Agichtein, Eric Brill, Susan Dumais, and Robert Ragno. Learning user interaction models for predicting web search result preferences. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–10, New York, NY, USA, 2006. ACM Press.

[Agichtein *et al.*, 2006b] Eugene Agichtein, Eric Brill, Susan Dumais, and Robert Ragno. Learning user interaction models for predicting web search result preferences. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–10, New York, NY, USA, 2006. ACM Press.

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

[Alvarado *et al.*, 2003] Christine Alvarado, Jaime Teevan, Mark S. Ackerman, and David Karger. Surviving the information explosion: How people find their electronic information. Technical Report AIM-2003-006, MIT AI Lab, April 2003.

[Amati *et al.*, 2007] Giambattista Amati, Claudio Carpineto, and Giovanni Romano, editors. *Advances in Information Retrieval, 29th European Conference on IR Research, ECIR 2007, Rome, Italy, April 2-5, 2007, Proceedings*, volume 4425 of *Lecture Notes in Computer Science*. Springer, 2007.

[Amatriain *et al.*, 2009] Xavier Amatriain, Neal Lathia, Josep M. Pujol, Haewoon Kwak, and Nuria Oliver. The wisdom of the few: a collaborative filtering approach based on expert opinions from the web. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '09, pages 532–539, New York, NY, USA, 2009. ACM.

[Amitay *et al.*, 2009] Einat Amitay, David Carmel, Nadav Har'El, Shila Ofek-Koifman, Aya Soffer, Sivan Yogev, and Nada Golbandi. Social search and discovery using a unified approach. In *HyperText*. ACM, 2009.

[Anagnostopoulos *et al.*, 2008] Aris Anagnostopoulos, Ravi Kumar, and Mohammad Mahdian. Influence and correlation in social networks. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, pages 7–15, New York, NY, USA, 2008. ACM.

[Asonov, 2004] Dmitri Asonov. *Querying Databases Privately: A New Approach to Private Information Retrieval*, volume 3128 of *Lecture Notes in Computer Science*. Springer, 2004.

[Au-Yeung *et al.*, 2008] C. M. Au-Yeung, N. Gibbins, and N. Shadbolt. A study of user profile generation from folksonomies. In *Proceedings of the WWW 2008 Social Web and Knowledge Management, Social Web Workshop*, April 2008.

[AUT, 2010] Autonomy IDOL Enterprise Desktop Search. http://www.autonomy.com/content/products/enterprise-search/index.en.html, 2010.

[Baeza-Yates and Ribeiro-Neto, 1999] Ricardo A. Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.

[Baker *et al.*, 2007] Ross H. Baker, Jacob E. Bardman, Patrick Baudisch, Richard Boardman, Mary Czerwinski, Nicolas B. Duarte, Danyel Fisher, Eric Freeman, David Gelernter, and Aydin Haririnia. *Beyond the Desktop Metaphor*. The MIT Press, 2007.

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

[Barreau and Nardi, 1995] Deborah Barreau and Bonnie A. Nardi. Finding and reminding: file organization from the desktop. *SIGCHI Bulletin*, 27(3):39–43, 1995.

[Belkin, 2008] Nicholas J. Belkin. Some(what) grand challenges for information retrieval. *SIGIR Forum*, 42(1):47–54, 2008.

[Bell *et al.*, 2007] Robert Bell, Yehuda Koren, and Chris Volinsky. Modeling relationships at multiple scales to improve accuracy of large recommender systems. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '07, pages 95–104, New York, NY, USA, 2007. ACM.

[Bender *et al.*, 2008] Matthias Bender, Tom Crecelius, Mouna Kacimi, Sebastian Michel, Thomas Neumann, Josiane Xavier Parreira, Ralf Schenkel, and Gerhard Weikum. Exploiting social relations for query expansion and result ranking. In *Proceedings of the 24th ICDE 2008*, Lecture Notes in Computer Science, pages 501–506. IEEE, 2008.

[Benz *et al.*, 2008] Dominik Benz, Marko Grobelnik, Andreas Hotho, Robert Ja"schke, Dunja Mladenic, Vito D. P. Servedio, Sergej Sizov, and Martin Szomszor. Analyzing tag semantics across collaborative tagging systems. In Harith Alani, Steffen Staab, and Gerd Stumme, editors, *Social Web Communities*, number 08391 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2008. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik.

[Berg and Forsyth, 2007] Tamara Lee Berg and David Forsyth. Automatic ranking of iconic images. Technical report, EECS Department, University of California, Berkeley, January 2007.

