Congenial Web Search

A Conceptual Framework for Personalized, Collaborative, and Social Peer-to-Peer Retrieval

Dissertation
zur
Erlangung des Doktorgrades (Dr. rer. nat.)
der
Mathematisch-Naturwissenschaftlichen Fakultät
der
Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von
Melanie Gnasa
aus
Koblenz

Bonn
2006

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Rheinischen Friedrich-Wilhelms-Universität Bonn. Diese Dissertation ist auf dem Hochschulschriftenserver der ULB Bonn `http://hss.ulb.uni-bonn.de/diss_online` elektronisch publiziert.

1. Referent: Prof. Dr. Armin B. Cremers, University of Bonn

2. Referent: Prof. Dr. Douglas Oard, University of Maryland, College Park

Tag der Promotion: 18.9.2006

Erscheinungsjahr: 2006

Contents

Acknowledgement

Being at the right place at the right time has always been essential for me in private as well as in academic concerns. This dissertation arose from many thoughts which I learned to combine in the past four years. During this time, I appreciated meeting people who supported me, guided me, and encouraged me in my research ’adventure’; thanks to all of you who helped me to make my research vision come true.

I would like to express my deepest appreciation to Prof. Dr. Armin B. Cremers, who not only served as my research supervisor, but also constantly gave me support and reassurance. I am grateful to Prof. Dr. Douglas Oard, who co-advised my work, providing valuable and countless resources, insight, and inspiration. Special thanks goes to Prof. Dr. Andreas Weber and PD Dr. Bernhard Schröder for participating so actively on my committee.

I am grateful to the members of the institute of computer science III, namely Sascha Alda, Julia Kuck, Dr. Uwe Radetzki, Dr. Jens E. Wolff, and Dr. Markus Won for their suggestions and support whenever I asked for it.

My research has further benefited from helpful comments and suggestions from Prof. Dr. Karin Harbusch who has always had time for my questions. In particular, I am much obliged to Dr. Jens Woch who has always been a strict and patient research reviewer for me.

Further on, I would like to thank all graduate students, namely Jasmin Grigull, Nadir Gül, Sebastian Marius Kirsch, Jens Rinne, Frank Reichartz, Thorsten Ruhl, and Inna Tichman for their help to implement a prototype that enabled me to show the results of this work.

Thanks to Ulrich Helsper, Joanne Mothes, Gunder-Lily Sievert, and Friederike Wolfrum for helping me with the details of the English language.

I want to thank numerous researchers for their intelligent and challenging comments on my dissertation at various workshops. I want to especially thank Prof. Dr. Bruce Croft and Prof. Dr. Alistair Moffat for their valuable comments at the SIGIR doctoral consortium in 2004. They advised me to focus on the essential parts of my research project to make it more manageable for evaluation.

Finally, I am very grateful to my husband, my parents, and my friends for their encouragement and patience.

Abstract

Traditional information retrieval methods fail to address the fact that information consumption and production are social activities. Most Web search engines do not consider the social-cultural environment of users' information needs and the collaboration between users. This dissertation addresses a new search paradigm for Web information retrieval denoted as Congenial Web Search. It emphasizes personalization, collaboration, and socialization methods in order to improve effectiveness.

The client-server architecture of Web search engines only allows the consumption of information. A peer-to-peer system architecture has been developed in this research to improve information seeking. Each user is involved in an interactive process to produce meta-information. Based on a personalization strategy on each peer, the user is supported to give explicit feedback for relevant documents. His information need is expressed by a query that is stored in a Peer Search Memory. On one hand, query-document associations are incorporated in a personalized ranking method for repeated information needs. The performance is shown in a known-item retrieval setting. On the other hand, explicit feedback of each user is useful to discover collaborative information needs. A new method for a controlled grouping of query terms, links, and users was developed to maintain Virtual Knowledge Communities. The quality of this grouping represents the effectiveness of grouped terms and links. Both strategies, personalization and collaboration, tackle the problem of a missing socialization among searchers.

Finally, a concept for integrated information seeking was developed. This incorporates an integrated representation to improve effectiveness of information retrieval and information filtering. An integrated information retrieval process explores a virtual search network of Peer Search Memories in order to accomplish a reputation-based ranking. In addition, the community structure is considered by an integrated information filtering process. Both concepts have been evaluated and shown to have a better performance than traditional techniques. The methods presented in this dissertation offer the potential towards more transparency, and control of Web search.

1 Introduction

We live in an age of information overload, an age bringing forth a yearly production of print, film, optical, and magnetic contents requiring roughly 1.5 billion gigabytes of storage (Lyman and Varian, 2000). An advanced navigation strategy is indispensable for overall user satisfaction. A look at today's Web search capabilities reveals that the Web is still in an embryonic state with limited search facilities.

Observation 1: Information providers are dependent on Web search engines, which give access to their information. Due to the ongoing competition among Web search providers, having only one main search engine would bring about the risk of a monopolization of information access.

Observation 2: Web search engines do not use transparent relevance assessments. No user-specific ranking functions are considered in order to utilize an individual relevance measure for each user.

Observation 3: Web search engines do not assist awareness of similar search interests of other users. Due to a system-centered design, collaborative retrieval strategies based on validated results of other users are not shared among users.

Observation 4: Web search engines do not foster socialization among their users in order to facilitate the reliability of document contents and user recommendations. No mediation of information based on communities is performed in today's pure reference-based retrieval systems.

All four observations identify shortcomings of the current Web search reality. Each observation reveals the need of a more user-centered design of a search engine. From the very first, Tim Berners-Lee's concern had been to build a communication system that connects people as well as machines. Today, machines do not have the ability of semantic document processing. Hence, a new contribution to the future will be the 'Semantic Web', an ambitious project that surpasses the original ideas of the World Wide Web. The particular contribution of this dissertation is a new search paradigm which incorporates the traditional roles of users and Web search engines in a decentralized manner. Instead of introducing a new retrieval model, this dissertation focuses on new cooperation issues among all users. They include advanced Web search techniques. The new paradigm for Web search is denoted as *congenial*.

1.1 Goal of Congenial Web Search

The main goal of *Congenial Web Search* is to bring people together in order to address information needs they have in common. The underlying search paradigm of Congenial Web Search expects users to rethink the familiar Web search idea. It aspires a continuous interaction among users in order to improve retrieval effectiveness. Individuals are not only 'users' of the system, but they are part of an information seeking process that combines information retrieval and information filtering. In order to bridge the gap between dynamic information sources and the variability of information needs, all users contribute to the system.

Congenial Web Search does away with the established role of client (user) and server (search engine). All individuals are not only clients of the search engine, they are treated as providers, also. Each user maintains a distributed search service in cooperation with a Web search engine. For both roles, two central requirements are defined:

Transparency Each process of Congenial Web Search must be transparent to the user. He must be aware of all information providers that have contributed to his search result.

Visibility Each user must be visible to other information consumers in order to share common interests. Furthermore, the interaction must be enhanced for a group of users with long-term information needs.

1.2 Towards Congenial Web Search

The success of a new service that provides users with access to diverse information sources depends on an effective information seeking process (Loeb and Terry, 1992). Several conceptual models for information seeking exist and are discussed by (Belkin et al., 1995), (Ellis, 1989), (Ingwerson, 1996), (Järvelin and Wilson, 2003), and (Kuhlthau, 1991). Such models assist the design of an information system corresponding to the users' needs. Information seeking begins when someone realizes that his current knowledge about a subject is inadequate. The information seeking process ends when the perceived need has been satisfied. In general, each information seeking process consists of three subtasks: collecting information sources, detecting useful information sources, and displaying them (Oard, 1997). In particular, the goal of an information (retrieval) system is to provide the user with information from the knowledge resource helping him in problem management (Belkin, 1984). Belkin and Croft (1992) described retrieval of documents from an archival collection and filtering documents from an incoming stream of documents as two sides of the same coin. This dissertation combines both processes in a user-centered manner. Each user contributes to the system with local relevance feedback enhancing personalized information sources. Owing to a collaboration among all users, ad-hoc communities can be identified for common interests. The interaction between users or Web search engines forms a *virtual search network* of personalized information sources. The interweaving of information retrieval and filtering in such a network is derived in two steps:

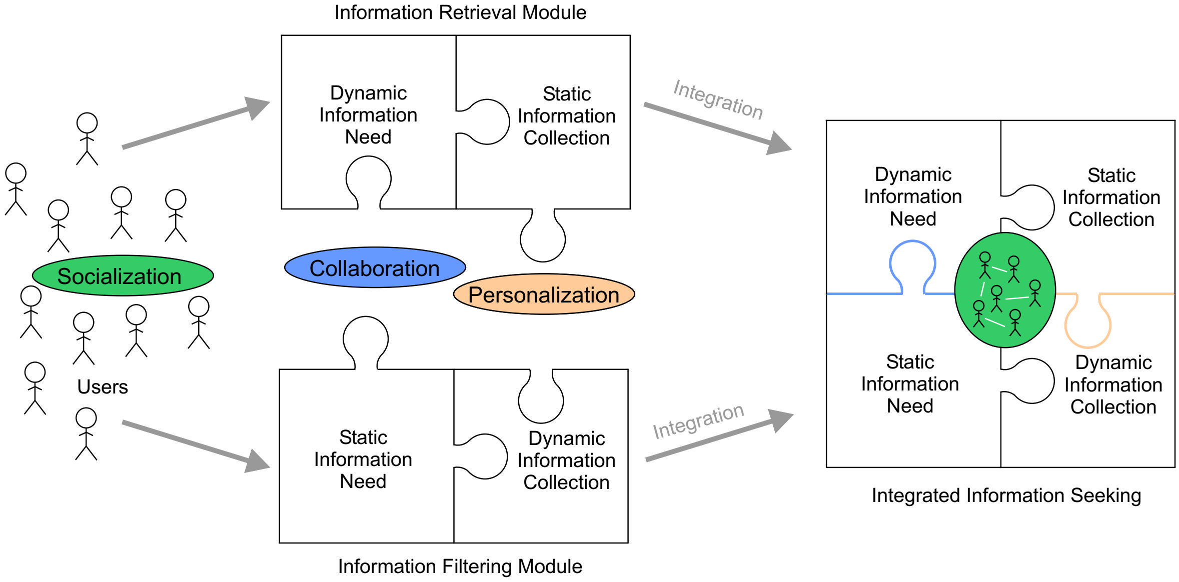

Information Retrieval Module
Dynamic Information Need
Static Information Collection
Integration
Socialization
Collaboration
Personalization
Users
Static Information Need
Dynamic Information Collection
Integration
Information Filtering Module
Dynamic Information Need
Static Information Collection
Static Information Need
Dynamic Information Collection
Integrated Information Seeking

Figure 1.1: Roadmap towards Congenial Web Search

1st Step - Congenial Web Search integrates users who maintain dynamic information collections. Each information need, and the retrieved relevant document are used to build a *Personalized Search Memory* with validated results.

2nd Step - Congenial Web Search comprises a networking of users based on common search interests. All users occupy two roles in the virtual search network, information consumer and information provider.

Figure 1.1 describes the roadmap towards Congenial Web Search. This approach relies on personalization, collaboration, and socialization techniques in a user-centered architecture. Each technique offers important advantages:

Personalization A personalized search can be maintained either on the server or client-side. The user benefits from a personalization on the client-side twofold:

- A personalization strategy on the client-side is independent of a search service.
- Search sessions with different providers are represented equally on each local machine.

The main task of the personalization component is a uniform access to data services and a collection of relevance feedback. The problem of ‘Keeping Found Things Found’ (Jones et al., 2001) is addressed by a novel approach to combine local relevance feedback with results from a Web search engine. An exploratory data analysis on a Weblog corpus confirmed repetitions among search sessions. These repetitions concern both types of duplicates, queries and link views. If a user requests a similar query regarding his prior interests, a personalized ranking assists the fusion of well-known and new results. The effectiveness of this approach has been shown with an evaluation of usage data. In summary, the personalization component defines a single-user strategy to build a personalized information collection. No other users and their relevance feedback are taken into account in this strategy.

Collaboration A collaboration among users during Web search is based on communities which represent validated information collections. It is a great challenge to discover common search interests transparently in order to maintain virtual communities. In a user-centered system design, the discovery of common search interests controlled by the user's actual interest has two benefits:

- Novice searchers profit from validated results of a user group.
- Users are organized according to common, long-term interests.

The main task of the collaboration component is to discover users with common search interests. Similar relevance assessments are exploited to assist the grouping of terms, documents, and users. The collaboration strategy is independent of the information seeking process. A novel representation of collaborative information needs supports dynamic interests, as well as static ones. Each relevance feedback assessment initiates a discovery process. The grouping of users, terms, and documents was shown to have a high quality. The community administration does not rely on a central instance. It is managed by all members in a self-organized manner. Users are organized in a community if its topic is of long-term interest for them.

Socialization Social relationships or social networks are useful for finding information. The collaboration component groups users without measuring how much someone contributes to a community. The assignment of social ranks to users has two benefits:

- All information providers are assessed by their importance for other members.
- Information consumers profit from a reputation-based ranking.

The main task of the socialization component is to explore the Virtual Search Network and to assign social ranks. The main challenge of a reputation management within this network is to preserve the anonymity of users. Owing to the personalization and collaboration component, Congenial Web Search relies on a common representation of information needs and collections (see Figure 1.1). A novel interaction-based strategy was developed to compute global authority scores for each user. Personalization, collaboration, and socialization techniques are the basis for an integration of both information seeking processes, information retrieval and information filtering. Users with short-term interests, as well as long-term interests are assisted by an integrated information seeking system.

Integrated Information Seeking A cooperative pull-push cycle is a general model for all interactions in the virtual search network. This network continuously grows by two primary user actions. On one hand, each client consumes information in the network by an *information pull*. On the other hand, each client is a provider of relevance assessments which are recommended to other users by an *information push*. Figure 1.2 shows a flow diagram of a distributed search with an integrated information seeking (IIS) system. In general, the integrated information seeking process is similar to an information retrieval process. It satisfies

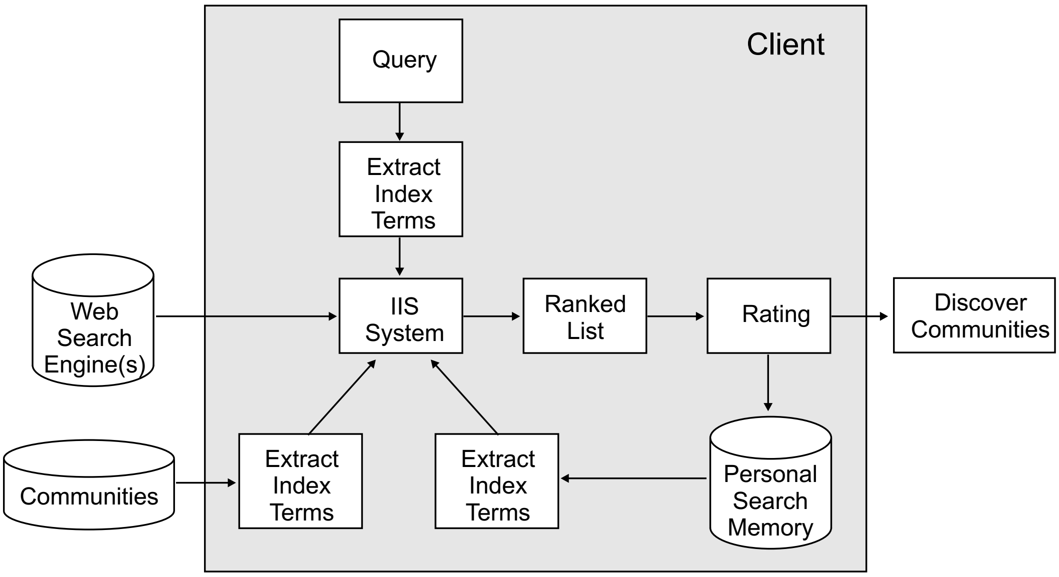


Figure 1.2: Flow Diagram of a Distributed Integrated Information Seeking System

a user's information need through retrieving a set of items that contain the desired information. The main difference arises from the combination of a retrieval and filtering process. The user's information need can be either a short-term or a long-term one. Dependent on the type of information need, the expression of the information need as a query is done by the user or by the system. System-generated queries rely on a user-specific search profile that associates terms with joined communities. The integrated information seeking (IIS) system can be broken down into the following functionalities:

1. The IIS system localizes documents that are possibly relevant to the queries. Only for the local search, features are extracted from queries, documents, as well as communities. In addition, the query is propagated to a Web search engine.

2. The IIS system ranks the documents of all data services depending on their relevance. The ranking function considers the personal search memory, as well as the users' reputation that is calculated the socialization strategy.

3. The IIS system collects user ratings for relevant documents. These documents are stored according to the personalization strategy in a Peer Search Memory. Furthermore, relevance assessments are used to discover communities according to the collaboration strategy.

The main challenge of the system design is the implementation of a unified framework for information pull and information push. An information pull routes a query to different data services. All clients process the query in a transparent manner. A selective usage of specific data services for retrieval or filtering is more effective than the present state of the art methods.

1.3 Research Contributions

In the current Web, the roles of providers and consumers are too strict. To avoid a system-centered design, a flexible framework is necessary. The research contributions have been broken down into seven claims:

1. *Model of personalized information sources.* A personalized information source harnesses collective information about all search sessions. Associations between queries and documents of a user build a fundamental user profile for all information seeking processes. The effectiveness of a personalized ranking strategy is examined with usage data.
2. *Architecture (P2P) of cooperating data services*. Each user maintains a data service in a network of cooperating services. The new architecture shares individual query-document associations among all peers. In addition, external data services of Web search engines are integrated in a personalized information retrieval process.
3. *Process for forming 'communities'.* A novel grouping approach discovers similarities among users' associations. Users can confirm their long-term interest of a specific topic with a membership to a community (Gnasa et al., 2003). The quality of the grouping was evaluated based on terms and links. An automated methodology was developed to measure link quality.
4. *User networking based on replication of associations.* Replications between peers possess quality assessments of information providers. They are utilized for two user-centered evaluation measures:
   - Global Reputation: Relevant information that has been provided to others leverages a global authority measure. The effectiveness was examined for a reputation-based ranking (Kirsch et al., 2006).
   - Community-specific Trust: The trust of a user in the recommendations of another user is a local measure between two peers. The effectiveness was examined for community-based filtering.
5. *Unified model for information seeking*. The unified search model is a novel integration of two information seeking processes, information retrieval and information filtering (Gnasa et al., 2004a). The interaction among data services are independent of the process type.
6. *Community-specific selection of recommendations.* Communities are used as a novel source for recommendations to decrease the number of peer requests. The new method is as effective as a state of the art retrieval method, but it is more efficient for a distributed environment without a central index.
7. *Open source reference implementation.* In order to benefit from a collective adaption, an open-source reference implementation ensures re-use and experimentation. ISKODOR (Gnasa et al., 2004a,b) is a prototype following existing standards for peer-to-peer applications.

1.4 Outline

The remaining part of this dissertation is structured as follows:

In Chapter 2, prior research on information retrieval and information filtering are surveyed, with the focus on personalization, collaboration, and socialization techniques. It also lists related systems.

Chapter 3 presents the concept requirements, and defines the architecture for Congenial Web Search. The system design is based on a peer-to-peer application framework that uses existing standards for network communication.

Chapter 4 describes the local storage of explicit relevance feedback with a Peer Search Memory. This chapter introduces a new client-based personalization for repeated information needs. A Weblog corpus was processed to explore users' search behavior, and to evaluate the effectiveness of the personalized ranking strategy.