[Borland and Ingwersen, 1997] Pia Borland and Peter Ingwersen. The development of a method for the evaluation of interactive information retrieval systems. *Journal of Documentation*, 53(3):225–250, 1997.

[Boyd *et al.*, 2010] Danah Boyd, Scott Golder, and Gilad Lotan. Tweet, Tweet, Retweet: Conversational Aspects of Retweeting on Twitter. In *HICSS'10*, 2010.

[Bozzon *et al.*, 2009] A. Bozzon, S. Comai, P. Fraternali, and G. Toffetti Carughi. Towards Web 2.0 Applications: A Conceptual Model for Rich Internet Applications. In San Murugesan, editor, *Handbook on Web 2.0, 3.0 and X.0: Technologies, Business and Social Applications*, pages 75–95. IGI Global, November 2009.

[Broder, 2002] Andrei Broder. A taxonomy of web search. *SIGIR Forum*, 36(2):3–10, 2002.

[Brusilovsky *et al.*, 2007] Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl, editors. *The Adaptive Web, Methods and Strategies of Web Personalization*, volume 4321 of *Lecture Notes in Computer Science*. Springer, 2007.

[Budzik and Hammond, 2000] Jay Budzik and Kristian J. Hammond. User interactions with everyday applications as context for just-in-time information access. In *IUI '00: Proceedings of the 5th international conference on Intelligent user interfaces*, pages 44–51, New York, NY, USA, 2000. ACM.

[Bush, 1945] Vannevar Bush. As we may think. *The Atlantic Monthly*, July 1945.

[Carman *et al.*, 2008] Mark J. Carman, Mark Baillie, and Fabio Crestani. Tag data and personalized information retrieval. In *SSM '08: Proceeding of the 2008 ACM workshop on Search in social media*, pages 27–34, New York, NY, USA, 2008. ACM.

[Carmel *et al.*, 2009] David Carmel, Naama Zwerdling, Ido Guy, Shila Ofek-Koifman, Nadav Har'El, Inbal Ronen, Erel Uziel, Sivan Yogev, and Sergey Chernov. Personalized social search based on the user's social network. In David Wai-Lok Cheung, Il-Yeol Song, Wesley W. Chu, Xiaohua Hu, and Jimmy J. Lin, editors, *CIKM*, pages 1227–1236. ACM, 2009.

[Catarci *et al.*, 2006] Tiziana Catarci, Benjamin Habegger, and Antonella Poggi. Intelligent user task oriented systems. In *Proceedings of the Workshop on Personal Information Management held at the 29th ACM International SIGIR Conf. on Research and Development in Information Retrieval*, 2006.

[Chakrabarti, 2002] Soumen Chakrabarti. *Mining the Web: Analysis of Hypertext and Semi Structured Data*. Morgan Kaufmann, August 2002.

[Chen *et al.*, 2010] Jilin Chen, Rowan Nairn, Les Nelson, Michael Bernstein, and Ed Chi. Short and tweet: experiments on recommending content from information streams. In *Proceedings of the 28th international conference on Human factors in computing systems*, CHI '10, pages 1185–1194, New York, NY, USA, 2010. ACM.

[Chernov *et al.*, 2005] Sergey Chernov, Pavel Serdyukov, Matthias Bender, Sebastian Michel, Gerhard Weikum, and Christian Zimmer. Database selection and result merging in p2p web search. In Gianluca Moro, Sonia Bergamaschi, Sam Joseph, Jean-Henry Morin, and Aris M. Ouksel, editors, *DBISP2P*, volume 4125 of *Lecture Notes in Computer Science*, pages 26–37. Springer, 2005.

[Chernov *et al.*, 2006a] Sergey Chernov, Gianluca Demartini, and Julien Gaugaz. L3s research center at trec 2006 enterprise track. In Ellen M. Voorhees and Lori P. Buckland, editors, *TREC*, volume Special Publication 500-272. National Institute of Standards and Technology (NIST), 2006.

[Chernov *et al.*, 2006b] Sergey Chernov, Tereza Iofciu, Wolfgang Nejdl, and Xuan Zhou. Extracting semantics relationships between wikipedia categories. In Max Völkel and Sebastian Schaffert, editors, *SemWiki*, volume 206 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2006.

[Chernov *et al.*, 2006c] Sergey Chernov, Christian Kohlschütter, and Wolfgang Nejdl. A plugin architecture enabling federated search for digital libraries. In Shigeo Sugimoto, Jane Hunter, Andreas Rauber, and Atsuyuki Morishima, editors, *ICADL*, volume 4312 of *Lecture Notes in Computer Science*, pages 202–211. Springer, 2006.

[Chernov *et al.*, 2007a] Sergey Chernov, Enrico Minack, and Pavel Serdyukov. Converting desktop into a personal activity dataset. In *RCDL '07: Proceedings of 8th Nationational Russian Conference on Digital Libraries*, 2007.

[Chernov *et al.*, 2007b] Sergey Chernov, Pavel Serdyukov, Paul-Alexandru Chirita, Gianluca Demartini, and Wolfgang Nejdl. Building a desktop search test-bed. In Amati et al. [2007], pages 686–690.

[Chernov *et al.*, 2008] Sergey Chernov, Gianluca Demartini, Eelco Herder, Michal Kopycki, and Wolfgang Nejdl. Evaluating personal information management using an activity logs enriched desktop dataset. In *Personal Information Management Workshop at CHI 2008*, April 2008.

[Chernov, 2008] Sergey Chernov. Task detection for activity-based desktop search. In Sung-Hyon Myaeng, Douglas W. Oard, Fabrizio Sebastiani, Tat-Seng Chua, and Mun-Kew Leong, editors, *SIGIR*, page 894. ACM, 2008.

[Chirita and Nejdl, 2006] Paul-Alexandru Chirita and Wolfgang Nejdl. Analyzing user behavior to rank desktop items. In Fabio Crestani, Paolo Ferragina, and Mark Sanderson, editors, *SPIRE*, volume 4209 of *Lecture Notes in Computer Science*, pages 86–97. Springer, 2006.

[Chirita *et al.*, 2005] Paul Alexandru Chirita, Wolfgang Nejdl, Raluca Paiu, and Christian Kohlschütter. Using ODP metadata to personalize search. In *SIGIR*

'05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, pages 178–185, New York, NY, USA, 2005. ACM.

[Chirita *et al.*, 2006a] Paul-Alexandru Chirita, Stefania Costache, Wolfgang Nejdl, and Raluca Paiu. Beagle^{++}: Semantically enhanced searching and ranking on the desktop. In *ESWC*, pages 348–362, 2006.

[Chirita *et al.*, 2006b] Paul Alexandru Chirita, Julien Gaugaz, Stefania Costache, and Wolfgang Nejdl. Desktop context detection using implicit feedback. In *Proceedings of the 2nd Workshop on Personal Information Management*. ACM Press, 2006.

[Chirita *et al.*, 2006c] Paul Alexandru Chirita, Julien Gaugaz, Stefania Costache, and Wolfgang Nejdl. Desktop context detection using implicit feedback. In *In Proceedings of the Workshop on Personal Information Management held at the 29th ACM International SIGIR Conf. on Research and Development in Information Retrieval*. ACM Press, 2006.

[Chirita *et al.*, 2007a] Paul Alexandru Chirita, Stefania Costache, Wolfgang Nejdl, and Siegfried Handschuh. P-tag: large scale automatic generation of personalized annotation tags for the web. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 845–854, New York, NY, USA, 2007. ACM.

[Chirita *et al.*, 2007b] Paul Alexandru Chirita, Claudiu S. Firan, and Wolfgang Nejdl. Personalized query expansion for the web. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 7–14, New York, NY, USA, 2007. ACM.

[Claypool *et al.*, 2000] Mark Claypool, Phong Le, Makoto Waseda, and David Brown. Implicit interest indicators. In *In Intelligent User Interfaces*, pages 33–40. ACM Press, 2000.

[Claypool *et al.*, 2001] Mark Claypool, Phong Le, Makoto Wased, and David Brown. Implicit interest indicators. In *IUI '01: Proceedings of the 6th international conference on Intelligent user interfaces*, pages 33–40, New York, NY, USA, 2001. ACM.

[Cleverdon, 1997] Cyril Cleverdon. The cranfield tests on index language devices. In *Readings in information retrieval*, pages 47–59, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.

[Costache *et al.*, 2010] Stefania Costache, Julien Gaugaz, Ekaterini Ioannou, Claudia Niederee, and Wolfgang Nejdl. Detecting contexts on the desktop using bayesian networks. In *Desktop Search Workshop at SIGIR 2008*, pages 3–6, July 2010.

[Cutrell *et al.*, 2006a] Edward Cutrell, Susan T. Dumais, and Jaime Teevan. Searching to eliminate personal information management. *Commun. ACM*, 49(1):58–64, 2006.

[Cutrell *et al.*, 2006b] Edward Cutrell, Daniel Robbins, Susan Dumais, and Raman Sarin. Fast, flexible filtering with phlat. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 261–270, New York, NY, USA, 2006. ACM.