Chapter 5 defines a novel concept to model communities by leveraging personalized information sources introduced in Chapter 4. Each user initiates the discovery process, and a controlled grouping method finds similar terms, documents, and users. The new community method is employed in improving information retrieval and filtering. A test corpus of log data was built to simulate community discovery.

Chapter 6 presents a novel approach for interweaving information retrieval and information filtering processes. This approach was used in conjunction with the new personalization and collaboration strategies. They are employed in improving authority-based retrieval and community-based filtering.

Chapter 7 describes the prototype implementation. Open-source components are integrated in order to reduce development time.

Lastly, Chapter 8 concludes the dissertation by discussing its limitations, its future work, and its impact.

2 Literature Review

This dissertation draws its inspiration from a number of different sources, and covers current and emerging trends in information retrieval and related fields. This chapter contains historical remarks about the World Wide Web and its relation to the information retrieval history. It reviews related concepts and systems for information seeking, personalization, collaboration, and socialization.

2.1 Historical Remarks

All predictions about future developments in the field of computer science beyond the next 10 years may be looked at as science fiction. Some phenomena, however, that were already observed sixty years ago, have become reality meanwhile or will come true within the next decade. According to Umberto Eco's (1996) pessimistic perspective of the future, a three-class society will develop with a large lower class, a proletariat without computer access, a middle class, the 'bourgeoisie', formed by people who use the computer only passively (e.g. by checking flight availability), and an upper class, the 'nomenclature', whose members know how to use a computer according to their needs, and are able to keep up with technological progress. But even this chosen minority is threatened: faced with a flood of information, it does no longer know what to select.

> If someone offered us one billion dollars, but on the condition that we counted out only one-dollar bills, we'd rather not accept. Supposed we needed one second per dollar, this operation would take thirty-one years. [...] Fortunately, we can accept a check for one billion dollars and that settles it. But with a voucher for one billion information units of the new technologies, things would not be settled.[1]

Today, technological progress approaches large information loads and fast network speeds. On the other hand, it is not possible to achieve a proportional increase of the human capacity load. That is why new tools need to be developed that keep pace with the rapid growth of digital information sources. The history of information retrieval can be traced back to approximately 2000 B.C. when the Sumerian literary catalogue was probably the first list of books ever written. In modern times, the popularization of the idea of information retrieval started in 1945 (Lesk, 1995). Vannevar Bush's 1945 article 'As we may think' (Bush, 1945)

[1] This citation was originally published in German (Eco, 1996).

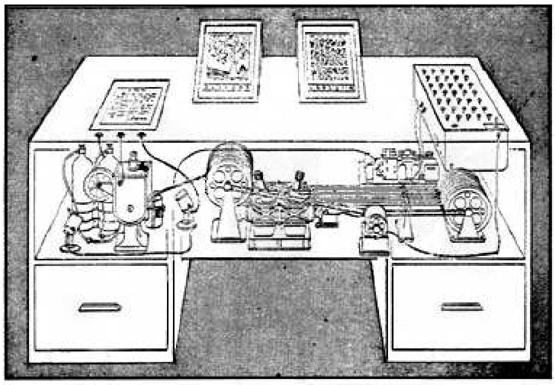


Figure 2.1: Vannevar Bush’s Memex

presented a vision of fast access to the contents of the world’s libraries. His vision has been a technical inspiration for many other researchers ever since, although the core of that vision has not been realized yet.

Vannevar Bush’s historical influence is sometimes forgotten or misunderstood. As the historian Michael Sherry said, "To understand the world of Bill Gates and Bill Clinton, start with understanding Vannevar Bush" (Zachary, 1997). Bush’s article reflected on how new technology could help to solve the problems of the post-war society. He envisioned a revolutionary personal information machine, which is why the cutting-edge of computer science refers to him as the godfather of the information age. At the end of World War II, computer science was still in its infancy, and Bush was particularly concerned about the explosion of scientific information. As one major requirement of a data record, he identified the need for a continuous *extension, storage*, and *retrieval* facility. Bush proposed a device that was to help individuals retrieve and store all essential human knowledge, as well as their own specific memories. Bush called this device a **‘memex’**, a *memory extender*. Figure 2.1 depicts the machine Bush proposed, which is part computer, part microfiche, and part database. The integrated information retrieval system used an associative method of information selection rather than an artificial index. Bush was aware that the human mental process could not be artificially duplicated, but that researchers ought to be able to learn from this process. A memex should enable the user to consult books, records, and communications in an efficient and flexible way. This goal can be achieved by an associative indexing, where the choice of a keyword initiates the selection of other entries associated with this term. For Bush, this process is realized by trails of associations which link terms, and which can manually be supplemented with new terms and associations. Owing to an aggregation with other memex contents of friends or authorities, Bush advocated new forms of encyclopedias. As a whole they would have to be much larger than the sum of their collective parts. Bush’s visionary article ends with the words:

> Yet, in the application of science to the needs and desires of man, it would seem to be a singularly unfortunate stage at which to terminate the process, or to lose hope as to the outcome.

Since the publication of Bush’s visionary article, other pioneers have been inspired by his work. In the course of the information age, his ideas have been enhanced and now, they can

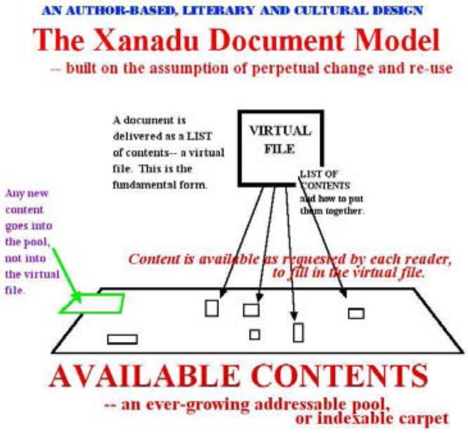

AN AUTHOR-BASED, LITERARY AND CULTURAL DESIGN
The Xanadu Document Model
-- built on the assumption of perpetual change and re-use
A document is delivered as a LIST of contents-- a virtual file. This is the fundamental form.
VIRTUAL FILE
LIST OF CONTENTS and how to put them together.
Any new content goes into the pool, not into the virtual file.
Content is available as requested by each reader, to fill in the virtual file.
AVAILABLE CONTENTS
-- an ever-growing addressable pool, or indexable carpet

Figure 2.2: Ted Nelson's Xanadu Model

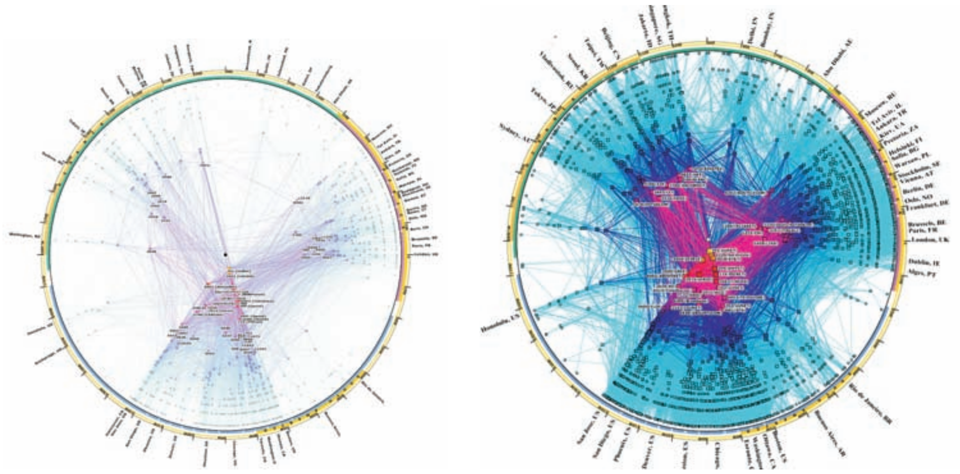
be found in nearly every computer science discipline. For example, the memex directly influenced and inspired the two researchers generally credited with the invention of hypertext, Douglas Engelbart and Ted Nelson. Even though the memex cannot be considered a true hypertext system, the major history of hypertext starts with it.

During the 1960s, Engelbart designed the **'oNLine System' (NLS)** (Engelbart, 1962), which was a revolutionary computer collaboration system. The tools he applied have been trend-setting for today's personal computers, which still define the current standards. Engelbart, whose team developed a linkage among heterogeneous data by pointers, is best known as the inventor of the computer mouse, and as a pioneer in the field of human-computer interaction. With NLS computer-interface elements, such as bit-mapped screens, multiple windows, groupware, hypertext, and precursors of graphical user interfaces had been developed long before the personal computer revolution (Engelbart and English, 1968).

The term 'hypertext' was coined by Nelson in 1965 (Wedeles, 1965). He founded **'Project Xanadu'** (Nelson, 1965) in 1960 with the goal to develop an authoring and browsing system. Due to the simultaneous and collective editing of a document, the differentiation between author and reader can be balanced. Today, the final aim of this project, the maintenance of the world's knowledge by use of a computer-supported concept network which organizes the access to a particular information item, still appears utopian. Nevertheless, with Xanadu a document management tool was conceived which allows automatic version management and rights management. The Xanadu model as depicted in Figure 2.2[2] assists content availability in a simple fashion. Based on a distributed storage, each document is maintained as a virtual file consisting of a list of contents. This capability of documents to include sections of other documents by reference is called *transclusion*. Such a model foresaw world-wide hypertext decades ago, but only a shallow structure led to the success of the World Wide Web. The Xanadu project itself failed to take off for a variety of controversial reasons.

In addition to the projects of Nelson and Engelbart, a number of experimental hypertext systems have been developed, but none of these systems achieved widespread success with a large interest community. All the earlier hypertext systems have quickly been outdated by the success of Tim Berners-Lee's **'World Wide Web'** (Berners-Lee, 1989) since 1989. This

[2] `http://xanadu.com/xuTheModel/`, last visit on 2006/03/01.



(a) Jan 15 - Jan 22, 2000 (b) Apr 4 - Apr 17, 2005

Figure 2.3: Macroscopic Snapshot of the Internet for Two Weeks by CAIDA

model lacks many features of the earlier systems such as typed links, transclusion and source tracking, which makes it a gross over-simplification. However, the internet connectivity increases continuously over the last five years as visualized in Figure 2.3[3]. The strong linkage of the actual hypertext model proves that many of Project Xanadu's proposed features have found their way into other hypertext systems beyond the Web.

To summarize, one can say that since Bush, each of the visions presented has turned out to be a step towards his dream of a world-wide sharing of knowledge. From today's viewpoint, the hardware he proposed seems mostly out of date, which can be explained by the fact that he could not predict the rapid developments of digital technology. However, the goals in software development envisioned by him have not been achieved yet as a whole. In looking to the future, Shneiderman advocates that we might again transform society by building '**genexes**' - *generators of excellence* (Shneiderman, 1998). The main goal of such inspirational environments is to empower personal and collaborative creativity by a four-phase model. The first phase of this model, which will be the focal point of this dissertation, is the collection of information from an existing domain of knowledge. In the context of the World Wide Web a large amount of information is accessible, but the construction of useful knowledge with enhanced retrieval facilities is still a difficult task. To achieve this goal, researchers will have to overcome many unresolved problems.

2.2 Information Seeking Concepts

Collecting the information sources, selecting the information sources, and displaying the information sources are three subtasks which are applicable to a variety of information seeking processes (Oard, 1997). These tasks are fundamental to information retrieval processes, as well as information filtering processes. This section discusses techniques that facilitate information retrieval or information filtering. In addition, advanced retrieval techniques for peer-to-peer networks are presented.

[3] `http://www.caida.org/analysis/topology/as_core_network/`, last visit on 2006/08/02.

2.2.1 Information Retrieval

Information retrieval is an approach with a long research history. Frakes and Baeza-Yates (1992) identify six facets for the classification of an information retrieval system: conceptual model, file structure, query operations, term operations, document operations, and hardware. The most general facet is the conceptual model because it is essential for the design of a system.

Several taxonomies have been proposed for structuring the established conceptual models. Faloutsos (1985) identifies three basic approaches: *text pattern search*, *signature search*, and *inverted file search*. The first approach is the most straightforward way of locating the documents that contain a certain search string (Faloutsos and Oard, 1995). Algorithms for full text scanning can be found in several surveys, for example (Knuth, 1973). The advantage of these methods is that they require no space overhead and minimal effort regarding insertions and updates. For large information sources, such methods are not very efficient and have a bad response time (Faloutsos and Oard, 1995). This disadvantage is also observed for signature file approaches using hashing (Knott, 1975) and superimposed coding (Faloutsos, 1985). A fast retrieval can be achieved by inverted file search, and is used by almost all commercial systems (Salton and McGill, 1983). Each document is represented by a list of keywords and for each keyword a list of pointers to the corresponding documents is maintained (Faloutsos and Oard, 1995). An alternative categorization of the conceptual models is proposed by Belkin and Croft (1987). In a first step, they divide retrieval techniques into *exact match* and *inexact match*. For a detailed survey on these models see (Salton and McGill, 1983) or (Belkin and Croft, 1987).

Text pattern search and *Boolean search* techniques are associated with the exact match category. Both models retrieve all documents with an exact match of the query with one or more text surrogates. Each query specifies precise retrieval criteria and the result is a set of documents. The unranked Boolean retrieval model is the most common exact match model. All documents are retrieved that satisfy a Boolean expression. Based on Boolean logic, standard operators (e.g. `AND`, `OR`, or `NOT`) are combined with terms or phrases to express a query. Documents are returned in no particular order, and the information collection is partitioned into a set of retrieved documents and a set of not-retrieved documents (Belkin and Croft, 1992). On one hand, this model has the advantage of an efficient query processing for large document collections. On the other, the major problem is the absence of any form of relevance ranking of the retrieved document set. Further problems have been discussed by several authors (see (Bookstein, 1985), (Cooper, 1988), (Frants et al., 1999)).

The observation that some objects are more likely to be relevant or more relevant to an information need than others led to the proposal of inexact match models. For this type of models, each query describes a retrieval criteria for the desired documents. Every document matches a query to some degree and the result is a ranked list of documents. With these models, the user has control over the size of the output and is assisted in managing large result sets (Marchionini, 1995). Recently, many approaches for inexact models have been developed. The major representatives of this category are the following: *fuzzy-set, vector-space, probabilistic, inference network, and clustering*. In this dissertation, the main emphasis is on the vector-space model that is used by many information retrieval systems, as well as information filtering systems.

The *vector-space model* (Salton and McGill, 1983) has been widely used in the traditional research community of information retrieval. Salton (1971) chose this model as a basis for the SMART system. Each document is encoded as a vector, where each vector component reflects the importance of a particular term in representing the semantics or meaning of that document (Berry et al., 1999). For a specific information collection, a t-dimensional vector is generated for each document and each query from sets of terms with associated weights, where t is the number of unique terms in the document collection. The information collection containing a total of d documents is represented as a $t \times d$ *term-by-document matrix* A. The value assigned to a term is typically a function of the frequency with which the term occurs in the document and in the document collection as a whole (Sparck Jones, 1972). These two factors are multiplied together with a length normalization factor to compute the resulting term weight. Thus, the matrix element a_{ij} is the weighted frequency at which term i occurs in document j (Berry et al., 1992). For example, the weights can be calculated based on the following two numbers: (1) term frequency, f_{ij}, the number of occurrence of term x_i in document y_j; (2) inverse document frequency, $g_i = log(N/d_i)$, where N is the total number of documents in the collection, and d_i is the number of documents containing term x_i (Li et al., 2002). The number f_{ij} is a local weight that reflects the importance of term x_i within document y_j itself. The range of local weights differ in their complexity from simple binary values to functions involving logarithms of term frequencies (Berry et al., 1999). The inverse document frequency belongs to the class of global weighting schemes. These schemes range from simple normalizations to advanced statistical-based approaches (see (Dumais, 1991), (Sparck Jones, 1972)).

In an information retrieval system using the vector-space model, a query is represented as a set of terms, perhaps with weights, represented like a document. The goal of query matching is to find the documents most similar to the query in usage and weighting of terms. For the results, those documents are selected that are geometrically closest to the query according to some measure. This similarity can be defined by different measures. Additional information about this topic is available in several surveys, e.g. (van Rijsbergen, 1979), (Salton and McGill, 1983), and (Zobel and Moffat, 1998). One common measure of similarity is the cosine of the angle between the query and the document vectors. An important assumption of the vector-space model is that terms are independent, i.e., the dimensions of the space are orthogonal. This is a first approximation, but the assumption that words are pairwise independent is not realistic (Foltz and Dumais, 1992) and leads to the development of enhanced models. These models include several statistical and AI techniques capturing term associations and domain semantics. For example, *latent semantic indexing* (LSI) is one of these methods which are extensions of the standard vector-space model. More details of this model are presented by Deerwester et al. (1990), Berry et al. (1992), and Hofmann (1999).

The traditional *probabilistic model* was first introduced in 1976 by Robertson and Sparck Jones (1976). This model became later known as the *binary independence retrieval* (BIR) model. The probabilistic approach is based upon direct application of the theory of probability to information retrieval systems (Kowalski, 1997). This model is based on the idea that for a user query a set of documents exists containing exactly the relevant documents and no other. If a description of this ideal answer set exists, there is no problem in retrieving its documents. Thus, the querying process can be interpreted as a process of specifying the properties of an ideal answer set (Baeza-Yates and Ribeiro-Neto, 1999). The problem with such a process is that the properties (index terms) characterizing the semantics of a document are unknown at query time. From this problem the fundamental assumption arises describing the *probability ranking principle*. With this principle documents are ranked in the order of their probability of relevance to the query. A well-known probabilistic weighting scheme is the Okapi BM25 formula (Robertson et al., 1995).

2.2.2 Information Filtering

Filtering of information is a concept which is not limited to electronic documents (Foltz and Dumais, 1992). Information filtering systems select documents from a dynamic text stream to satisfy a relatively stable and specific information need. Information filtering combines many processes that are responsible for the selection of information. ‘Filtering’ is a frequently used term, and a distinction from other processes such as information retrieval, routing, categorization, or information extraction is often not clearly defined (Belkin and Croft, 1992). For further details on the differentiation between other processes see (Belkin and Croft, 1992) or (Oard, 1997). One of the earliest works on information filtering is known as ’Selective Dissemination of Information’ (SDI) (Houseman and Kaskela, 1970). This technique was integrated into Luhn’s Business Intelligence System (Luhn, 1958) in order to recommend new documents to scientists published in their areas of expertise. While SDI was implemented on a large scale, it was used far less than predicted (Packer and Soergel, 1979). In the eighties, most of the attention was focused on generating information. Denning (1982) advocated to focus more attention on receiving information. This process includes the controlling and filtering of information in order to prevent an unwanted reception. Belkin and Croft (1992) identify three major characteristics that are essential to the information filtering process. First, an information filtering system is an information system for unstructured or semi-structured data. Second, such a system processes large amounts of data from streams of external information sources. Third, the information need of an individual or a group is described with a profile including long-term interests.