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

[Ding *et al.*, 2006] Yi Ding, Xue Li, and Maria E. Orlowska. Recency-based collaborative filtering. In *Proceedings of the 17th Australasian Database Conference - Volume 49*, ADC '06, pages 99–107, Darlinghurst, Australia, Australia, 2006. Australian Computer Society, Inc.

[Dmitriev *et al.*, 2010] Pavel Dmitriev, Pavel Serdyukov, and Sergey Chernov. Enterprise and desktop search. In Michael Rappa, Paul Jones, Juliana Freire, and Soumen Chakrabarti, editors, *WWW*, pages 1345–1346. ACM, 2010.

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

[Dragunov *et al.*, 2005] Anton N. Dragunov, Thomas G. Dietterich, Kevin Johnsrude, Matthew McLaughlin, Lida Li, and Jonathan L. Herlocker. Tasktracer: a desktop environment to support multi-tasking knowledge workers. In *IUI '05: Proceedings of the 10th international conference on Intelligent user interfaces*, pages 75–82, New York, NY, USA, 2005. ACM.

[DT, 2007] DragonTalk Project. http://dragontalk.opendfki.de/, 2007.

[Dumais *et al.*, 2003a] S. Dumais, E. Cutrell, JJ Cadiz, G. Jancke, R. Sarin, and Daniel C. Robbins. Stuff i've seen: A system for personal information retrieval and re-use. In *SIGIR*, 2003.

[Dumais *et al.*, 2003b] Susan Dumais, Edward Cutrell, JJ Cadiz, Gavin Jancke, Raman Sarin, and Daniel C. Robbins. Stuff i've seen: a system for personal information retrieval and re-use. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 72–79, New York, NY, USA, 2003. ACM.

[Elsweiler and Ruthven, 2007] David Elsweiler and Ian Ruthven. Towards task-based personal information management evaluations. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 23–30, New York, NY, USA, 2007. ACM Press.

[Farrell and Lau, 2006] Stephen Farrell and Tessa Lau. Fringe contacts: People-tagging for the enterprise. In *Workshop on Collaborative Web Tagging, WWW 2006*, 2006.

[Freeman and Gelernter, 1996] Eric Freeman and David Gelernter. Lifestreams: a storage model for personal data. *SIGMOD Record*, 25(1):80–86, 1996.

[Freeman and Gelernter, 2007] Eric Freeman and David Gelernter. *Beyond the Desktop Metaphor*, chapter Beyond Lifestreams: The Inevitable Demise of the Desktop Metaphor, pages 19–48. MIT Press, 2007.

[Gaugaz *et al.*, 2008] Julien Gaugaz, Stefania Costache, Paul-Alexandru Chirita, Claudiu Firan, and Wolfgang Nejdl. Activity based links as a ranking factor in semantic desktop search. In *LA-WEB '08: Proceedings of the 2008 Latin American Web Conference*, pages 49–57, Washington, DC, USA, 2008. IEEE Computer Society.

[GD, 2011] Google Desktop. http://desktop.google.com, 2011.

[Guy *et al.*, 2008a] Ido Guy, Michal Jacovi, Noga Meshulam, Inbal Ronen, and Elad Shahar. Public vs. private: comparing public social network information with email. In *CSCW '08: Proceedings of the ACM 2008 conference on Computer supported cooperative work*, pages 393–402, New York, NY, USA, 2008. ACM.

[Guy *et al.*, 2008b] Ido Guy, Michal Jacovi, Elad Shahar, Noga Meshulam, Vladimir Soroka, and Stephen Farrell. Harvesting with SONAR: the value of aggregating social network information. In *CHI*. ACM, 2008.

[Gyllstrom and Soules, 2008] Karl Gyllstrom and Craig Soules. Seeing is retrieving: Building information context from what the user sees. In *IUI '08: Proceedings of the 13th international conference on Intelligent user interfaces*, pages 189–198, New York, NY, USA, 2008. ACM.

[Gyllstrom *et al.*, 2007] Karl Anders Gyllstrom, Craig Soules, and Alistair Veitch. Confluence: enhancing contextual desktop search. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 717–718, New York, NY, USA, 2007. ACM.

[Haveliwala, 2002] Taher H. Haveliwala. Topic-sensitive pagerank. In *WWW '02: Proceedings of the 11th international conference on World Wide Web*, pages 517–526, New York, NY, USA, 2002. ACM.

[Hawking, 2004] David Hawking. Challenges in enterprise search. In *Proceedings of the 15th Australasian database conference - Volume 27*, ADC '04, pages 15–24, Darlinghurst, Australia, Australia, 2004. Australian Computer Society, Inc.