Information filtering techniques have been applied to several areas of applications. For a survey of these areas see (Baudisch, 2001). The following list presents filtering applications of the following domains:

- *Usenet News*: InfoScope (Fischer and Stevens, 1991), GroupLens (Resnick et al., 1994), SIFT (Yan and Garcia-Molina, 1995), BORGES (Smeaton, 1996), NewsSieve (Haneke, 2001)
- *Electronic Mail*: InformationLens (Malone et al., 1987), Tapestry (Goldberg et al., 1992)
- *Web Pages*: WebWatcher (Joachims et al., 1997), Fab (Balabanovic and Shoham, 1997), Select (Alton-Scheidl et al., 1999)
- *Movies, Music*: MovieLens (Miller et al., 2003a), Ringo (Shardanand and Maes, 1995)

Aside from the differences between information retrieval and information filtering, many techniques originally developed for text retrieval can be modified to support the filtering

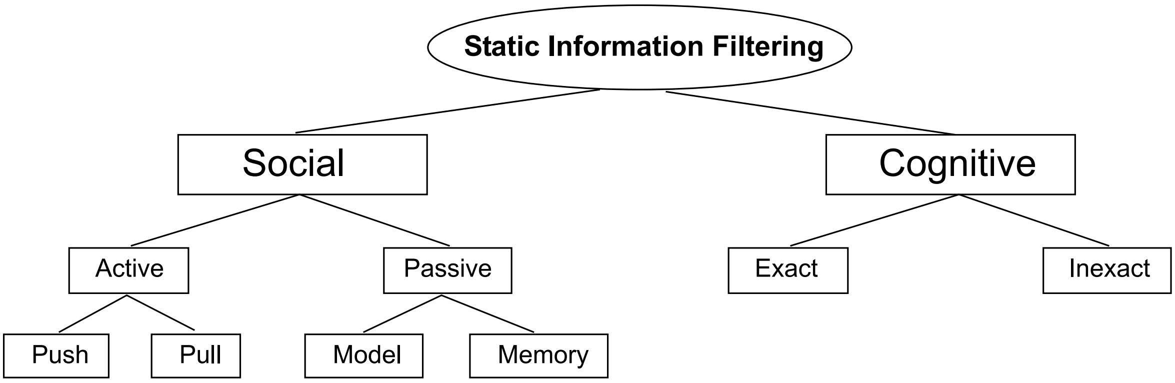


Figure 2.4: Classification of Information Filtering Approaches

process. Malone et al. (1987) refer to such techniques as *social*, *cognitive*, or *economic* approaches to information filtering. The two main research paradigms are cognitive and social filtering. Cognitive filtering is also denoted as *content-based filtering* by several authors, e.g. (Baudisch, 2001) or (Oard, 1997). The research heritage of cognitive filtering has its roots in the information retrieval community, and many of its techniques are employed in information filtering systems. Cognitive filtering approaches characterize the content of a message and the information needs of potential message recipients and then use these representations to match messages to receivers (Malone et al., 1987). This approach underlies the assumption that the meanings of objects and queries are captured in specific words or phrases (Marchionini, 1995). Specific models have been developed in information retrieval for this task, and Figure 2.4 depicts the major alternatives in regard to the conceptual models for information retrieval. Beside the vector-space model as a major approach to document filtering, for example, inference networks are used for document filtering (see (Callan, 1996)).

Pure content-based approaches have several shortcomings. First, only a shallow analysis of certain kinds of content can be supplied (Balabanovic and Shoham, 1997). For some applications, it is not possible to extract features of items such as movies or music. Even with regard to text documents, the discussion of the limitations of existing cognitive models captured only certain aspects of the content. Second, systems have an over-specialization (Balabanovic and Shoham, 1997) due to the restriction of recommended items scoring highly against a user's profile. Third, a common problem of most information filtering systems is getting user feedback. It is an onerous task for users to rate documents. On one hand, the fewer ratings are required the better is the user acceptance. One the other, user ratings are the only factor influencing the performance of future recommendations. Hence, the performance of pure content-based systems depends on the quantity of feedback information. Further details of relevance feedback are discussed in Section 2.3.1.

To overcome the shortcomings of pure content-based approaches, *social filtering* techniques have been proposed. These techniques are also known as *collaborative filtering* (see (Malone et al., 1987), (Goldberg et al., 1992)). In this dissertation, the term collaborative filtering is used denoting the following description. This type of filtering approach automates the social process known as 'word of mouth' (Shardanand and Maes, 1995). In our society people rely on recommendations from other people either by word of mouth, recommendation letters, movie or book reviews, or general surveys (Baudisch, 2001). The automation of this process, in turn, relies on the fact that people's tastes are not randomly distributed. It has been observed that general trends and patterns exist within the taste of a person, as well as within a group of people. As depicted in Figure 2.4, social filtering is classified into active and passive approaches. *Active collaborative filtering* (Maltz and Ehrlich, 1995) builds on the common practice where people tell their friends or colleagues about interesting documents. This approach covers the active behavior of a user who finds and evaluates a document to share that knowledge with particular people. Furthermore, active collaborative filtering is classified as either *push active* or *pull active* (Baudisch, 2001). This differentiation depends on whether the system selects the recipients, or recipients select recommenders.

Passive collaborative filtering approaches are well suited to situations where users benefit from the aggregation of votes of many users. These approaches are called 'passive' or 'automated' (Herlocker et al., 2000) because no direct connection between a user casting a vote and users filtering documents based on these aggregated votes exists. A significant distinction to active collaborative filtering is that instead of a referral network (Kautz et al., 1997), which must be maintained in the users's minds, thousands of users and thousands of different items can be considered (Shardanand and Maes, 1995). To summarize, passive collaborative filtering consists of up to three sub-components: a user profile records the user's interests, a similarity function of user profiles weights each profile for its degree of similarity, and a selection function denotes a set of the most similar profiles. A detailed discussion of algorithms for passive collaborative filtering is elaborated on Section 2.3.2.

Finally, pure collaborative filtering approaches try to solve shortcomings given for pure content-based systems (Balabanovic and Shoham, 1997). However, this approach leads to certain problems on its own. At first, a new item cannot be recommended until more information is obtained by user ratings or similarity specification. Furthermore, without a sufficient amount of ratings it is not possible to determine the neighborhood of a user. This problem is denoted as *cold-start problem* (Maltz and Ehrlich, 1995) or *bootstrapping problem* (Resnick et al., 1994). Second, if a user has unusual tastes compared to the rest of the population, he gets poor recommendations as long as no other user with similar interests is detected (Maltz and Ehrlich, 1995). The last two problems both indicate that collaborative filtering approaches depend on the size and the composition of the user population. To tackle these problems hybrid approaches for content-based, collaborative filtering have been developed (Balabanovic and Shoham, 1997).

2.2.3 Peer-to-Peer Information Retrieval

Recently, peer-to-peer (P2P) systems have emerged as popular way to share huge volumes of data. The underlying paradigm holds more than simple file sharing via search engines or peer-to-peer networks. However, information retrieval methods for peer-to-peer systems are still at their infancy. Information retrieval in peer-to-peer networks can be characterized by two goals: efficient retrieval of documents and effective finding of a set of best matching documents. Intelligent routing strategies are necessary to avoid a high network load. Many of the most efficient routing strategies rely on relatively simple retrieval methods and homogeneous network environments. The existing peer-to-peer schemes can be broadly categorized into: (1) *unstructured* P2P networks (Loser et al., 2003; Lu and Callan, 2003), which have the salient feature that data objects do not have global unique ids, and queries are formulated with set of keywords, and (2) *structured* P2P networks (Tang et al., 2003; Tang and Dwarkadas, 2004), which include systems that can be characterized by unique identification keys. The second approaches mostly focus on the use of P2P overlay networks for distributed indexing of document collections. Commonly, a hash of the content is used to build Distributed Hash Tables (DHT). They can be distributed over several nodes within a network. A concrete implementation of a DHT is realized in the Content-Addressable Network (CAN) model by Ratnasamy et al. (2001). In this model, all peers are arranged in a (logical) d-dimensional Cartesian coordinate space. The entire coordinate space is dynamically partitioned into so-called zones among all the nodes in the system. Each node is dedicated as the owner of exactly one zone. The partition into zones is utilized to conduct requests (insert, lookup, or delete) to key/value pairs. Each key is mapped onto one point in the coordinate space through a common hash function. This point does also correspond to a distinct zone that is maintained by a peer. Following the CAN model, the value of this key can be inserted or retrieved in the hash table of this peer. If this zone is not maintained by the requested node, the request is routed through a range of intermediate nodes towards the node that contains the key in his local hash table. To do so, each node additionally maintains a routing table that contains a number of adjacent nodes in the table. The topology of a CAN-based system is not fixed, new nodes can be inserted, existing nodes can be deleted and so on. For information retrieval techniques in peer-to-peer networks, distributed hash tables can not be easily used due to the limitations in scalability.

The work on distributed information retrieval (Callan, 2000) and metasearch is related research to peer-to-peer information retrieval. It is mainly concerned with the merging of results and database content discovery. The World Wide Web as a highly distributed information source intensified the research in this area. Balke (2005) revisited three approaches that are also common in peer-to-peer information retrieval:

- *Abstracts of Information Sources:* The set of terms in the collections's inverted index are often used for an abstract of the individual collection. Bloom filters (Bloom, 1970) are a popular technique for an efficient representation of these abstracts. PlanetP (Cuenca-Acuna and Nguyen, 2002) uses Bloom filters for retrieval and a gossiping algorithm disseminates a peer's index in a predefined community. More details of PlanetP are discussed in Section 2.4.2.

- *Collection Selection:* A major problem in distributed environments is the identification of resources containing documents relevant to a query. In general, benefit estimators are used to estimate the expected result quality for each individual collection. CORI (Callan et al., 1995) is the most popular benefit estimator for peer-to-peer information retrieval, because only a limited amount of statistical data needs to be exchanged.

- *Metacrawlers:* A related research field to collection selection are so-called metacrawlers. For example, GlOSS (Glossary of Server Servers) (Gravano et al., 1999) addresses the problem of selecting the most promising document collection from the WWW with respect to a query. It collects only meta-data about the individual collections like the number of documents in each collection and how many documents for each keyword. Metacrawlers can not be easy integrated in a peer-to-peer infrastructure because their index must be updated whenever the collection changes.

Related work on distributed information retrieval and metasearch addresses only the problem of integrating a small and typically rather static set of underlying retrieval engines and information sources (Balke, 2005). Federated search in such systems is less challenging than a collaborative search process in highly dynamical peer-to-peer systems. The major problems of peer-to-peer information retrieval is the peer's autonomy and the relatively high network churn. Tryfonopoulos et al. (2004) use the ideas of self-organized overlay-networks for an architecture to support both query and publish/subscribe functionalities. In their architecture, they differentiate between two kinds of nodes: super-peers and clients. All super-peers are equal and have the same responsibilities. Each of these peers serves a subset of clients. In addition, a generic architecture for a P2P-IR system is proposed by Arberer et al. (2004). The IR process is decomposed into four different layers. (1) Transport Layer Communication, (2) Structured Overlay Networks, (3) Document and Content Management, and (4) Retrieval Models. All layers have the advantage of using the same infrastructure provided at the lower layers. A key-based routing of (structured) overlay networks is identified as the key contribution of P2P systems to support P2P-IR efficiently. Furthermore, the modular design enables resource sharing of knowledge, and saves resources in global information retrieval.

The overhead of maintaining indexes in the presence of network churn is an important aspect for peer-to-peer information retrieval. The simplest method for querying peer-to-peer systems is flooding a query to all adjacent peers in a certain number of hobs. With this strategy, answers only from a limited radius around the querying peer are received. Routing indices (Crespo and Garcia-Molina, 2002) are a more sophisticated strategy to find very commonly queried items. The goal of routing indices is to choose the best neighbors of a peer to forward a query until the maximal number of desired results is reached. In addition, the idea of social metaphors is proposed for a locality-based routing. Tempich et al. (2004) incorporate in their research this strategy for routing queries to peers that may offer interesting documents. Each peer maintains a local index about content providers that have offered relevant documents for a query in the past. Typical popularity distributions show a high amount of replication of popular items in recent file sharing application (Chawathe et al., 2003). For a high retrieval quality, Bender et al. (2005) motivated the novelty concept of collections. An efficient query routing is proposed by a bookmark-driven approach for Web search (Bender et al., 2004). Every peer has a full-fledged search engine with a (thematically focused) crawler. All peers in this system are autonomous and share their local index by posting meta-information about their bookmarks.

2.3 Related Concepts

The selection of related concepts is inspired by emerging trends in the context of Web search. A new trend is observed for commercial and non-commercial Web search engines. On one hand, Web search services expand their service by personalized search interfaces, e.g. Google Personalized[4] or Yahoo! My Web 2.0[5]. On the other hand, collaboration among users can be explicitly assisted due to social relationships. For example, Yahoo! My Community's Web provides a service to administrating contacts that are considered during Web search.

[4] http://labs.google.com/personalized, last visit on 2006/03/01.

[5] http://myweb2.search.yahoo.com/, last visit on 2006/03/01.

Although these services allow users to participate in a more individual search process, they fail to address the fact that information production and consumption are implicit social activities. This dissertation addresses this problem with new techniques for an integration of personalization, collaboration, and socialization.

2.3.1 Personalization Strategies

This section elaborates on personalization strategies with particular emphasis on the Web. The process of Web personalization is defined as "a customization of a Web site to the needs of specific users, taking advantage of the knowledge acquired from the analysis of the user's navigational behavior (usage data) in correlation with other information collected in the Web context" (Eirinaki and Vazirgiannis, 2003). There are many kinds of data that can be collected on the Web. Srivastava et al. (2000) divide such data into four categories: *content*, *structure*, *usage*, and *user profile*. The content of a Web page is usually described as text and provides the main content resource for information collections. This content is organized by an intra-type structure including various HTML or XML tags or by an inter-type structure connecting pages with hyper-links. Usage data describe the patterns of usage of Web pages, such as IP addresses, page references, and the date and time of access. The collection of such data can be performed on a server level, client level, or proxy level (Srivastava et al., 2000). A user profile provides information about users of Web sites. In particular, a profile contains demographic information, such as name, age, country etc., for each user, as well as information about the user's interests and preferences. This information is acquired through registration forms or questionnaires, or is inferred by an analysis of usage data. The objective of a Web personalization system is to "provide users with the information they want or need, without expecting from them to ask for it explicitly" (Mulvenna et al., 2000).

Pitkow et al. (2002) describe two general approaches to personalizing search results for individual users: (1) query augmentation or (2) individual re-ranking of results. In this work, we focus on result re-ranking. For this approach, information about individuals is needed in the form of user profiles. Searchers are required to express their information need with a set of query terms being submitted to a search system. For this task, an information need existing implicit in the mind of the searcher is transformed into a search expression or query. This process is known as *query formulation*. Choosing the right description for an information need is not an easy task for a searcher. The resulting query is the compromised information need (Taylor, 1962). In the domain of information filtering a compromised information need is denoted as *profile*. It represents the user's long-term information needs. Two fundamental types of user profiles are identified by Kuflik and Shoval (2000):

Content-based Profile: This approach is concerned with the representation of a profile similar to a query. For query modification, relevance feedback (Salton and Buckley, 1990) is the main post-query method for automatically improving a system's representation of an information need (White et al., 2003).

Collaborative Profile: This type of profile consists of a set of 'nearest neighbor' users (Balabanovic and Shoham, 1997) whose past ratings have the strongest correlation. The correlation is based on the rating patterns of all users.

The distinction of profiles is derived from the classification of information filtering approaches discussed in Section 2.2.2. In this section the main emphasis is on content-based profiles. Collaborative issues are primarily elaborated on Section 2.3.2. In addition to content-based profiles, *filtering-rules* can be used to express information needs. Such rules include demographic and social characteristics of the user in order to compare different relevance judgments of a certain object by different users. Regardless of the type of a user profile, techniques for its *creation* and *update* are necessary. Two fundamental methods for the creation and update of profiles are classified as *user-created profile* or *system-created profile* (Kuflik and Shoval, 2000). Both methods deal with the creation of content-based profiles. In particular, system-created profiles are based on automatic indexing in order to identify the most frequent and meaningful terms constituting the profile. The following description is primarily focused on system-created profiles.

For the information filtering task, the similarity between a profile and each incoming document is calculated. All documents with similarities higher than a defined threshold are retrieved. The problem of setting *dissemination thresholds* is usually based on large sets of labelled sample documents (see (Callan, 1998), (Zhang and Callan, 2001)). All retrieved documents are presented to the user who provides *relevance feedback* to the system. This feedback information is used to update the profile. Owing to the similarity of information filtering to the traditional information retrieval task, many techniques originally developed for information retrieval can be applied to document filtering systems. In particular, profiles are usually indexed by methods such as the vector-space model. As discussed in Section 2.2.1, several conceptual information retrieval models can be distinguished, as well as basic approaches to relevance feedback in each of the models. For a survey of relevance feedback techniques developed on different retrieval models see (Ruthven and Lalmas, 2003).

Rocchio (1971) was the first who formalized a relevance feedback technique on the vector-space model. His approach is one of the most effective and widely applied algorithm. The objective of Roccio's algorithm is the expansion of an original query vector with terms that best differentiate the relevant documents from the non-relevant documents (Ruthven and Lalmas, 2003). In the context of the SMART system, several experiments with relevance feedback have been performed in order to examine different aspects, such as only using relevant documents, varying the number of documents, and using non-relevant documents (see (Ide, 1971), (Ide and Salton, 1971)). Several studies have shown the effectiveness of relevance feedback yielding major improvements with respect to the original query (Salton and Buckley, 1990). In recent years, several modifications to Roccio's algorithm have been proposed that improve the performance of this algorithm. Improvements can be achieved due to better *term weighting* (Singhal et al., 1996), *query-zoning* (Singhal et al., 1997), *dynamic feedback optimization* (DFO) (Buckley and Salton, 1995), *word contribution* (Hoashi et al., 1999, 2000), or *genetic algorithms* (Lopez-Pujalte et al., 2003). Furthermore, prior research suggested that the Exponentiated Gradient (EG) algorithm is as effective as Rocchio augmented with dynamic feedback optimization (Callan, 1998). A detailed comparison of Rocchio, EG (Kivinen and Warmuth, 2003), and Widrow-Hoff (Widrow and Hoff, 1960) algorithms is provided by Lewis et al. (1996). An approach dealing with the detection of shifts in user interests is presented by (Lam et al., 1996).

Independent of the relevance feedback algorithm, all approaches require users to assess a sample of the retrieved documents. Several studies have shown that *explicit feedback* from the user is clearly useful (see (Goldberg et al., 1992), (Yan and Garcia-Molina, 1995)). However, the criteria under which a user makes a relevance assessment can be subject to a number of factors (Ruthven and Lalmas, 2003):

- The *order* in which documents are shown to the user is important when assessing the relevance of a document (see (Florance and Marchionini, 1995), (Eisenberg and Barry, 1988)).
- Different representations of documents (e.g. title, abstract, or full-text) can affect relevance assessments (Janes, 1991).
- In practice, relevance assessments are often partial judgements, i.e., a document is only somewhat relevant to the topic, or the user is not sure of the document's relevance (Spink et al., 1998).

An alternative to explicit feedback is the usage of implicit feedback in order to infer the document relevance from users' behavior. Several types of implicit data can be captured as surveyed by the studies of (Hill et al., 1992), (Morita and Shinoda, 1994), (Nichols, 1998), (Konstan et al., 1997), (Oard and Kim, 1998). For example, the relevance of a document can be inferred from the time spent viewing a document. Unlike these studies focusing on newsgroup documents and relying on users interaction with the actual document, White et al. (2002) extend these concepts onto Web result lists. Their system seeks to capture a user's ephemeral interactions during a single search session, and predicts relevance based on this interaction. Summarizing, the study of White et al. (2002) shows that implicit feedback can be an effective substitute for explicit feedback, although it is not as accurate as explicit feedback. Hence, based on statistical methods an implicit feedback approach for interactive information retrieval (White et al., 2004) can use unobtrusive monitoring of interaction to help the system improve on the relevance of documents presented to the searcher. The feasibility of personalizing Web search by using an automatically constructed user profile as relevance feedback has been investigated by Teevan et al. (2005b).