[Herder, 2006] Eelco Herder. *Forward, back and home again : analyzing user behavior on the web.* PhD thesis, Enschede, 2006.

[Heymann *et al.*, 2008] Paul Heymann, Georgia Koutrika, and Hector Garcia-Molina. Can social bookmarking improve web search? In *WSDM '08: Proceedings of the international conference on Web search and web data mining*, pages 195–206, New York, NY, USA, 2008. ACM.

[Horvitz *et al.*, 1998] Eric Horvitz, Jack Breese, David Heckerman, David Hovel, and Koos Rommelse. The lumiere project: Bayesian user modeling for inferring the goals and needs of soft. In *Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence (UAI-98)*, pages 256–26, San Francisco, CA, 1998. Morgan Kaufmann.

[Hotho *et al.*, 2006a] A. Hotho, R. Jaschke, C. Schmitz, and G. Stumme. *Information retrieval in folksonomies: Search and ranking*, volume 4011, pages 411–426. Springer, 2006.

[Hotho *et al.*, 2006b] Andreas Hotho, Robert Ja"schke, Christoph Schmitz, and Gerd Stumme. Information retrieval in folksonomies: Search and ranking. In York Sure and John Domingue, editors, *The Semantic Web: Research and Applications*, volume 4011 of *LNAI*, pages 411–426, Heidelberg, June 2006. Springer.

[Huh *et al.*, 2007] Jina Huh, Lauretta Jones, Thomas Erickson, Wendy A. Kellogg, Rachel K. E. Bellamy, and John C. Thomas. Blogcentral: the role of internal blogs at work. In *CHI '07: CHI '07 extended abstracts on Human factors in computing systems*, pages 2447–2452, New York, NY, USA, 2007. ACM.

[Jaffe *et al.*, 2006] Alexandar Jaffe, Mor Naaman, Tamir Tassa, and Marc Davis. Generating summaries and visualization for large collections of geo-referenced photographs. In *MIR '06: Proceedings of the 8th ACM international workshop on Multimedia information retrieval*, pages 89–98. ACM, 2006.

[Jannach *et al.*, 2011] Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. *Recommender Systems An Introduction.* Cambridge University Press, 2011.

[Joachims *et al.*, 2005] T. Joachims, L. Granka, B. Pang, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In *ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 154–161, 2005.

[Joachims, 2002] Thorsten Joachims. *Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms.* Kluwer Academic Publishers, Norwell, MA, USA, 2002.

[Joachims, 2006] Thorsten Joachims. Training linear svms in linear time. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 217–226, Philadelphia, PA, USA, 2006. ACM.

[Jones and Teevan, 2007] William Jones and Jaime Teevan, editors. *Personal Information Management*, page 3. University of Washington Press, 2007.

[Jones *et al.*, 2007] M. Cameron Jones, Mert Bay, J. Stephen Downie, and Andreas F. Ehmann. A "do-it-yourself" evaluation service for music information retrieval systems. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, page 913, 2007.

[Kaptelinin, 2003] Victor Kaptelinin. Umea: translating interaction histories into project contexts. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 353–360, New York, NY, USA, 2003. ACM.

[Kautz *et al.*, 1997] Henry Kautz, Bart Selman, and Mehul Shah. Referral web: combining social networks and collaborative filtering. *Commun. ACM*, 40(3):63–65, 1997.

[Kelly, 2006] Diane Kelly. Evaluating personal information management behaviors and tools. *Commun. ACM*, 49(1):84–86, 2006.

[Kennedy and Naaman, 2008] Lyndon S. Kennedy and Mor Naaman. Generating diverse and representative image search results for landmarks. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 297–306, New York, NY, USA, 2008. ACM.

[Kennedy *et al.*, 2007] Lyndon Kennedy, Mor Naaman, Shane Ahern, Rahul Nair, and Tye Rattenbury. How flickr helps us make sense of the world: context and content in community-contributed media collections. In *MULTIMEDIA '07: Proceedings of the 15th international conference on Multimedia*, pages 631–640, New York, NY, USA, 2007. ACM.

[Kim and Chan, 2005] Hyoung Rae Kim and Philip K. Chan. Personalized search results with user interest hierarchies learnt from bookmarks. In Olfa Nasraoui, Osmar R. Zaïane, Myra Spiliopoulou, Bamshad Mobasher, Brij M. Masand, and Philip S. Yu, editors, *WEBKDD*, volume 4198 of *Lecture Notes in Computer Science*, pages 158–176. Springer, 2005.

[Kim and Croft, 2009] Jinyoung Kim and W. Bruce Croft. Retrieval experiments using pseudo-desktop collections. In *Conference on Information and Knowledge Management, Hong Kong, China, 2-6.11.09*, 2009.

[Kim and Croft, 2010] Jinyoung Kim and W. Bruce Croft. Ranking using multiple document types in desktop search. In *SIGIR '10: Proceedings of the 33rd annual international ACM SIGIR conference on Research and development in informaion retrieval*, New York, NY, USA, 2010. ACM.

[Kobsa, 2007a] Alfred Kobsa. Privacy-enhanced personalization. *Commun. ACM*, 50(8):24–33, 2007.

[Kobsa, 2007b] Alfred Kobsa. Privacy-enhanced web personalization. In Brusilovsky et al. [2007], pages 628–670.

[Kopycki, 2008] Michal Kopycki. Exploiting Implicit Feedback to Identify Usage Patterns on the Desktop - Bachelor's thesis, Department of Computer Science, Leibniz University of Hannover, Hannover, Germany, January 2008.

[Kraft *et al.*, 2006a] Reiner Kraft, Chi C. Chang, Farzin Maghoul, and Ravi Kumar. Searching with context. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 477–486, New York, NY, USA, 2006. ACM Press.

[Kraft *et al.*, 2006b] Reiner Kraft, Chi C. Chang, Farzin Maghoul, and Ravi Kumar. Searching with context. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 477–486, New York, NY, USA, 2006. ACM Press.

[Lee *et al.*, 2005] Uichin Lee, Zhenyu Liu, and Junghoo Cho. Automatic identification of user goals in web search. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 391–400, New York, NY, USA, 2005. ACM.

[Liu *et al.*, 2004] Fang Liu, Clement Yu, and Weiyi Meng. Personalized web search for improving retrieval effectiveness. *IEEE Trans. on Knowl. and Data Eng.*, 16(1):28–40, 2004.

[Lu *et al.*, 2007] Chang-Tien Lu, M. Shukla, S.H. Subramanya, and Yamin Wu. Performance evaluation of desktop search engines. In *Information Reuse and Integration, 2007. IRI 2007. IEEE International Conference on*, pages 110–115, Aug. 2007.

[Luxenburger *et al.*, 2008] Julia Luxenburger, Shady Elbassuoni, and Gerhard Weikum. Matching task profiles and user needs in personalized web search. In *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*, pages 689–698, New York, NY, USA, 2008. ACM.

[Manning *et al.*, 2008] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schtze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[Millen *et al.*, 2006] David R. Millen, Jonathan Feinberg, and Bernard Kerr. Dogear: Social bookmarking in the enterprise. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 111–120, New York, NY, USA, 2006. ACM.

[Minack *et al.*, 2010] Enrico Minack, Raluca Paiu, Stefania Costache, Gianluca Demartini, Julien Gaugaz, Ekaterini Ioannou, Paul-Alexandru Chirita, and Wolfgang Nejdl. Leveraging personal metadata for desktop search: The beagle++ system. *Web Semant.*, 8(1):37–54, 2010.

[Naaman *et al.*,] Mor Naaman, Yee Jiun Song, Andreas Paepcke, and Hector Garcia-Molina. Automatic organization for digital photographs with geographic coordinates. In *JCDL '04*, Tuscon, AZ, USA.

[Naaman *et al.*, 2010] Mor Naaman, Jeffery Boase, and Chin-Hui Lai. Is it Really About Me? Message Content in Social Awareness Streams. In *CSCW'10*, 2010.

[Noll and Meinel, 2007] Michael G. Noll and Christoph Meinel. Web search personalization via social bookmarking and tagging. In Karl Aberer, Key-Sun Choi, Natasha Noy, Dean Allemang, Kyung-Il Lee, Lyndon Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber, and Philippe Cudré-Mauroux, editors, *The Semantic Web*, volume 4825 of *LNCS*, pages 367–380. Springer, 2007.

[Oliver *et al.*, 2006] Nuria Oliver, Greg Smith, Chintan Thakkar, and Arun C. Surendran. Swish: semantic analysis of window titles and switching history. In *IUI '06: Proceedings of the 11th international conference on Intelligent user interfaces*, pages 194–201, 2006.

[Phelan *et al.*, 2009] Owen Phelan, Kevin McCarthy, and Barry Smyth. Using twitter to recommend real-time topical news. In *Proceedings of the third ACM conference on Recommender systems*, RecSys '09, pages 385–388, New York, NY, USA, 2009. ACM.

[Qiu and Cho, 2006a] Feng Qiu and Junghoo Cho. Automatic identification of user interest for personalized search. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 727–736, New York, NY, USA, 2006. ACM Press.