2.3.2 Collaboration Strategies

Collaborative situations can be found with different facets in the Web context. From a network-based viewpoint, several social information spaces exist that facilitate communication and collaboration networks. Such spaces characterize the Web as a large social network (see Section 2.3.3). In 1993, Masinter and Ostrom (1993) identified two visions of how the usage of this global network will evolve in the future:

> First, individuals will use the network as an information and entertainment resource, providing access to material from libraries and other suppliers of information and entertainment. Second, in addition to communicating with these data sources, people will communicate with each other, using a variety of interactive text, audio, and video conferencing methods.

To date, applications resulting from both visions exist. This section places emphasis on specific collaboration strategies for information seeking processes as stated in the first vision. Moreover, Section 2.3.3 is devoted to the envisioned support of communication among users. Recently, most of the approaches assisting information seeking processes have been focusing on individual people with individual information needs, although shared information needs lead to a *collaborative information retrieval* (CIR), as well as a *collaborative information filtering* (CIF).

Collaborative Information Retrieval

Most of today's information retrieval applications are designed to serve individual users rather than people working in groups. In general, *collaborative information retrieval* focuses on information seeking as a cooperative process. This assumption leads to distinct manifestations of collaborative information retrieval:

- Definition by Baeza-Yates and Pino (1997): *A group of people trying to find at the same time some information needed by the group.*
- Definition by Fidel et al. (2000): *CIR focuses on situations where team members collaborate during various processes of information retrieval.*
- Definition by Hansen and Järvelin (2005): *CIR is an information access activity related to a specific problem solving activity that, implicitly or explicitly, involves human beings interacting with other human(s) directly and/or through texts (e.g. documents, notes, figures) as information sources in a work task related information seeking and retrieval process either in a specific workplace setting or in a more open community or environment.*

All definitions aim at a group of people benefiting from the experiences of others, although different facets of the composition of a group exist. According to Fidel et al. (2004), information retrieval is *collaborative* only when the parties involved are colleagues. From a global point of view, Hansen and Järvelin (2005) assume a specific workplace setting or a more open community. Indeed, information seeking has always been a social process (Wilson, 1981) and it is neither an individual activity nor a task in a rather isolated situation. Hence, a common paradigm of collaborative information retrieval must address *static groups*, as well as *dynamic groups*.

To date, the discussion of collaborative information retrieval primarily concentrates on established groups due to a growing emphasis on collaborative teamwork in modern workplaces (Fidel et al., 2000). In particular, research in computer support for cooperative work (CSCW) and in collaborative filtering has focused on this aspect. In this section primary collaboration concepts are discussed in relation to CSCW. This research area is devoted to the collaboration within organizations and work groups, as well as to systems supporting collaboration such as organizational memory, organizational information handling, and information sharing. Romano et al. (1999) discussed that Group Support Systems (GSS) lack integrated support for collaborative searching and visualization. Hence, they merge both paradigms of information retrieval and GSS into a *Collaborative Information Retrieval Environment* (CIRE). This new paradigm supports both individuals and team work. The contemporary prototype will be discussed in Section 3.2. Generally, CSCW-motivated systems are based on cooperative activities which can be classified according to Hansen and Järvelin (2005) as: (1) *asynchronous* or *synchronous* activities, (2) activities based on traditional *human communication* or *computer-mediated*, and (3) *loosely* or *tightly* coupled activities. Recommendations from other people based on observations of information seeking behavior are advantageous in loosely coupled activities. Tightly coupled activities aim at sharing queries and query reformulation. For a detailed discussion on collaboration in the context of CSCW see Hansen and Järvelin (2005).

The Web as a large dynamic network reveals numerous dynamic groups. A broader discussion of contemporary social information spaces and Web communities can be found in Section 2.3.3. However, this section comprises the detection of *shared information needs* supporting the collaborative information retrieval process. A dynamic group is assumed to have a shared information need if there is a relatively large overlap in user interests and queries. Collaborative information retrieval can benefit from this overlap by exploiting users' search processes for subsequent searches. A first approach based on this assumption is proposed by Hust et al. (2002). This approach unintrusively learns from all users' search processes. In a restricted CIR scenario all searchers cooperate due to the sharing of old queries and relevant answer documents to these queries. Based on a standard test collection, the CIR approach by Hust et al. (2002) performs better in combination with pseudo relevance feedback (Xu and Croft, 1996), although a general evaluation of such a scenario requires the query and interest distribution of a real world system. A similar approach based on usage data has been developed by Wen et al. (2002). Their assumption is that many people are interested in the same questions - the Frequently Asked Questions (FAQs). For this purpose, the goal of *query clustering* is to group queries/questions together in order to discover FAQs. Two queries are similar if they correspond to the same or similar document clicks. An evaluation setting with usage data from the Encarta Web site demonstrates that the new grouping of similar queries is more effective than using keywords by themselves. Further approaches to trend detection have been proposed by (Zhang et al., 2002), (Allan et al., 2003), and (Amitay et al., 2004).

Collaborative Information Filtering

Collaborative filtering was first introduced by Goldberg et al. (1992). In general, the term 'collaborative filtering' seems to denote joint ventures between people with shared information needs (Lueg, 2003). Indeed, such relationships are not explicit in fully automated systems. For a generalization of the approach, Resnick and Varian (1997) proposed the term 'recommender systems'. In this regard, Thor and Rahm (2004) presented a detailed top-level classification of recommenders. This section is devoted to a detailed discussion of *passive collaborative filtering algorithms* (see Figure 2.4 for classification details) integrated in recommender systems. The key idea of collaborative filtering refers to the notion of multiple users 'sharing' recommendations in a balanced cost-benefit relationship (Aggarwal et al., 1999). On one hand, collaborating users incur the cost (in time and effort) of rating various subsets of the items. On the other hand, each user receives a benefit from sharing knowledge in the collaborative group. Collaborative filtering tackles several drawbacks found in content-based filtering (Shardanand and Maes, 1995). However, the main challenge of collaborative filtering is to resolve a number of problems (Cöster and Svensson, 2002): *bootstrapping*, *concept drift*, and *scalability*. The first problem arises from sparse rating data leading to poor recommendations, if little information has been collected (cold-start problem). The second problem labels the phenomenon that information needs change over time, and user profiles need to be adapted to new long-term interests. The last problem is concerned with large numbers of users and titles that collaborative filtering algorithms have to scale up to.

In particular, the main attempt of collaborative filtering algorithms is the ability to make fast and accurate predictions. The computation of these predictions, as well as their presentation lead to a finer classification of prediction algorithms and applications. First of all, prediction algorithms are classified into either *memory-based* or *model-based* (Breese et al., 1998). Then, applications integrating a special type of prediction algorithms are classified into a presentation of predictions one-at-a-time or as a list of recommended items (Breese et al., 1998). For instance, memory-based and model-based applications present items to the user one-at-a-time along with a rating indicating the potential interest (for example GroupLens (Resnick et al., 1994)) or as an ordered list (for example Jester (Gupta et al., 1999)).

Memory-based algorithms operate over the entire user database to make predictions (Kim et al., 2004). This class of algorithms is primarily used in research and practice. Formally, memory-based collaborative filtering uses a nearest-neighbor approach (Herlocker et al., 2002) to find a subset of all users which are most similar to an active user in a three-stage process:

1. *Weighting Neighbors:* The task aims at the weighting of all users with respect to their similarity to an active user. The most common approach to similarity weighting is the *Pearson correlation coefficient* (Resnick et al., 1994). Taking into account that only a small sample of ratings on common items for pairs of users may exist, the similarity weight is adjusted with significance weighting and variance weighting.

2. *Selecting Neighborhoods:* This task aims at the selection of a subset of users as a set of predictors. In practice, not every user can be selected to be in the active user's neighborhood. Two techniques have been used so far to determine how many neighbors to select: First, *correlation-thresholding* is employed to set an absolute correlation threshold in order to select all neighbors with absolute correlation greater than a given threshold (Shardanand and Maes, 1995). Second, the strategy *best-n-neighbors* selects the best n correlates for a given n (Resnick et al., 1994).

3. *Making a Prediction:* The last step combines all neighbors' ratings into a prediction. A basic technique performed by all published work by the use of a neighborhood-based algorithm is to compute an average of the ratings using the correlations as weights. This averaging technique assumes that all users rate on approximately the same distribution (Herlocker et al., 1999). Pennock et al. (2000a) have shown that averaging techniques are a way to combine ratings with well-accepted axioms of social choice theory. Moreover, the basic approach can be modified by *rating normalization* and *weighting neighbor contribution* (Herlocker et al., 2002).

For a further discussion of the presented prediction process see (Herlocker et al., 2002). The advantage of memory-based predictions is a dynamic structure allowing immediate reactions to changes in the user database. An approach to tackle the sparsity problem of such a structure has been proposed by Huang et al. (2004). They apply an associative retrieval framework and *spreading activation* algorithms (Salton and Buckley, 1988) to explore transitive associations among users. However, the advantage of the dynamic structure can also lead to a potential drawback. For one, new ratings can be included in every nearest-neighbor search, and also, an online computation scales linearly with the number of users. To overcome these shortcomings, Goldberg et al. (2000) designed the *Eigentaste* algorithm, which provides accurate and efficient recommendations to users in constant online time. A more general drawback of memory-based algorithms is the possibility of modelling that one person is a reliable recommender for another person with respect to a subset of items (Hofmann, 2001). Hence, solutions to such a problem need to model the multi-dimensional nature of human preferences.

Up to now, model-based methods have not reached the same level of popularity as memory-based methods. They have gained much attention because they do not suffer from the performance and memory bottlenecks due to performing of complex computations in an offline modus. In practice, many e-commerce Web sites (e.g. amazon.com (Linden et al., 2002)) use model-based, sometimes called item-based, collaborative filtering. Various techniques have been applied to model-based collaborative filtering such as Bayesian networks, Latent Semantic Indexing, Singular Value Decomposition, and Mixture models. Two popular model-based algorithms are the *aspect model* (AM) (Hofmann, 2003) and the *Personality Diagnosis model* (Pennock et al., 2000b). The first model is a generalization of statistical techniques proposed as *probabilistic Latent Semantic Analysis* (pLSA) modelling individual preferences as a convex combination of preference factors (Jin et al., 2004). The personality diagnosis approach treats each user in the training database as an individual model. To predict the rating of an item by a test user, two steps are performed. First, the likelihood is computed for the test user to be in the ‘model’ of each training user. Second, the aggregate average of ratings is used for the item by the training users as an estimator. Recently, a new approach denoted as *clickstream-based* collaborative filtering has been receiving much attention (Kim et al., 2004) due to its scalability when performing collaborative filtering in Web personalization. Markov models, sequential association rules, and clustering are used as common prediction models for recommendations. Model-based algorithms perform as good as memory-based predictions in terms of predictive accuracy (Breese et al., 1998). The major drawback of this method is its inherent static structure (Cöster and Svensson, 2002).

2.3.3 Socialization Strategies

The support of information seeking in a *social information space* is an intrinsically social activity (Lueg, 2003). The research context is related to traditional information-seeking support (Marchionini, 1995), information retrieval (see Section 2.2.1), and social navigation (Munro et al., 1999). The design of advanced interfaces requires a careful consideration of the many ways in which users may interact with such spaces. Furthermore, the Web is a giant social network representing a wide range of human activities and interests. Information on the Web is authored and made available by (and for) millions of different individuals. All Web users operate independently in respect to their social backgrounds and knowledge. In general, a social network is modelled by a graph where the vertices represent individuals and an edge between vertices indicates that a direct relationship between the individuals exists (Kautz et al., 1997). From the viewpoint of statistical physics, the theory of random networks can be applied to these communication networks. Newman and Park (2003) argue that in general, social networks differ from most other types of networks. In practice, several simple models (e.g. classical random graph of Erdős and Rényi (1960), small-world networks (Watts and Strogatz, 1998), and Barabási-Albert model (Barabási and Albert, 1999)) are still far from being reality and only address particular phenomena in the real world.

Social Information Spaces

Each individual is involved in *social spaces*, both online and offline (Fisher, 2003). While in *offline spaces* people tend to be highly attuned to the social signals they send and receive from each other, the online world can be far more muted. *Online spaces* are active conversations (e.g. by email discussion lists or by instant messaging) in their own rights and in their connections to the real world. Hence, online conversation moves more slowly than offline. Recently, several asynchronous online social information spaces have come into existence where users can engage in conversation, make their presence known through contributions, and share ideas. Two of the contemporary information spaces are discussed below as representatives of differing design philosophies:

Usenet News This information space is a distributed decentralized bulletin board system which was first launched in the late 1970s. Users can perform two central tasks: read the collections of existing messages or post messages in response. All messages are clustered in newsgroups which the user can subscribed to. The strength of Usenet news is the discussion of a wide variety of topics where the boundaries among groups are well drawn because users focus on reading one group at a time. One of the weaknesses of Usenet news is the difficulty of finding a group. Especially for new users, it is hard to find groups of interests due to the overwhelming number of topics. Hence, Usenet news became the major domain for information filtering approaches. GroupLens was the first service which applied collaborative filtering to a social information space (Miller et al., 2003b). To date, Usenet news are exposed to a special kind of damage that is caused by the pollution of a group through "spam", for example with identical advertising messages.

Wiki Webs This information space is an expansion of traditional Web logs and guest books. Wiki Webs use a simplified mark-up language offering users powerful control over Web pages. As users are at liberty to contribute wherever they choose, Wiki webs are largely unstructured. Hence, a variety of social controls have been developed to ensure that users contribute in an orderly way. The strengths of Wiki webs are proved in a number of contexts which call for more local use than Usenet news, such as integrated links or a history of all changes and editors. Despite a large leeway for any individual to make substantial changes, it is impossible for a malevolent user to destroy old information. Nevertheless, Wiki webs are not protected against vandalism and systems have to deal with the restoration of damaged pages. An enhancement of Wiki webs for educational usage are collaborative webs (CoWebs). Further aspects of CoWebs are elaborated in (Dieberger and Guzdial, 2003).

The real challenge of all online social spaces is to decode subtle social signals. The collaboration in Wiki webs and conversations in Usenet news shape a group in a variety of important ways which are summarized by (Fisher, 2003) in five attributes: (1) access control, (2) the ability to contribute anonymously, (3) the ability to connect to offline discussions, (4) the ability to thread and organize conversations, and (5) the availability of archives. Most interfaces to social networks do not present any information about the social context of the interactions such as basic social cues about the size, activity, and demographics of groups. Hence, the design of social interfaces should address the fact that people are engaged in four major tasks when coming to a social information space: *discovery, selection*, *evaluation*, and *motivation* (Smith, 2003). In general, the success in managing collective goods depends on three features (Ostrom, 1990). First, reputation services offer a mutual awareness by reward and punishment for an effective self-regulation. Second, reputation and behavior tracking systems provide the state of social relationships between stable identities in real time. Third, social accounting systems track interactions, as well as transactions between groups in order to benefit both by receiving qualitative information and by building a reputation. The propagation of trust is a major problem for a number of e-commerce related sites. Guha et al. (2004) developed a formal framework for trust propagation schemes showing how distrust has significant effects on how trust is propagated.

Web Communities

Despite the decentralized, unorganized, and heterogeneous nature of the Web, an efficient identification of communities is possible due to a self-organization of the link structure (Bharat et al., 2001). The identification of Web communities aims at several applications such as automatic Web portals, focused search engines, content filters, and personalized search (Flake et al., 2004). In general, the Web is modelled as a graph where vertices are Web pages and hyperlinks are edges. Furthermore, simple generative network models explain a considerable amount of the Web's structure (Pennock et al., 2002). Several approaches assume this model for their applications (see (Dean and Henzinger, 1999), (Kleinberg, 1999), (Kleinberg et al., 1999), (Bharat and Henzinger, 2001), (Ng et al., 2001)). The assumption that the Web is a graph leads to a definition of a *Web community* as "a collection of Web pages such that each member page has more hyperlinks (either direction) within the community than outside of the community" (Flake et al., 2002). The task of community identification is a simplification of graph partitioning and clustering, although the basic task is differentiated from these fundamental problems by being within the Web domain. Hence, the goal of community identification can be summarized as a grouping of items that are similar to some seed set of elements. Several algorithms for the identification of a Web community can be distinguished based on the *degree of locality* used for assessing whether or not a page should be considered a community member (Flake et al., 2004). In particular, two main properties, such as *local* and *global*, characterize the community methods. A third type of method is characterized by the combination of both properties.

First, local methods reveal only the properties of the local neighborhood around two vertices to decide if the two are in the same community. Bibliographic metrics (Ikpaahindi, 1985) such as *bibliographic coupling* (Kessler, 1963) and *co-citation coupling* (Small, 1973) were formulated to capture the similarity between scientific articles. Both metrics are complementary because either the amount of overlap between the bibliographies or the referrers for two different documents are compared. Despite the simple representation of the metrics by linear algebra, two practical shortcomings concerning storage and disproportionate link overlaps exist (Flake et al., 2004). Hence, *bipartite cores* enhance the framework of bibliographic metrics so that the pages within a collection can be related to each other in an aggregate sense. Dense bipartite subgraphs are an important indicator of community formation (Chakrabarti et al., 2002). A formal model for a Web community based on a bipartite graph is formulated by Greco et al. (2004). In particular, Kumar et al. (1999) mine tens of thousands of bipartite cores and empirically observe that a large fraction are in fact topically focused and so specific that they are often not part of any existing portal hierarchy. Hence, owing to the self-organization of the Web, community cores are "natural" because they are not an artifact of a single individual entity.

Second, global methods demand that every edge in a Web graph be considered in order to decide if two vertices are members of the same community. Two fundamental algorithms are concerned with this task, which is denoted as *link analysis* (Henzinger, 2000) in the area of Web retrieval: (1) Kleinberg's *Hyperlink-Induced Topic Search* (HITS) (Kleinberg, 1999) algorithm regards a subset of the Web graph, and computes a *hub* and *authority* score for each page; (2) the *PageRank* (Brin and Page, 1998) algorithm is motivated by a random walker model of the Web. Both methods are referred to as spectral methods because they can commonly be described in terms of the spectral properties of an adjacency matrix, as well as the long-term behavior of a random walker (Flake et al., 2004). Neither HITS nor PageRank are primary methods for community identification. Hence, several variants exist adapting both algorithms to this problem. The HITS algorithm can be adapted for community identification by deploying less significant *eigenvectors* in a similar manner as classical spectral graph partitioning or principal component analysis (see (Kleinberg, 1999), (Gibson et al., 1998)). A generalization of the PageRank algorithm is denoted as *topic-sensitive PageRank* (Haveliwala, 2002). The random walker model is adapted by a focused move to a particular topic such as a periodic restart with a smaller set of favorite bookmarked pages.

Finally, community identification methods can also have local and global properties. Flake et al. (2000) formalized a community algorithm which can operate on the entire Web graph or a sub-graph. This algorithm is based on a *maximum flow framework*. For a detailed discussion of this algorithm in comparison to HITS and PageRank see Flake et al. (2004).