[Qiu and Cho, 2006b] Feng Qiu and Junghoo Cho. Automatic identification of user interest for personalized search. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 727–736, New York, NY, USA, 2006. ACM Press.

[Quan *et al.*, 2003] Dennis Quan, David Huynh, and David R. Karger. Haystack: A platform for authoring end user semantic web applications. In *Proceedings of ISWC*, 2003.

[Rattenbury and Canny, 2007] Tye Rattenbury and John Canny. Caad: an automatic task support system. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 687–696, New York, NY, USA, 2007. ACM.

[Rattenbury *et al.*,] Tye Rattenbury, Nathaniel Good, and Mor Naaman. Towards automatic extraction of event and place semantics from flickr tags. In *SIGIR '07*, Amsterdam, The Netherlands.

[Rhodes, 2000] Bradley James Rhodes. *Just-in-time information retrieval*. PhD thesis, 2000. Supervisor-Pattie Maes.

[Ronen *et al.*, 2009] Inbal Ronen, Elad Shahar, Sigalit Ur, Erel Uziel, Sivan Yogev, Naama Zwerdling, David Carmel, Ido Guy, Nadav Har'El, and Shila Ofek-Koifman. Social Networks and Discovery in the Enterprise (SaND). In *ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2009.

[Rose and Levinson, 2004] Daniel E. Rose and Danny Levinson. Understanding user goals in web search. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 13–19, New York, NY, USA, 2004. ACM.

[Rowlands *et al.*, 2007] T. Rowlands, D. Hawking, and R. Sankaranarayana. Workload sampling for enterprise search evaluation. *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 887–888, 2007.

[Sauermann, 2003] Leo Sauermann. The gnowsis – using semantic web technologies to build a semantic desktop. Diploma thesis, Technical University of Vienna, 2003.

[Schmitz, 2006] P. Schmitz. Inducing ontology from flickr tags. In *Proc. of the Collaborative Web Tagging Workshop (WWW 06)*, May 2006.

[Serdyukov *et al.*, 2007] Pavel Serdyukov, Sergey Chernov, and Wolfgang Nejdl. Enhancing expert search through query modeling. In Amati et al. [2007], pages 737–740.

[Serdyukov, 2009] Pavel Serdyukov. *Search for expertise : going beyond direct evidence*. PhD thesis, Enschede, June 2009.

[Shen and Dietterich, 2007] Jianqiang Shen and Thomas G. Dietterich. Active em to reduce noise in activity recognition. In *IUI '07: Proceedings of the 12th international conference on Intelligent user interfaces*, pages 132–140, 2007.

[Shen *et al.*, 2005] Xuehua Shen, Bin Tan, and ChengXiang Zhai. Implicit user modeling for personalized search. In *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 824–831, New York, NY, USA, 2005. ACM.

[Shen *et al.*, 2006] Jianqiang Shen, Lida Li, Thomas G. Dietterich, and Jonathan L. Herlocker. A hybrid learning system for recognizing user tasks from desktop activities and email messages. In *IUI '06: Proceedings of the 11th international conference on Intelligent user interfaces*, pages 86–92, 2006.

[Sigurbjörnsson and van Zwol, 2008a] Börkur Sigurbjörnsson and Roelof van Zwol. Flickr tag recommendation based on collective knowledge. In *Proceeding of the 17th international conference on World Wide Web*, WWW '08, pages 327–336, New York, NY, USA, 2008. ACM.

[Sigurbjörnsson and van Zwol, 2008b] Börkur Sigurbjörnsson and Roelof van Zwol. Flickr tag recommendation based on collective knowledge. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 327–336, Beijing, China, 2008. ACM.

[Soules and Ganger, 2005] Craig A. N. Soules and Gregory R. Ganger. Connections: using context to enhance file search. In *SOSP '05: Proceedings of the twentieth ACM symposium on Operating systems principles*, pages 119–132, New York, NY, USA, 2005. ACM.

[Tan *et al.*, 2006] Bin Tan, Xuehua Shen, and ChengXiang Zhai. Mining long-term search history to improve search accuracy. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 718–723, New York, NY, USA, 2006. ACM Press.

[Teevan *et al.*, 2004] Jaime Teevan, Christine Alvarado, Mark S. Ackerman, and David R. Karger. The perfect search engine is not enough: a study of orienteering behavior in directed search. In *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 415–422, New York, NY, USA, 2004. ACM.

[Teevan *et al.*, 2005a] Jaime Teevan, Susan T. Dumais, and Eric Horvitz. Personalizing search via automated analysis of interests and activities. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 449–456, New York, NY, USA, 2005. ACM Press.