2.4 Related Work

The information retrieval community has a long research history, and many systems have been developed supporting information retrieval or information filtering processes. However, up to date none of the existing systems perform these tasks in an integrated manner. A high diversity of techniques have been combined in particular systems. Furthermore, heterogenous application domains have led to an unmanageable number of systems. The success of the Web significantly contributed to the rapid developments of the past years. For these purposes, the selection of related work is concentrated on specific aspects revealed in the problem statement: (1) personalized interaction, (2) community assistance, and (3) collaborative filtering.

2.4.1 Systems for Personalized Interaction

Web personalization strategies have been developed to adapt information or services provided by a Web site to the needs of an individual user (Eirinaki and Vazirgiannis, 2003). Further details on Web personalization have been discussed in Section 2.3.1. For this task, navigational behavior and individual interests are taken into account. The main goal of Web personalization is the determination of relevant information without an explicit request (Mulvenna et al., 2000). An exploratory study by Teevan et al. (2005a) shows that, despite the high level of interest in this topic, most Web search engines offer none, or limited, personalization features, at all. In the following discussion, a selection of systems aiming at personalized interaction is presented:

MyView The MyView project (Wolff and Cremers, 1999; Wolff, 2000) integrates structured and unstructured bibliographic information from heterogeneous digital libraries. Based on a personalized warehouse for bibliographic data in a unified scheme, techniques such as browsing and ad hoc queries are available. These functionalities are supported by a transformation of gathered bibliographic data records into a uniform scheme and by a storage of these records in a personal database. A user-centered information access is implemented by an efficient data retrieval and query post processing. MyView combines fully automatic parts (query generation and submission) and manual parts (adding information providers, defining the information need) to support the users in time-consuming and monotonous tasks, but leaves the responsibility to them in mission critical details.

Memex This system is dedicated to Vannevar Bush's dream of a memex (see Section 2.1) as an enhanced supplement to personal and community memory. In addition to this goal, Chakrabarti et al. (2000) designed a 'Memex' for the Web as a browsing assistant for individuals and groups with focused interests. In this context, personalization can be applied to both groups and individuals. Memex helps organize the browse history into coherent topics, and relate topics between different users. It also enables a search over the entire surfing history. Based on a client-server architecture, the stream of data from Web surfers is analyzed to mine community browsing experience. For the classification, learning algorithms combine features from text, hyperlink, and folder placement. The representation of a surfer's interest profile is a set of weights associated with each node of a theme hierarchy.

Outride This system is designed to be a generalized architecture for the personalization of search across a variety of information sources (Pitkow et al., 2002). Outride is integrated into the sidebar of the Internet Explorer. Based on a personalization engine, this interface performs the interaction with an Intra/Internet search engine. The main techniques for a personalized search are query augmentation and result processing. First, the component 'query augmentation' computes the similarity between the query and a user model based on information such as content interests, demographics, click stream, search history, and application usage. Second, a result processing technique individualizes the result set of the search engine by filtering based upon information in the user's model.

MyView was a worthwhile step in the direction of personalized information access. The application domain of literature provides first insights which can be assigned to the general Web search. Furthermore, it is an example for the paradigm shift from data-centered to a user-centered information approach as proposed by Watters and Shepherd (1994). A user-centered design is also essential for an integrated information seeking system with a balanced relationship between manual and automatic tasks. In addition to a user-centered design, the surfing behavior of the users is customized by Memex in order to assist a community memory. This central entity learns a classification scheme for Web sites, and each client can use this scheme to adapt the topic structure. Despite the collaborative exchange of surfing histories, the search facility of Memex is limited to the automatically collected click-through data. This shortcoming is also observed for the Outride service. The search service is a stand-alone application providing access to a Web search engine individualizing the result set based upon a user model. This user model is built of implicit usage data such as click-through data without qualitative document judgements by the user or the user group.

The discussion of related work for personalized interaction reveals the strengths and the weaknesses of the presented services. The combination of the advantageous concepts promises new insights into a system for integrated information seeking with: (1) *personalized access* to distributed information collections, (2) *collaborative exchange* of search histories, and (3) *classification* of Web sites into topics.

2.4.2 Systems with Community Assistance

The state of the art of community support is primarily influenced by the CSCW community. The discussion of advanced information sharing techniques has revealed community aspects in the context of collaborative information retrieval and collaborative information filtering (see Section 2.3.2). Also, Web communities have been interpreted as socialization strategies in order to exploit a networking among information providers (see Section 2.3.3). The distinction between collaboration and socialization strategies is based on two observations. On one hand, communities such as a group of users working as a team are used for collaborative information seeking in order to detect common goals and shared information needs. On the other, the primary goal of socialization strategies is to detect communities for specific topics in the context of large information spaces such as the Web. A linkage between a dynamic detection and their usage for information seeking is not considered in the related work. To date, no related information filtering system exists that takes into account memberships to distinct communities. Hence, three systems are selected that integrate communities for the retrieval task.

CIRE The system CIRE (Collaborative Information Retrieval Environment) (Romano et al., 1999) is dedicated to the support of collaborative information seeking and retrieving. It constitutes the implementation of an integrated knowledge creation environment in which information retrieval and GSS (Group Support Systems) are combined to provide integrated group support for all tasks required for teams to work together. Based on an information retrieval memory, all queries and the retrieved Web sites of a group are accessible to group members. Furthermore, each user can assess and comment on a Web site.

PlanetP The project PlanetP (Cuenca-Acuna and Nguyen, 2002) aims at the problem of content search in peer-to-peer (P2P) communities. For this task, PlanetP provides a framework for ad hoc sets of users setting up P2P information sharing communities without any central entity. The main goal of this project is to adapt a vector-space model instantiated with the $tf * idf$ ranking rule to the P2P environment. In a first step, each community member creates an inverted index of the documents that he shares. These local indexes are summarized in a compact form and are diffused throughout the community. Each peer collects the index summaries of all members and can query the collective information store of the community. The peers can be ranked according to their likelihood of having relevant documents.

YouSearch The application YouSearch (Bawa et al., 2003) is a distributed (peer-to-peer) search application for personal Web servers operating within a shared context (e.g. corporate intranet). It supports the aggregation of peers into overlapping (user defined) groups and the search over specific groups. The hybrid peer-to-peer architecture is augmented with a light-weight centralized component. The main goal of YouSearch is the exchange of data and information among users in a network. Moreover, rather than a simple file sharing, also a content-based search is implemented.

CIRE is a first approach at combining group support and information retrieval. It assists the exchange of information about search processes between all members, but predefined groups do not assist a topic specific classification across all users. In the context of knowledge management, it is essential to be aware of other groups' working contexts and to detect authorities. Hence, ad hoc communities promise a flexible association of users and topics. They can be assisted by the peer-to-peer (P2P) model. In this model, any two users wishing to interact can form a P2P community. PlanetP, as well as YouSearch have the advantage of a decentralized application without a single-point-of-failure problem. The difference between both systems is that PlanetP does not support the search among distinct communities. Instead, YouSearch assists the manual aggregation of users to overlapping groups. Its strength is a community-specific search for users working in the same context. However, the adaptation of such a system to large information spaces is difficult due to missing shared contexts. In particular, a manual aggregation of users is not manageable on the Web.

In summary, the strength of the related work is observed in corporate environments by a manual grouping of users. The P2P model promises the formation of ad hoc communities. First prototypes are limited to a manual accumulation, or provide no access to other communities. For these purposes, it is advantageous for an integrated information seeking system to assist: (1) the *discovery of ad hoc communities* and (2) *self-organization of memberships*.

2.4.3 Systems for Collaborative Filtering

The term *mutual awareness* summarizes the potentiality of a system being aware of other users. In general, this interpretation conforms to collaborative filtering strategies where all users are involved in the information seeking process. In this section, information filtering systems and their facility to support mutual awareness are primarily considered. In the context of the Web, such systems are usually referred to as recommender systems assisting an information push. According to Bates (2002), hundreds of millions of dollars have been invested during the Internet boom, but the push technology has largely failed. For example, in the mid-nineties the push service InfoGate (formerly PointCast) rapidly enjoyed great popularity during its peak with over 1.5 million members. This push service provided information through special channels for stocks, sports, weather, or business news. Designed as a client-server system, InfoGate paralyzed many networks. Several shortcomings caused the cancellation of the service by many users, for example the need for a permanent connection to the server, and the restriction to specific topics. Today, the InfoGate service no longer exists. In March 2004, Google came out with a new push service called Web Alerts[6]. At the same time, a personalized Web search[7] was launched. With Google as a major Web search engine, new approaches for pull and push services are maintained. However, both services are still independent, and results do not influence each other. This observation can be found true for most of information filtering systems. For further details on such systems see (Pretschner and Gauch, 1999). Only a selection of representative systems are discussed in this section:

Tapestry *Tapestry* is an experimental subsystem of the Xerox Mail Service at the Palo Alto Research Center (Goldberg et al., 1992). This system is based on a client-server architecture and can also be integrated into other systems, for example into NetNews systems. E-mails are classified through collaborative filtering methods as relevant or irrelevant according to special user interests. For these purposes, users are asked to give feedback on read e-mails.

GroupLens The system *GroupLens* (Resnick et al., 1994) extracts relevant subsets of NetNews articles for a user. In analogy to the Tapestry system, a collaborative filtering technique is used to generate recommendations. This process exploits positive user feedback of the past for future interests. For this task, GroupLens expects an explicit numerical ranking of an article in the range of one (not recommended) to five (excellent). The system is designed as a distributed system in order to collect rankings of several users for recommendations to other users.

NewsSIEVE The filtering system NewsSIEVE (Haneke, 1997, 2000) describes a learning algorithm for the classification of textual information. For this task, an adaptive filter is designed which is suitable for operation in conjunction with central servers. One essential requirement for this filter is the generation of ‘small’ interest profiles which are similar to the queries used for information retrieval.

All presented information filtering systems aim at a user-centered information flow in order to exclude irrelevant information. Rather than push services on the Web, traditional information filtering systems try to cope with the information load in specific domains (e.g. Usenet news or emails). Mutual awareness is achieved by collaborative filtering, although the degree of awareness differs across the systems. This degree depends on the privacy concept of each system. Only GroupLens has such a concept and it assists the usage of pseudonyms

[6] `http://www.google.com/webalerts`, last visit on 2006/03/01.

[7] `http://labs.google.com/personalized`, last visit on 2006/03/01.

for assessments and recommendations. Independent of the application domain it can be observed that all filtering system are limited in their support of mutual awareness. The usage of pseudonyms does not consider that the ‘word of mouth’ principle relies on implicit trust among people. Collaborative filtering is not based on a reputation service dealing with this implicit trust. Such a service is related to the privacy concept, as well as to the personalization concept. For a closed user community, reputation is based on naturally grown friendships among users and the system needs incorporates the closeness between two users. In more open communities, a privacy concept combined with a reputation service is necessary to facilitate virtual friendships.

In summary, collaborative filtering is restricted in the presented systems to individual assessments and recommendations without a reputation service logging the quality of recommendations. Hence, for integrated information seeking systems a collaborative filtering strategy must assist: (1) *community recommendations* and (2) *trust* in a particular user.

2.5 Summary

This section described many related techniques which inspire the concept of Congenial Web Search. Basic information seeking techniques such as information retrieval and information filtering gain large interest since the success of the Web. New challenges for information seeking techniques exists, and emerging trends are observed such as personalization, collaboration, and socialization. The primary goal of personalization is an individual customization due to usage profiles. Usage profiles aim at an inference of a more detailed view on the information needs based on past usage. This strategy does not consider explicitly that the information seeker is part of a community of like-minded individuals. Collaborative information seeking processes determine the information need of a user from different views. A collaboration technique compares and combines the profiles of different users as is popular in information filtering systems. For collaborative information retrieval, the survey of the existing techniques showed that research mainly concentrates on explicit collaboration. The retrieval process assists a common information need of a predefined user group, or a user can access the search history of other project members. For implicit collaboration, neither the group of users nor their interests is known in advance. New techniques are necessary for a discovery of collaborative search contexts in a generic setting. The transition of personalization to collaboration is determined by an implication of individual information needs to a group of individuals with shared interests. The characteristics of such a group with regard to information seeking are not surveyed in literature. In the context of socialization strategies, we explored two strategies to organize users and Web content. First of all, online spaces are active conversations in their own rights of like-minded persons. Social relationships or authority information are implicitly or explicitly coded in so called social software projects. Second, Web communities aim at several applications that utilize a clustering of the Web content. This approach does not incorporate the fact that Web pages are viewed during Web search by different users with individual information needs. In addition, the usage of Web pages is only maintained by search engines, and users cannot actively cooperate with other users during a search session. A new architecture for Web search is necessary to address this shortcoming.

3 Architecture for Congenial Web Search

This chapter introduces an architecture for Congenial Web Search. At first, we analyzed the requirements of a framework for a personalized, collaborative, and social Web search. For the conceptual design, we paid attention to functional and nonfunctional requirements. In particular, functional requirements impact the conceptual framework. This framework supports a common representation of queries, documents, and associations. An individual context is defined for a user's search interest. Our research was focused on the development of a prototype supporting search transparency. A peer-to-peer architecture fulfills this requirement, and we defined self-organizing system of equal, autonomous information providers which aims for the shared usage of distributed resources. In order to avoid cold-start problems, we integrated traditional Web search engines as external information providers.

3.1 Requirements Analysis

The process of requirements engineering aims at defining requirements of the system under construction including two main activities (Bruegge and Dutoit, 2004): *requirements elicitation* and *analysis*. In general, requirement elicitation and analysis focus only on the user's view of the system. All aspects which are not visible to the user such as system structure or design are not part of the requirements. The difference between both requirements engineering activities is the language and notation they use. Indeed, both activities express the same information. For this purpose, the emphasis of this section is the requirements elicitation which is written in natural language, whereas the analysis model is usually presented in a semi-formal notation. The first step to present the requirements of Congenial Web Search is to map a problem statement into a requirements specification including a set of *actors, scenarios, use cases*, and *nonfunctional requirements* (Bruegge and Dutoit, 2004).

3.1.1 Problem Statement

The first step of requirements elicitation is the specification of an initial problem statement. This dissertation aims at a system that matches rapidly changing information with highly variable interests. Recently, traditional Web search engines cannot deal with such a scenario, and several shortcomings have been elaborated in the first chapter. This problem statement summarizes these observations, and points out the main objectives of Congenial Web Search.

The prototype of such a system is called ISKODOR[1]. The problem statement is divided into four subsections:

Objectives

ISKODOR should be able to assist the general subtasks of information seeking as defined by Oard (1997): *collecting* information sources, *selecting* information sources, and *displaying* them. Moreover, the objectives of the ISKODOR system are to:

- provide an infrastructure for a *personalized interaction* with a Web search engine, a local search memory, and search histories of other users.
- provide an infrastructure to build *communities* and to manage community memberships.
- provide a framework to assist *mutual awareness* among users and information collections.

Functional Requirements

ISKODOR supports three types of users:

- The *operator* should be able to manage the users and the community infrastructure.
- The *active user* should be able to formulate a request, to view a list of results, to assess relevant documents, and to join a community.
- A *passive user* should be able to answer requests of active users.

Nonfunctional Requirements

- *Scalability.* The system must support the interaction of an arbitrary number of users.
- *Extensibility.* The operator must be able to add new information seeking algorithms, new reputation formulas, and new result presentation styles. Such additions may require a restart the system in order to add new modules to the system.
- *Low-bandwidth network.* Users should be able to access the system via ISDN modem or faster.

Target Environment

- ISKODOR should run on any operating system for personal computers.
- All users should be able to use a local application accessing the Web and the ISKODOR user infrastructure.

In summary, the problem statement comprises the objectives and it is a first source for functional and nonfunctional requirements of the ISKODOR system. The second step of requirements elicitation derives a scenario for Congenial Web Search.

[1] Abbreviation of the question "*Is sharing knowledge online a dream or reality?*"

3.1.2 Actors and Congenial Web Search Scenario

The second step of the requirement elicitation is the identification of different types of users who will be supported by the future system. In general, the identified actors are external entities that interact with the system. An actor can be a human or an external system. In addition to the problem statement, several actors are identified: `Operator`, `ActiveUser`, `PassiveUser`, `WebSearchEngine`, `LocalSearchMemory`, `Community`, and `NetworkGateway`.

After the main actors of the system are identified, scenarios define concrete examples of the future system in use. The core functionality of the system is searching information in the Web. Hence, a first example scenario, `firstSearch` (see Table 3.1) is developed to explore this functionality in more detail. This scenario describes the usage of ISKODOR by a new active user. The event flow includes all actors of the query process. In this scenario, for an active user the local search memory grows with each successful search. Also, he can collect additional group memberships. Once the general scope of the system is elaborated, the acquired knowledge can be formalized in form of high-level use cases.

Table 3.1: `firstSearch` scenario for ISKODOR

Scenario name	`firstSearch`
Participating actor instances	`karen:Operator`, `jonas:ActiveUser`, `leonie:PassiveUser`, `google:WebSearchEngine`, `peersy:LocalSearchMemory`, `java:Community`, `mygroups:NetworkGateway`
Flow of events	1. Karen adds Jonas to the group of ISKODOR users. 2. Jonas needs information about the Java application programming interface and he formulates a query ’*java api*’ for his information need. 3. Jonas’ query is sent to Google and Mygroups. 4. Google is used to retrieve Web documents. 5. Mygroups searches for existing communities relevant for the query. 6. Mygroups finds the community ‘java’ where all members used queries with a high similarity to Jonas’ query. 7. Leonie is a member of the ‘java’ community, and sends Jonas her documents associated with the community. 8. The results of Google and Leonie are merged and presented to Jonas. 9. Jonas selects documents and views each with a Web browser. 10. Jonas judges a document as relevant for his query and stores it in his PeerSy. 11. If a document proposed by the ‘java’ community is assessed as relevant by Jonas, he can join the community.

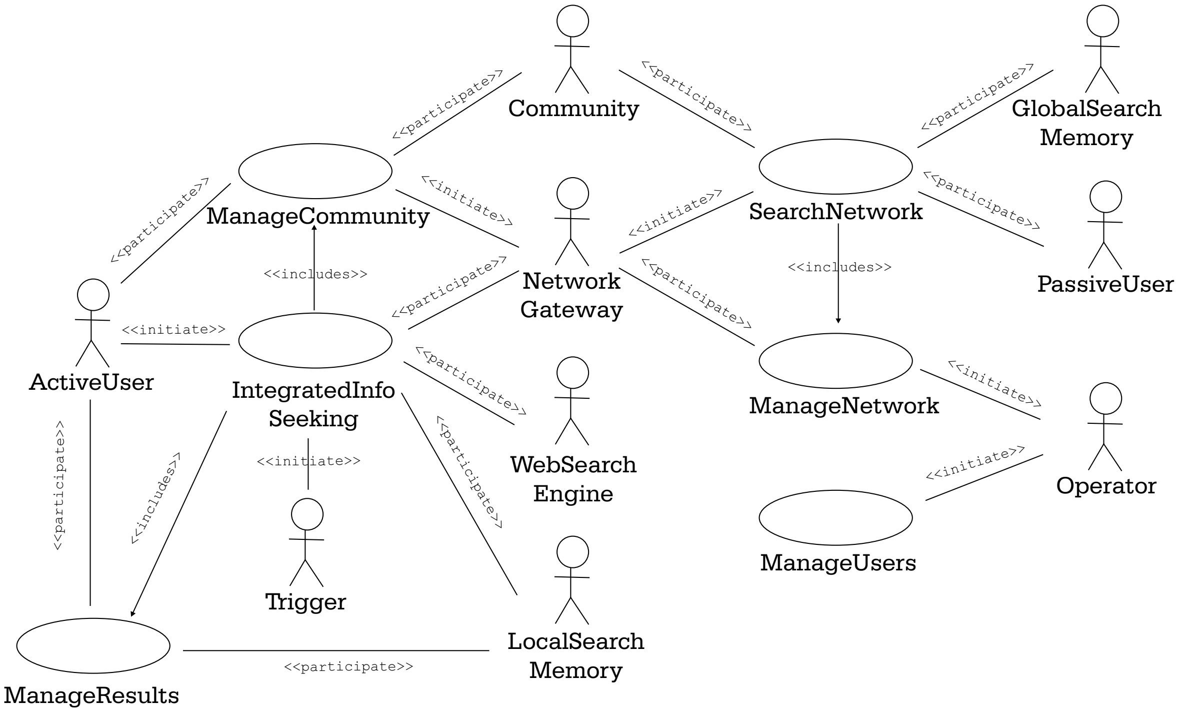


Figure 3.1: High-level Use Cases identified for ISKODOR

3.1.3 High-level Use Cases

The third step of the requirements elicitation is a complete representation of the future system by a set of use cases which are derived from the example scenario. In practice, this step is an iterative process in order to refine the use cases and their relationships. To generalize the situation of using ISKODOR for the first time, a broad range of functionalities initiated by all actors have to be covered by a use case. Owing to a simplification of such a large use case, the generalization attempts to split the use case into self-contained and independent use cases initiated by single actors. For this purpose, Figure 3.1 depicts all high-level use cases identified for ISKODOR.

High-level use cases primarily focus on the tasks accomplished by the actors. Moreover, the application domain is described with use cases capturing how different actors collaborate. To generalize the `firstSearch` scenario, the related functionalities are split into two use cases, `IntegratedInfoSeeking` and `SearchNetwork`. A brief description of both use cases reveals two new actors. First, a `Trigger` can start the integrated information seeking process automatically. Second, a `GlobalSearchMemory` maintains all information not grouped into communities.

IntegratedInfoSeeking The `ActiveUser` or the `Trigger` activates integrated information seeking which is either a retrieval or filtering process. The information need is processed in a parallel manner due to the participation of the `NetworkGateway`, the `WebSearchEngine`, and the `LocalSearchMemory`. The results are presented in a graphical user interface that is implemented by the `ManageResults` use case. It is included by the `IntegratedInfoSeeking` use case.

SearchNetwork A search query is propagated through the network in order to find matching communities. All passive users of these communities provide information to the request. The use case `ManageNetwork` is included in order to direct the requests. A `GlobalSearchMemory` is requested if no matching community can be found.

Besides the functionality concerning a running system, specific management tasks have to be performed by the `ActiveUser`, the `LocalSearchMemory`, and the `Operator`. For example, both use cases initiated by the `Operator` deal with the initialization of the network and the confirmation of new users. In addition, the `Operator` activities lead to a derivation of nonfunctional requirements which will be discussed in the next section. The activities of the `ActiveUser` and the `LocalSearchMemory` are used to further shorten the main use cases. Hence, the functionalities to manage the retrieved information, as well as the community are split off into two use cases:

ManageResults Documents which are found during integrated information seeking are presented to the `ActiveUser`. The presentation includes different presentation styles and organizes documents according their source. `ActiveUser` selects relevant documents, and the assessment is stored in `LocalSearchMemory`. Results which are found in previous search sessions can be managed by reassessing the relevance and by setting privacy variables.

ManageCommunity If similar interests are detected in the network, a community is announced, and advertisements to join the group are propagated in the network to specific active users. Community memberships are included during integrated information seeking.

This discussion of high-level use cases summarizes all interactions of the actors, although the diagram alone does not describe much functionality. Instead, the use case diagram can be considered as an index into further descriptions produced during this phase of requirements elicitation. The next step in the requirement analysis process is to write detailed use cases specifying the interactions between the actors and the system. A detailed description of each high-level use case includes the participating actors, entry and exit conditions, and a flow of events. Such a description captures all relationships among actors that the system must be aware of during the refinement of each high-level use case.

3.1.4 Nonfunctional Requirements

The process of requirements elicitation is completed with the identification of nonfunctional requirements that are visible for the user, but not directly related to the functionality. Indeed, these aspects have much impact on the development of the system. The problem statement of Section 3.1.1 already specified performance and implementation requirements. To ensure the identification of all essential nonfunctional requirements, the FURPS+ categories (Bruegge and Dutoit, 2004) are used to achieve completeness (see Table 3.2).

The last step of requirements elicitation comprises a number of nonfunctional requirements including typically conflicts among the requirements. In order to finalize the specification, a prioritization is necessary for these nonfunctional requirements in order to address them consistently during the implementation of the system. This dissertation mainly emphasizes on new insights to functional requirements. A further discussion of the nonfunctional requirements is related to the system design presented in the next section.

3.2 System Design

ISKODOR implements a virtual search network with cooperative information exchange in a pull-push cycle. Through suitable structuring and linkage of information, it facilitates information retrieval, as well as information filtering. Techniques for personalization, collaboration, and socialization are integrated in this platform. The general architecture of this platform is derived from the high-level use cases depicted in Figure 3.1. The major use case at this level defining the general system architecture is the `IntegratedInfoSeeking` use case. In this use case the active user interacts with three different actors: `LocalSearchMemory`, `WebSearchEngine`, and `NetworkGateway`. The architecture of the Web search engine is determined by an external provider. Hence, a common platform for local repositories such as the `LocalSearchMemory` interacting via the `NetworkGateway` must be defined. The actor `NetworkGateway` is a broker between a requester and all other users involved in the system.

3.2.1 Network Topology

In the World Wide Web, client-server architectures model a broker function by asymmetric relationships between information consumers and providers. Well-known shortcomings of such architectures are the ‘single source bottleneck’ and the ‘single point failure’ problem. All major Web search engines have a client-server architecture and they are very popular. However, these search engines are very efficient due to this architecture, and they provide a large coverage of the Web. For these purposes, we do not claim a new system architecture to exceed the existing systems with respect to their index size. Instead, a flexible architecture for providing information is implemented that relies on the computational power and the bandwidth of all participants. Such a decentralized architecture is based on the peer-to-peer (P2P) paradigm. If common search interests among users are detected, the equality between users enables their interaction. As depicted in Figure 3.2, pure centralized systems can evolve to pure decentralized systems.

A P2P architecture (Steinmetz and Wehrle, 2005) offers a transparent service. Each peer is anonymous with the optimum assistance of individualization. On this account, a traditional Web search engine is integrated as a Web service to guarantee efficient processing of requests. Each network peer is an information provider, as well as an information consumer. The consumption of information is interpreted as the active part of the peer, and when information is provided, it becomes the passive part. Furthermore, each peer works with others for a common purpose. In summary, a peer-to-peer architecture has four essential advantages for our system design:

Table 3.2: Nonfunctional Requirements of ISKODOR

Category	Nonfunctional requirements
Usability	(1) ActiveUsers must be able to search the network without prior knowledge of the network members and with an empty LocalSearchMemory. (2) Standard graphical user interfaces in the style of Web search engines must be supported.
Reliability	(1) Information about a Community must be cached. (2) A failure of the Internet connection allows only the access of the LocalSearchMemory, and an actual search session is reset. (3) Each ActiveUser is responsible for the backup of the LocalSearchMemory. (4) The permanent availability of the NetworkGateway is not guaranteed. (5) The availability of the WebSearchEngine depends on the external provider. (6) Only the Operator is responsible for updates due to security reasons. (7) The privacy of the LocalSearchMemory is guaranteed for all information which is not assessed as ’public’ by the ActiveUser.
Performance	(1) ActiveUsers and PassiveUsers should be able to access the system via ISDN or faster. (2) The system must support the parallel usage of the NetworkGateway for ActiveUsers and PassiveUsers without a limited number of participants. (3) The performance of the system depends on the latency of the NetworkGateway having the highest variance due to technical implementation issues.
Supportability	(1) The Operator must be able to propagate updates of the service to all users. An update adapts the information seeking algorithm of the ActiveUser, replaces reputation formulas on PassivePeers, and adds new result presentation styles for the ActiveUser. Such additions may require to restart the system in order to add new modules to the local system. (2) The Operator must be able to exclude users from the system infringing the policies of the system.
Implementation	(1) ISKODOR must run on any operating system for personal computers. (2) All users must be able to use a local application accessing the Web and the ISKODOR user infrastructure.
Interface	The system must interact with a browser supporting Javascript and Java applets.
Packaging	The ActiveUser installs the system and all necessary components.
Legal	(1) Each ActiveUser has to accept the usage policy of ISKODOR. (2) The Operator is not responsible for the topics of the communities and the content of LocalSearchMemories.

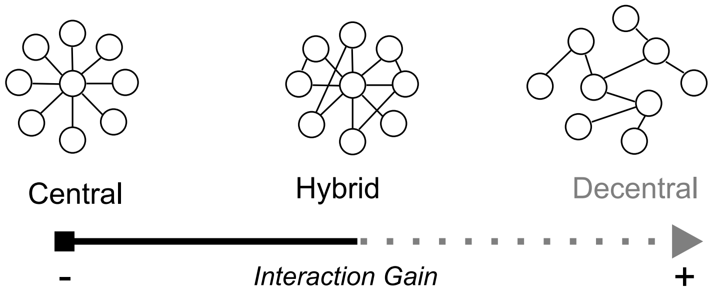


Figure 3.2: Evolving Decentralization by Interaction Gain

Autonomy Each user is an autonomous searcher in the hybrid peer-to-peer network (Eberspächer and Schollmeier, 2005). He decides which information sources are requested, which particular document is relevant, and if it is of personal or public interest. Furthermore, he commits his membership to a Virtual Knowledge Community explicitly. It enables nearly no form of tyranny of the majority, because each individual interest is treated equally.

Transparency The local storage of usage data such as explicit feedback information enables full system transparency. Furthermore, user objections regarding the centralized storage of personalized search information can be diminished by such an architecture. In addition, the filtering process based on a community becomes transparent due to explicit memberships.

Reputation A reputation model is essential for a peer-to-peer network. In general, information is replicated on several peers. For a reliable resource selection process, reputation can be considered by each interaction. In the context of file sharing, the success of an interaction can be rated by implicit and explicit feedback information (see EigenTrust (Kamvar et al., 2003)). In addition, if a user stores a retrieved or recommended document of another user in his Peer Search Memory, this action can be used for modelling reputation.

Self-Organization The main advantage of the hybrid peer-to-peer network is its feasibility of a self-organization of users (De Meer and Koppen, 2005). Each peer-to-peer network is an ad-hoc network of users with a common search interest. In general, all users are connected by ‘weak ties’ in such a network (Granovetter, 1973). Based on specific search requests, these ties evolve to a strong connected component of the network in form of a peer group.

The integration of a centralized Web search provider enables the ISKODOR architecture to avoid cold-start problems. The number of users and their search behavior define how fast ISKODOR evolves to a decentralized system. For a general software architecture, design decisions can be derived from existing systems. Peer-to-peer systems are becoming increasingly popular, although application development is currently not very efficient. Many applications share the same properties, such as discovery of peers, searching, and file or data transfer. Many developers solved the same problems by duplicating similar infrastructures. Hence, most applications are unable to communicate and share data with other applications. To avoid

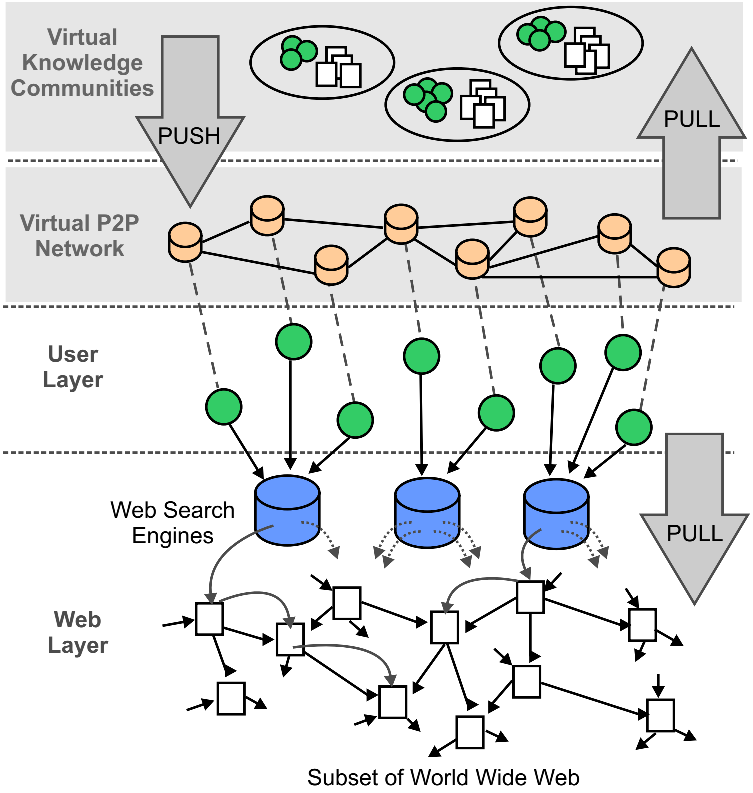


Figure 3.3: Topology of a Hybrid Peer-to-Peer Network

such problems, the prototype for Congenial Web Search uses a platform with basic functions necessary for a P2P network. The primary advantage of such a general platform is the interoperability. Specific ISKODOR components of Congenial Web Search are implemented as services enabling peers to communicate with each other in different search sessions.

For the assistance of both information detection processes, we want to improve existing Web search facilities in order to model dynamic information sources. All common Web search engines have a client-server architecture, where no interaction among users is feasible. This scenario is depicted with the ‘Client Layer’ and a ‘Server Layer’ in Figure 3.3. A user selects one or more search servers, and all retrieved documents provide an access point for a navigation through the World Wide Web. On this account, the user performs a continuous selection of an information source, which has a stable collection of documents at the moment of request. The interaction among users is limited to an information pull. For the assistance of pull and push services, we propose a hybrid peer-to-peer network. The main advantages of this network is the support of interaction among users. It is a hybrid network, because existing Web search engines are integrated for an efficient information pull. The effectiveness can be enriched by the interaction of users and their exchange of relevant documents in a ‘Virtual P2P Network’ (see Figure 3.3). For a tracking of new information sources, an additional layer of ‘Virtual Knowledge Communities’ is implemented, in order to restrict the push to selected users, which are organized in a community.

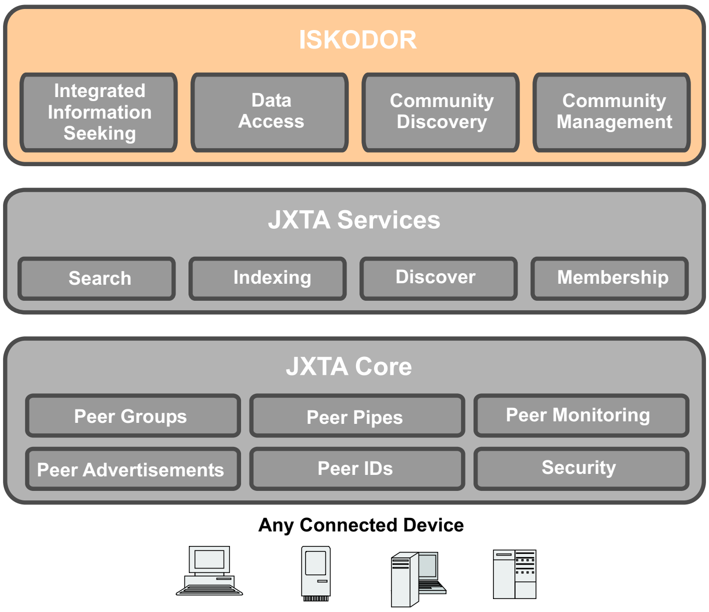


Figure 3.4: ISKODOR Software Architecture based on JXTA

3.2.2 Software Architecture

For efficient software development, the architecture of ISKODOR is based on Project JXTA[2]. This platform provides basic functions necessary for a P2P network. JXTA is not a specific application but rather a system architecture based on existing standards such as XML and TCP/IP. The project seeks to overcome potential shortcomings in many of the existing P2P systems: interoperability, platform independency, and ubiquity. Furthermore, the conceptual goals of JXTA define standardized protocols for discovery of peers, self-organization into peer groups, advertisements and discovery of network services, communication with other peers, and monitoring each other.

For a prototype implementation, we chose the JXTA framework for a standardized communication and organization of peers within peer groups. According to our findings, the usage of JXTA is promising due to the following aspects:

- *De-facto standard:* To date, JXTA constitutes the most sophisticated technology for creating P2P architectures. The JXTA standard is fully implemented in terms of an open reference implementation in the programming language Java.
- *Peer Grouping:* JXTA provides suitable concepts for grouping peers into self-governed groups, which can be used for the Virtual Knowledge Community approach.
- *High Scalability:* JXTA's efficient routing and retrieval algorithms support our demand for a widely used decentralized application.

JXTA has its own system model to describe transactions among participants. These concepts which are essential for the ISKODOR prototype are elaborated in detail:

[2] `http://www.jxta.org`, last visit on 2006/03/01. The term 'JXTA' is short for juxtapose, as in side by side.

Peer A peer is a network node implementing one ore more JXTA protocols. Each peer can be any connected device such as a PC or a server. It is identified by a unique Peer ID. Peers operate independently and asynchronously from all others. For the communication with other peers, each peer publishes one or more network interfaces using the JXTA protocols. A published interface is advertised as a peer endpoint to establish direct point-to-point connections. Intermediary peers are used to route such messages to peers that have no physical network connection.

Peer Group Each peer is typically configured to spontaneously discover each other on the network to form transient or persistent relationships denoted as peer groups. A peer group is a collection of peers that have agreed upon a common set of services. Each peer group is identified by a unique peer group ID. Each peer can be a member of several peer groups. By default, a first group called NetPeerGroup is instantiated where all peers are members. Each peer group provides a set of services called peer group services. ISKODOR defines additional peer group services which will be discussed in Section 3.2.3.

Pipe Peers use pipes to send messages to one another. They are based on an asynchronous and unidirectional message transfer mechanism which is based on a virtual communication channel. The communication is virtual because pipes connect peers that do not have a direct physical link. Pipes offer two modes of communication: point-to-point pipes and propagate pipes. The first type of pipes connects exactly two pipe endpoints together. These endpoints are dynamically bound to peer endpoints at runtime. The second type of pipes connects one output pipe with several input pipes. The propagation is done within the scope of a peer group.

Message The object which is sent between peers is called a message. A message is sent and received by the Pipe Service. Each message consists of an ordered sequence of named and typed contents in a XML or binary representation. JXTA protocols are specified as a set of messages exchanged between peers.

Advertisement All JXTA network resources (peers, peer groups, pipes, and services) are represented by an advertisement. A peer resource is published by an advertisement with a XML document. Peers discover resources by searching for corresponding advertisements, and cache any discovered advertisement locally. Each advertisement has a lifetime that specifies the availability of its associated resource. This lifetime facilitates the deletion of obsolete resources without any centralized control. If the advertisement is republished, the lifetime is extended.

Besides the essential primitives of P2P networking, JXTA supports different levels of resource access. All peers operate in a role-based trust model. Security aspects have been characterized as non-functional requirements in Section 3.1.4. JXTA already provides confidentiality, authentication, authorization, data integrity, and refutability.