[Teevan *et al.*, 2005b] Jaime Teevan, Susan T. Dumais, and Eric Horvitz. Personalizing search via automated analysis of interests and activities. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 449–456, New York, NY, USA, 2005. ACM Press.

[Teevan *et al.*, 2008] Jaime Teevan, Susan T. Dumais, and Daniel J. Liebling. To personalize or not to personalize: modeling queries with variation in user intent. In

SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, pages 163–170, New York, NY, USA, 2008. ACM.

[Teevan *et al.*, 2009] Jaime Teevan, Meredith Ringel Morris, and Steve Bush. Discovering and using groups to improve personalized search. In *WSDM '09: Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pages 15–24, New York, NY, USA, 2009. ACM.

[Toyama *et al.*, 2003] Kentaro Toyama, Ron Logan, and Asta Roseway. Geographic location tags on digital images. In *MULTIMEDIA '03: Proceedings of the eleventh ACM international conference on Multimedia*, pages 156–166, Berkeley, CA, USA, 2003. ACM.

[Vaidya *et al.*, 2005] Jaideep Vaidya, Yu Michael Zhu, and Christopher W. Clifton. *Privacy Preserving Data Mining (Advances in Information Security)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

[Vapnik, 1999] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, NY, November 1999.

[Voorhees and Harman, 2005] E. Voorhees and D. Harman. Trec experiment and evaluation in information retrieval. MIT Press, MA, USA, 2005.

[Voorhees, 2001] Ellen Voorhees. The philosophy of information retrieval evaluation. In *Proc. of the 2nd Workshop of the Cross-Language Evaluation Forum (CLEF)*, 2001.

[Wal, 2005] T. Vander Wal. Explaining and showing broad and narrow folksonomies, 2005. Available at http://www.personalinfocloud.com/2005/02/explaining_and_.html.

[Wal, 2007] Thomas Vander Wal. Folksonomy coinage and definition. Website, Februar 2007. http://vanderwal.net/folksonomy.html.

[Wang *et al.*, 2006] Jun Wang, Arjen P. de Vries, and Marcel J. T. Reinders. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 501–508, New York, NY, USA, 2006. ACM.

[White and Drucker, 2007] Ryen W. White and Steven M. Drucker. Investigating behavioral variability in web search. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 21–30, New York, NY, USA, 2007. ACM.

[Witten *et al.*, 1999] Ian H. Witten, Alistair Moffat, and Timothy C. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images, Second Edition.* Morgan Kaufmann, 1999.

[WS, 2011] Windows Search. http://www.microsoft.com/windows/products /winfamily/desktopsearch/default.mspx, 2011.

[Wu *et al.*, 2006] Xian Wu, Lei Zhang, and Yong Yu. Exploring social annotations for the semantic web. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 417–426, New York, NY, USA, 2006. ACM.

[Xu *et al.*, 2008] Shengliang Xu, Shenghua Bao, Ben Fei, Zhong Su, and Yong Yu. Exploring folksonomy for personalized search. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 155–162, New York, NY, USA, 2008. ACM.

[Yanbe *et al.*, 2007] Yusuke Yanbe, Adam Jatowt, Satoshi Nakamura, and Katsumi Tanaka. Can social bookmarking enhance search in the web? In *JCDL '07: Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*, pages 107–116, New York, NY, USA, 2007. ACM.

[Yang and Jeh, 2006a] Beverly Yang and Glen Jeh. Retroactive answering of search queries. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 457–466, New York, NY, USA, 2006. ACM Press.

[Yang and Jeh, 2006b] Beverly Yang and Glen Jeh. Retroactive answering of search queries. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 457–466, New York, NY, USA, 2006. ACM Press.

[Zerr *et al.*, 2010] Sergej Zerr, Elena Demidova, and Sergey Chernov. deskweb2.0: Combining desktop and social search. In *Desktop Search Workshop at SIGIR 2008*, pages 9–12, July 2010.

[Zerr *et al.*, 2011] Sergej Zerr, Kerstin Bischoff, and Sergey Chernov. Guideme! the world of sights in your pocket. In Serge Abiteboul, Klemens Böhm, Christoph Koch, and Kian-Lee Tan, editors, *ICDE*, pages 1348–1351. IEEE Computer Society, 2011.

[Zhou *et al.*, 2008] Ding Zhou, Jiang Bian, Shuyi Zheng, Hongyuan Zha, and C. Lee Giles. Exploring social annotations for information retrieval. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 715–724, New York, NY, USA, 2008. ACM.