In order to address a large number of users, the system is designed to be platform and browser independent. Based on the Project JXTA software architecture, the ISKODOR prototype has a three-layered system design as shown in Figure 3.4. The *platform layer* encapsulates minimal primitives for P2P networking with associated security primitives. Essential mechanisms

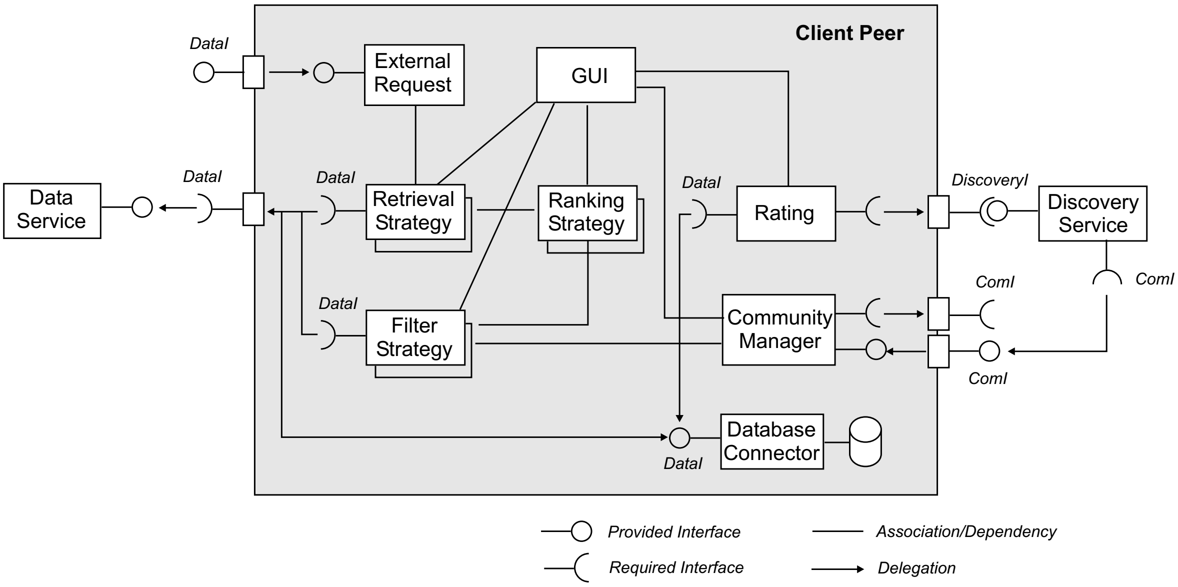


Figure 3.5: ISKODOR Application Design based on UML 2.0 Component Diagrams

such as discovery, transport, the creation of peers and peer groups are necessary for a P2P network to operate. Instead, the *service layer* includes optional network services. Examples of such services include searching and indexing, discovery, or membership services. On the top level of the architecture, the *application layer* includes implementations of integrated applications. ISKODOR is an integrated application on top of the first two layers. The design of the application is depicted in Figure 3.5 as a component diagram based on UML 2.0 (Booch et al., 2005). All sub-components are grouped to four main components:

- *Data Access*: This component manages all data accesses on the local peer, as well as in the network. It includes the `DataService`, the `DataConnector`, and the `ExternalRequest` sub-component. All sub-components provide the interface `DataI`. The `DataService` sub-component is an external data service. It mediates the access to a Web search engine and all members of the same peer group as the client peer. The `ExternalRequest` sub-component provides an interface for external requests from other peers. A specific interface is implemented to control the access from other peers. The `DataConnector` provides access to data stored in a local database. It represents the Peer Search Memory which is modelled in Chapter 4.

- *Integrated Information Seeking*: This component includes all sub-components for the retrieval, the filtering, and the ranking of documents. The sub-component `GUI` implements the graphical user interface and provides a search box, different result presentations modes, as well as administrative functions. The `RetrievalStrategy` sub-component processes a query according to the integrated information retrieval concept as detailed in Section 6.2. The `FilteringStrategy` sub-component implements the integrated information filtering concept as detailed in Section 6.3. All documents either retrieved or filtered are ranked by the `RankingStrategy` sub-component. It considers all different data sources and applies a personalized, a reputation-based, or a community-based ranking. Owing to a modular design, all sub-components can be exchanged with different retrieval, filtering, or ranking strategies.

- *Community Discovery*: This component stores explicit feedback and synchronizes the feedback with a service to discover collaborative search interests. The Rating sub-component implements two required interfaces. First, the DataI interface enables a local database connection to store explicit relevance feedback. Second, the DiscoveryI interface delegates the rating to an external discovery service. The DiscoveryService sub-component processes all users' relevance assessments and groups terms, links, and users. If a community is discovered, the CommunityManager of the peer that initiated the community is notified. For this task, each client peer provides the ComI interface.
- *Community Management*: This component maintains peer groups in a self-organized manner. For this task, other peers are notified about discovered communities. The interface ComI is used to offer community memberships to other peers and to collect their confirmations. This component is aware of all actual members of a community with a membership of the client peer.

In summary, all presented concepts of the JXTA framework are used by the ISKODOR application in a modular manner. With the JXTA framework, the ISKODOR architecture is based on a standardized architecture for peer-to-peer networks. The entire system is modular with a combination of existing and new services for Congenial Web Search. With the core JXTA services, the actor NetworkGateway is fully represented. On the application layer, additional concepts are necessary adapting the core JXTA services. This decomposition is based on services which are identified in the concept design and the high-level use cases. In general, the distinction between services and applications is not rigid. An application of one customer can be viewed as service for another customer.

3.2.3 Peer Interaction and Services

Owing to the three layered JXTA software architecture (see Figure 3.4), it is possible to overcome the shortcoming of proprietary system development by standardized network primitives. In addition to the basic peer group services, new services are developed for Congenial Web Search. These ISKODOR services provide personalized access to information sources. The specific type of access depends on the information need. For Congenial Web Search two access types are facilitated: *local access* and *global access*. These access types are correlated with the information need they support. Each peer supports a Peer Search Memory (PeerSy) which is the fundamental basis of the model. The Peer Search Memory is modelled in Chapter 4. For an information retrieval process, PeerSy assists the user with repeated queries in terms of a personalized ranking and display. This process is visualized with a collaboration diagram in Figure 3.6.

In this process, a user formulates a query, and it is submitted via the search interface that is implemented by the GUI sub-component. The query is delegated to the RetrievalStratey module. In parallel, the query is propagated first to the local database and second to the external data services. We chose this order, because we expect a faster response of a local data storage than external ones. Both sub-components send their retrieval results back to the RetrievalStratey module as soon as they are available. The DataService sub-component

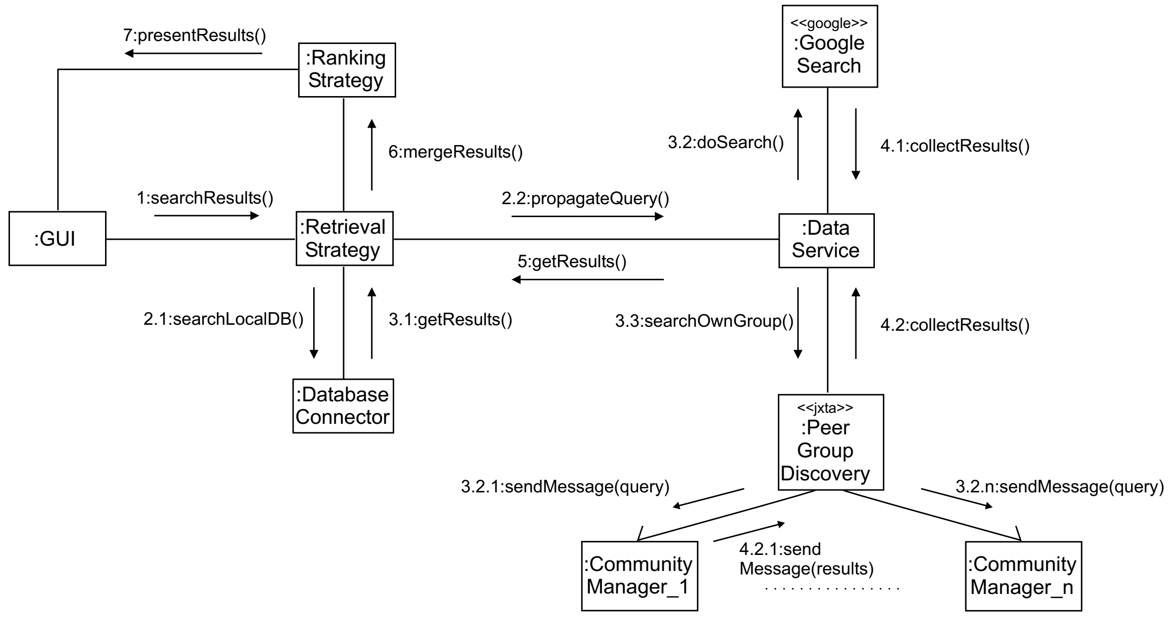


Figure 3.6: Collaboration Diagram for Information Retrieval

provides a common interface to request a Web service and a peer group service. For example, we integrate Google's Web service to access a large information collection. In order to retrieve communities relevant for a query, we use JXTA's peergroup discovery protocol. For this task, a peer group resource is published by an advertisement that is maintained by the `CommunityManager` of each peer. A community search is implemented by an asynchronous request because not all peers offer the peer group service. All results are merged and ranked by the `RankingStrategy`. Finally, all ranked results are presented in the user interface.

The filtering process works similar to the retrieval process as depicted in Figure 3.7. The main difference is the initialization of the collaboration. The `CommunityManager` initiates the community filtering periodically with a trigger. All recent community memberships are considered, and each community is filtered. Instead of a query, the `FilteringStrategy` module activates a process to generate a set of query terms which are used to request all community members. In order to get results from the peer group, a JXTA peer discovery service is used to send asynchronous message to all members of the group. The `ExternalRequest` module of each client peer processes the message and retrieves documents from the local database. The `DataService` collects all results from peers that have answered the request. A timeout is implemented to handle different response times within the network. The `FilteringStrategy` module collects all results, and initiates their relevance prediction by the `RankingStrategy`. All recommended documents are presented in the user interface.

The concept of peer interaction requires specific non-functional requirements. Privacy and security constraints get a high priority in order to guarantee a reliable exchange of information. Each user decides which transactions are public available for others. This is a first step in order to distinguish between personal and public information. All personal transactions are hidden during an interaction with other internal sources. Furthermore, this privacy concept is combined with a security concept preventing an unauthorized access of personal information. It guarantees that no process of another peer is performed on a local one. In-

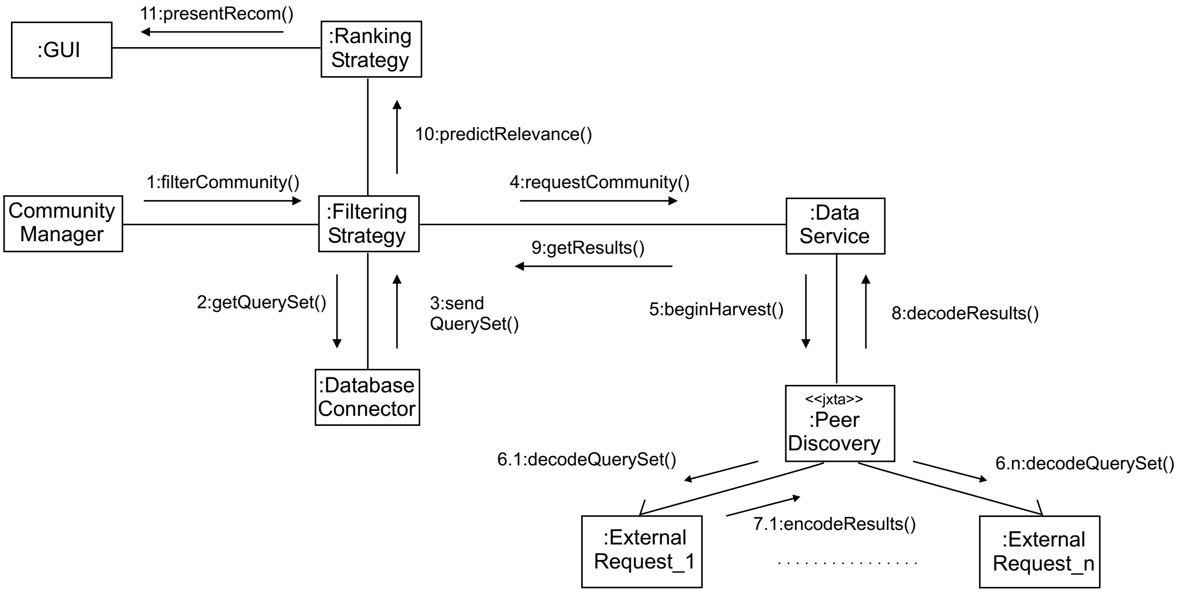


Figure 3.7: Collaboration Diagram for Information Filtering

teractions among internal sources are managed by services regarding the privacy and the security of each interaction. These services send either *point-to-point* or *propagate* messages. Both terms distinguish whether an interaction is performed between two specific users or all users. The ISKODOR software architecture uses three services which handle all interactions:

Request Service A request service establishes connections between users in order to exchange messages. A message consists of query terms and the last request time if it is repeated. After a request is performed, the service waits for results of the answer services.

Answer Service The answer service computes a ranked output for a particular request. The answer message consists of a set of document represented by URLs. The local selection process considers the timestamp of the last request. Only for repeated interests this timestamp is necessary in order to select new documents since the last request.

Community Service For the detection of Virtual Knowledge Communities, a peer group service is responsible for the identification of common search interests. This service is not a pure P2P service. At first, a central entity in the network discovers common search contexts in individual search sessions. If a community is detected, a membership service facilitates the peer group management in a decentralized manner.

3.3 Summary

This chapter presented the architecture for Congenial Web Search. At first, the requirements analysis identified a set of actors, scenarios, use cases, and non-functional aspects. The analysis showed that active and passive user participations are basic functions of the system. This functional requirement needs a user-centered system design which conforms to the main goal of Congenial Web Search. A hybrid peer-to-peer architecture is utilized to assist information consumers and information providers. Furthermore, peer-to-peer characteristics such as autonomy, transparency, reputation, and self-organization are combined with large-scale search services. This combination prevents cold-start problems. ISKODOR is a prototype implementation that maintains requests, answers, and community services. Owing to mutual access and exchange, information is propagated in the network. With JXTA a de-facto standard for peer-to-peer applications is selected for a software architecture. ISKODOR is an integrated application on top of the JXTA core, and its predefined services. The local data access is realized with a peer service which models a Peer Search Memory (Chapter 4). The result of the discovery process of all individual transactions is a community service (see Chapter 5). Request and answer services perform particular information requests during Congenial Web Search. Several instances of these services are running in a parallel manner. Retrieval and filtering processes rely on the integrated information seeking concept which is elaborated in Chapter 6. More implementation details of the prototype are discussed in Chapter 7.

4 Modeling a Peer Search Memory

This chapter describes the first pillar of Congenial Web Search. A personalization of individual search processes models distributed information collections which enhance Web search. At first, we analyzed the search behavior with a usage log in order to verify the conceptual requirements, and the design of a Peer Search Memory. We noticed a high individuality in all search sessions. Support for the user is achieved by a personalized access to a Web search engine regarding the user's search history.

4.1 Analysis of Search Behavior

In general, users' search behavior can be observed through their formulated queries and viewed documents. We analyzed the search behavior in order to verify the necessity of a Peer Search Memory. Its goal is a storage of explicit relevance feedback to form a distributed information collection. During search, this local history of relevant documents enables the system to distinguish between two subsets of the result list which are presented to the user: (1) a set of known documents, and (2) a set of new documents. It depends on the user's information need from which subset he prefers his results. We investigated the impact of a local storage by focussing the following aspects:

Diversity The analysis of the diversity in all search sessions considers queries and links. A high diversity in all search sessions of a user shows that a Peer Search Memory will represent a dynamic information collection. From the diversity of all individual collections, we can predict the future evolution of the system and its scalability.

Repetition Besides the dynamic of individual information collections, repeated queries and links are a sign for recurrent interests, or the interest could not be satisfied in previous search sessions. The amount of repetitions will give insights about the effectiveness of recent Web search, and the impact of an individual user to a user group by sharing information.

In order to observe both aspects in the real search behavior, we explored a Weblog corpus. It includes search sessions of a user group which have not been assisted by a search history. We analyzed all search sessions according to their diversity and repetitions. On one hand, we performed a user-centered verification. For each individual user, we analyzed the diversity of his information collection and the repetitions within it. On the other hand, we accomplished a

Table 4.1: Weblog Test Corpus collected over 536 Days

	number
Queries (distinct)	63,164
Links	160,043
- HTML	138,625
- PDF	19,128
- Doc	1,058
- others	1,178
Sessions	220,505
Peers	724

system-centered verification in order to analyze the global diversity of interests of the whole user group. The analysis of repetitions among all users shows that the limitations of the own information collection can be enhanced by others. From this global point of view, a collaboration strategy is motivated that considers similar search interests.

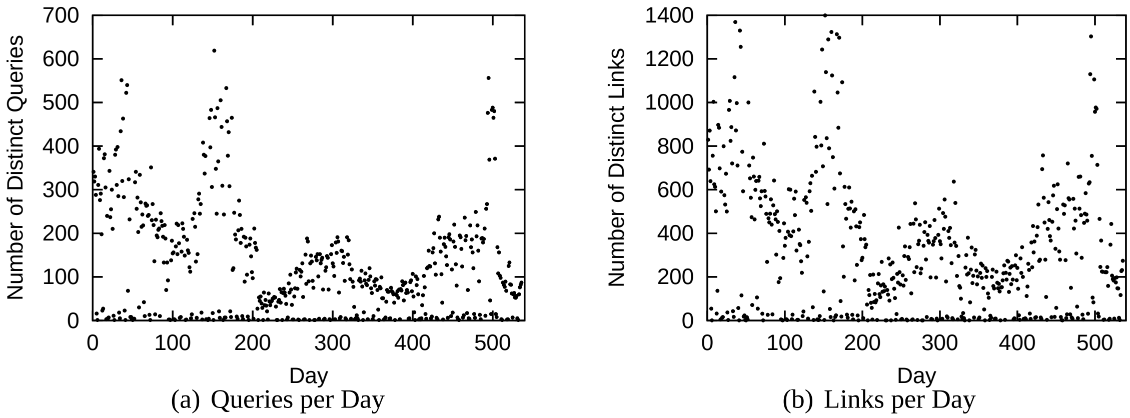
4.1.1 Weblog Corpus

The test corpus is based on user logs from a German campus Web proxy. We collected user logs of 536 days from October 7, 2003 to August 4, 2005. From these logs, 220,505 *search sessions* to a Web search engine were extracted. Each session specifies a 4-tuple (q, p, t, u) where q is a query of user u who views a Web page p at time t. The search time is represented in the UNIX timestamp format. All timestamps of a day were grouped to one time unit. For the analysis of the exploratory data, only logs to one Web search engine were considered. We used Google, because it was the most frequently used search engine of all users. In total, 724 users asked 63,164 distinct queries and viewed 160,043 distinct documents (see Table 4.1). Documents are represented by their URL in our data set. We assigned a random id to each user.

In general, the test collection is very heterogeneous according to the number of query and link occurrences on each time unit. The query rate is very low at weekends and holidays. During the main business days of the campus, Monday to Friday, the search behavior underlays high fluctuations. Figure 4.1 depicts both the query frequency and the link frequency per day. The occurrence of all distinct queries and links is counted on each day. On average, 137 queries were performed, and 347 links were viewed per day. The maximum number of queries and links was observed at March 23, 2004. We cannot associate this peak (5,874 queries and 14,917 links) with a known event. Thus, this day is not visualized in both figures.

In addition to the user log, all viewed Web pages are locally indexed. The index was built with the Lucene[1] library which provides Java-based indexing and search technologies. A total amount of 96,586 HTML documents were indexed. The size of the index is 1.08 GB. A link rot rate of 17.03% was measured during indexing.

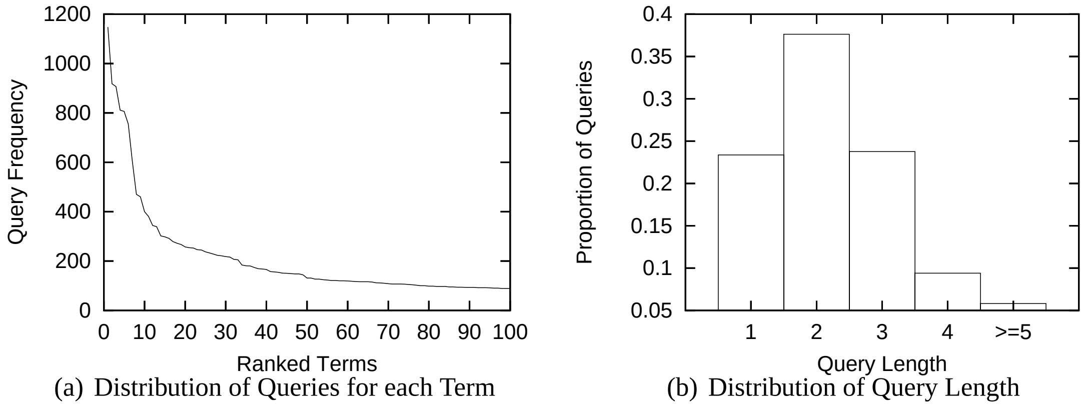
[1] http://lucene.apache.org/, last visit on 2005/08/30.



(a) Queries per Day

(b) Links per Day

Figure 4.1: Request Rates per Day



(a) Distribution of Queries for each Term

(b) Distribution of Query Length

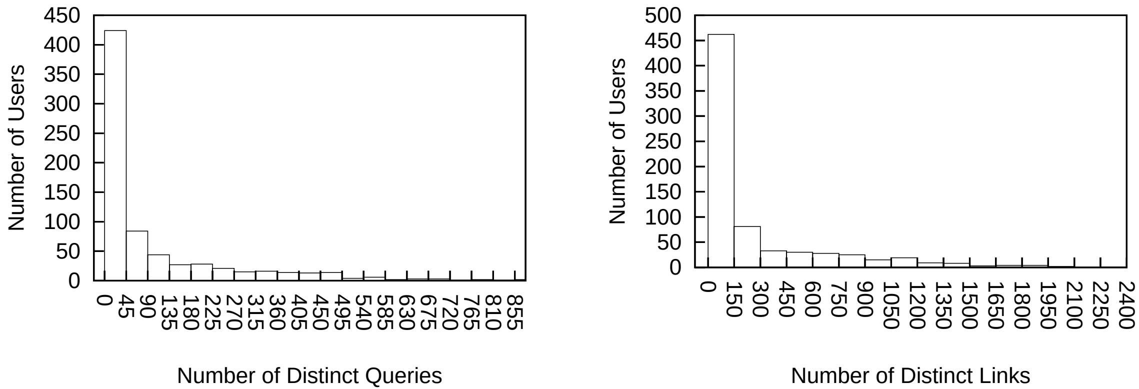
Figure 4.2: Analysis of Individual Queries

4.1.2 Diversity of Search Sessions

The diversity of search sessions was analyzed with respect to queries and links. We investigated the data volume that is produced by individual users in order to estimate if a high-scalable design of the Peer Search Memory is needed. By a grouping of all user sessions per day with the same query and user, we identified 75,692 *grouped search sessions*. For each user's query we grouped a set of links that have been viewed at one day. In particular, we made the following observations in the data set:

- Above 35% of all terms are used in at least two different queries. The query frequency of the 100 most common terms is visualized in Figure 4.2(a). The sample term `in` is the most frequently used query term. It is a stopword in German and English. The terms `bonn` and `download` are the next most commonly used terms which are no stopwords. The frequent usage of these terms shows that people try to avoid terms with a low discrimination of the document collection.

- Figure 4.2(b) illustrates the distribution of query lengths in terms of the number of words. We noticed that 23% of the queries contain only one term, and 38% of the



(a) Queries

(b) Links

Figure 4.3: Distribution of Users according to their Number of Queries and Links

queries contain two keywords. The average length of all queries is 2.41. The maximum number of query terms is 18, and the standard deviation of terms in a query is 1.28.

- Each user had asked on average 101 (stddev 157.16) distinct queries. The maximum number of queries of a single user has been 888 queries. Figure 4.3(a) visualizes the distribution of users according to number of distinct queries they have asked. More than 70% of all users asked less than 90 queries.
- In total, each user viewed on average 256 (stddev 400,46) distinct links during all search sessions. The maximum number of links a user has viewed was 2271. The distribution of the number of users according to their link views is depicted in Figure 4.3(b). Nearly 64% of all users viewed less than 150 links over the time period of 536 days.
- In each grouped session, the number of viewed links for each query by a user is small on average (2.91). Whether a user found a relevant document in this set, can not be automatically derived from the corpus. Figure 4.4 depicts a histogram of link view numbers. It shows what percent of queries have only one page view, what percent have between 2 and 5 page views, and so forth. For 85% of all queries, the users viewed less than 5 links.

All observations show that the Weblog corpus represents a typical Web search scenario where many user have heterogeneous search interests. The distribution of query length is similar to those reported by others: an analysis of major Web search engines revealed that, on the average, each query contained 2.21 terms according to (Jansen et al., 1998) and 2.35 according to (Silverstein et al., 1999). In addition, the term source is quite diverse. With a storage of explicit feedback each user builds a small dynamic information collection. We conjecture that in a real application the number of distinct links will decrease due to explicit relevance feedback. We do not assume that all viewed links of the log file have been relevant for the user. Nevertheless, we see no need of a high-scalable system design of the Peer Search Memory. Even for users with a high search activity of 888 queries and 2271 links over a time period of 536 days, no specific requirement for the local storage device is necessary. The goal of the Peer Search Memory is primarily a storage of associations between queries and links. No documents will be fully stored on a local machine.

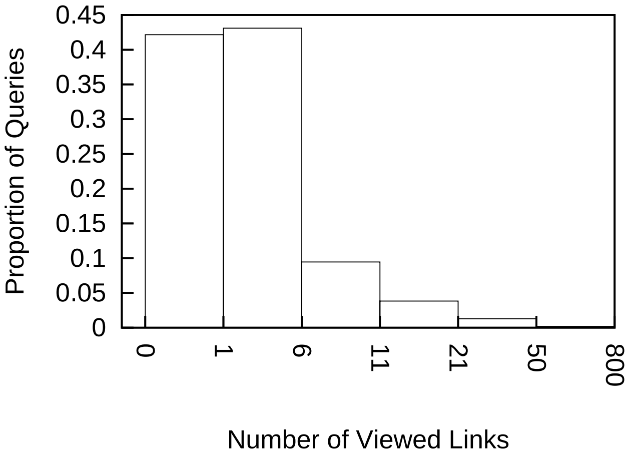


Figure 4.4: Distribution of Viewed Links

4.1.3 Repetitions of Search Sessions

The prior analysis showed that the collection of all search sessions is quite diverse. The Weblog corpus does not provide relevance information from each user, but in general, the more documents a user has viewed the harder it was for him to satisfy his information need rapidly. A personal storage of a search history enables the detection of repetitions. On one hand, it is useful to discriminate between well-known, and new documents in a result list of a Web search engine. On the other hand, repetitions among users enable the system to recommend already validated results from other users. Both types of repetitions are present in the Weblog corpus and must be handled by a Peer Search Memory. We made the following observations in the data:

- Two queries are classified as the same if they have the same terms and word order. We worked on a case insensitive basis. Figure 4.5 visualizes the percentage of queries asked only once, twice and so forth in all search sessions. In all search sessions, we observed that 42% of all queries have been asked only once. Information needs related to these queries have been satisfied effectively.

- The usage of repeated queries in all grouped sessions is very seldom. Only 8% of all queries are used repeatedly on distinct days or by different users. Table 4.2 depicts the 10 most frequent queries. In this listing, we observed that some queries are asked by a large user group. For example, the query `routenplaner` has been submitted by 14% of all users. This query was requested on 98 distinct days. In Figure 4.6(a), a more detailed analysis of this query shows that the number of accumulated queries per day increases faster than the accumulated number of links. The information need represented by this query seems to be very specific for all users. We observed a large overlap of all viewed links. In addition, also unspecific queries can be detected by a plot of accumulated query and link occurrences. For example, the query `lean management` shows a reversed behavior as the query `routenplaner`. Figure 4.6(b) shows a faster increase of the number of accumulated links than for queries. The overlap between link views is very rare in the user group. 66 links have been viewed by 16 users at 10 days.

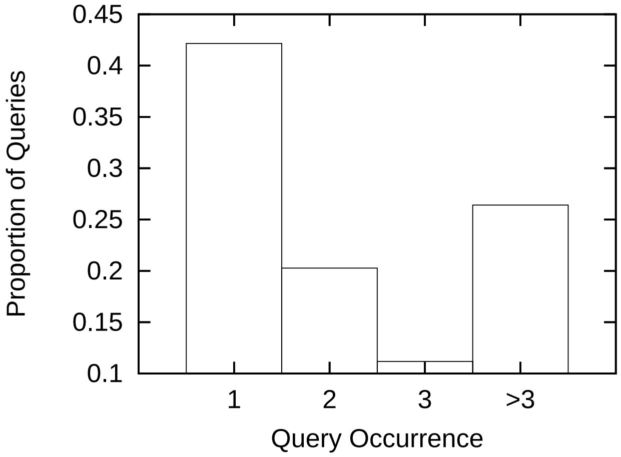
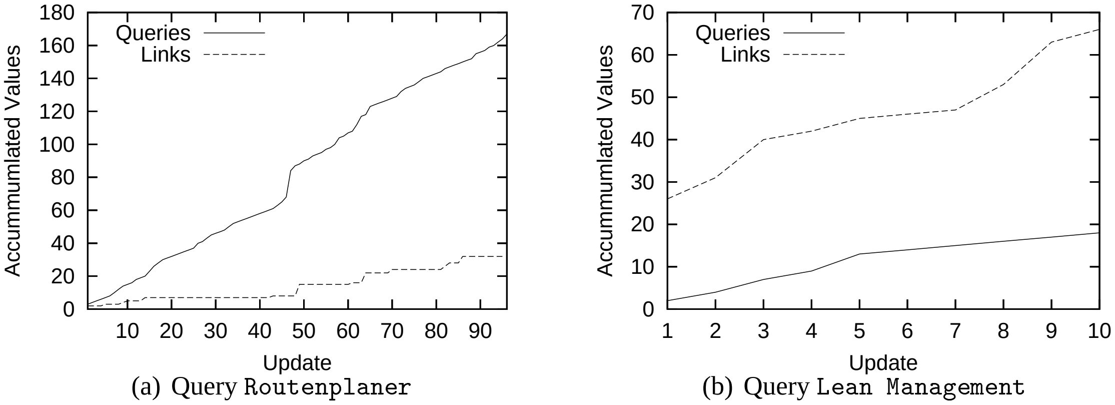


Figure 4.5: Statistics of Query Occurrence

Table 4.2: The 10 most repeated queries on distinct days, and how may distinct users asked the query, and how many distinct links are viewed.

Query	Days	Users	Links
`ksk ahrweiler`	101	11	11
`routenplaner`	96	98	32
`ebay`	55	63	9
`studentenwerk`	49	12	9
`mobile.de`	43	8	4
`telefonbuch`	41	38	12
`wörterbuch`	32	36	7
`selfhtml`	30	22	7
`web.de`	30	34	14
`brocken`	27	1	18



(a) Query `Routenplaner`

(b) Query `Lean Management`

Figure 4.6: Example for Specific (a) and Unspecific (b) Information Needs

Table 4.3: Top 10 most repeated links, and how many queries are related to the link, how many peers viewed the link, and on how many days it was viewed. Only distinct queries, peers, and days are used in the count.

Links	Queries	Peers	Days
http://www.googleadservices.com/pagead/adclick?	996	276	322
http://news.google.de/news?	116	82	82
http://www.chip.de/forum/thread.html?	109	73	84
http://stats.komdat.com/c/track.mdir?	87	82	67
http://comit.eanalyzer.de/index.php3?	83	72	63
http://www.springerlink.com/link.asp?	82	61	63
http://www.amazon.de/exec/obidos/external-search?	73	60	49
http://www.zanox.affiliate.de/ppc/?	71	63	55
http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?	63	30	52
http://www3.interscience.wiley.com/cgi-bin/resolverdoi?	62	46	53

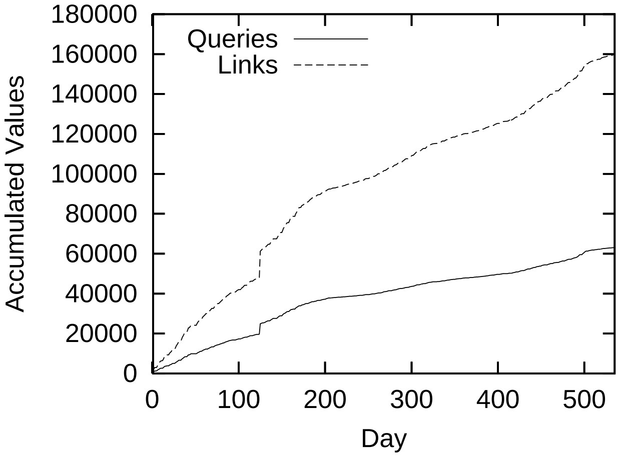


Figure 4.7: Evolution of Queries and Links

- The 10 most repeated links are listed in Table 4.3. All links are ordered by the number of distinct queries that are associated with the links. We observed that all links are general links which are relevant for heterogenous information needs. Their analysis revealed that they are rarely used in the total user group (0.4% of all search sessions). This result emphasizes the need of an individual user history because the statistics of frequently viewed links provides no impact for a user.

- A repeated link is detected if the identical URL is selected at another day. The occurrence of repetitions over a time period of 536 days is essentially in order to analyze the specific type of information need. A query expresses either a short-term or a long-term information need. The temporal distributed of repeated links enables a classification of the particular type. Moreover, the overall evolution of queries and links shows that from a global point of view the search behavior is very unspecific (see Figure 4.7).

In summary, the analysis of the repetitions in all search sessions is high, but on grouped search sessions rather low. A high number of repetitions for particular queries is a sign that the user needs to be assisted. From the diversity of all viewed links we observed a not optimal Web search. Users have to view in 60% of all cases more than one link. We can not verify with the data if the information need has been satisfied after each grouped search session or it had been reformulated. The effectiveness of search can only be increased if the user is assisted during search. The exploratory data shows a specific search behavior that can be assisted by a Peer Search Memory:

- Users have repeated information needs which are expressed by similar queries.
- Users selected links that have been viewed previously for the same query.

With the analysis of the search behavior, we gained new insights over a 536 day period of search sessions. In general, the search sessions are not dominated by repeated interests, neither for the whole user group nor for individual users. Duplicated interests expressed by an identical query or link are rarely used on distinct days or by other users. Otherwise, many queries are asked repeatedly over a short period of time per day. Only 42.15% of all queries have been asked in one search session. In all other cases, the user viewed more than one document in order to find a relevant document.

4.2 Representation of a Peer Search Memory

The Peer Search Memory is a local storage device of search sessions that can be accessed in a personalized manner. The previous analysis of the search behavior showed that each user asked heterogenous topics. In addition, we observed the necessity to assist the user with repeated information needs. From both observations, we elicit conceptual requirements for a Peer Search Memory. In our approach, we use explicit relevance feedback to develop a user profile. This profile is used to merge results from external and internal providers on the client machine.

4.2.1 Conceptual Requirements

The basic requirement of Congenial Web Search is a model of personalized information sources. In the roadmap towards Congenial Web Search, a personalization strategy enables an integration of dynamic and static information collections. For a personalized information source model, two types of s are distinguished:

External Providers External providers such as Web search engines collect a large amount of information. These services provide no information of how the ranked output has been computed. Web search engines actively dynamically collect information with autonomous agents.

Internal Providers All users of the ISKODOR system are denoted as internal providers. Each user maintains a local information source that is dynamically updated with each search session.

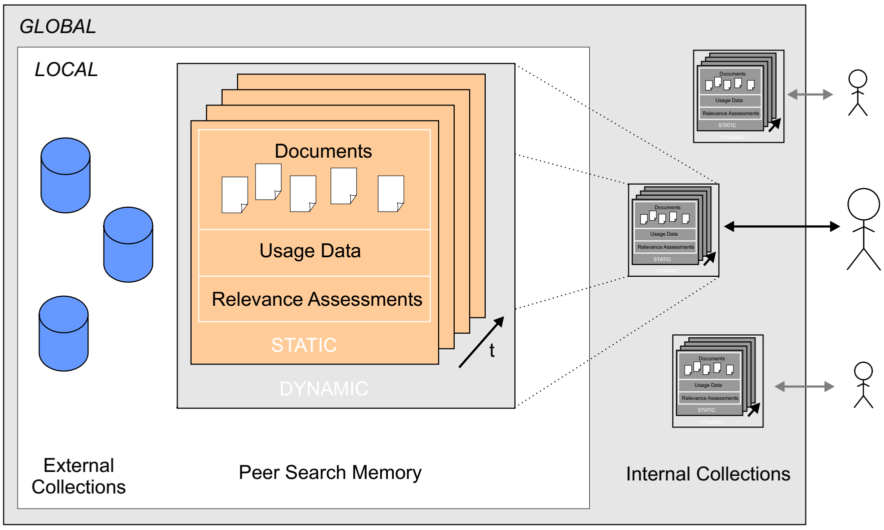


Figure 4.8: Model of Personalized Information Collections

Figure 4.8 depicts the general model of personalized information collections. The *Peer Search Memory* (PeerSy) stores a user profile that incorporates the following information:

1. *User-based Aspects*: For each query session, all documents which have been viewed by the user are stored in the Peer Search Memory. In addition to this click-through data, for all relevant documents satisfying an information need, PeerSy collects user feedback in order to judge a document explicitly.
2. *Content-based Aspects*: All documents in the Peer Search Memory are analyzed in order to build a term index. Furthermore, all terms are weighted. The weights are used to calculate document's has a scores for a personalized ranking.
3. *Usage-based Aspects*: For the dynamic part of the information source, click-through data are collected with a timestamp of the first and the last access of the document.

The collection of internal information collection includes the Peer Search Memories (PeerSies) of all users. Non-functional requirements such as scalability must not be considered for the local storage. The analysis of the search behavior showed that a user with the largest storage need collected 888 queries and 2271 links over a time period of 536 days. With explicit feedback, we expect that the number of links will decrease.

4.2.2 User Profile

The user profile incorporates information about the user's search sessions. The local storage ensures privacy, because no personal information is communicated to a central server. The local profile is transparent for the user at any time. It includes all aspects identified as conceptual requirements:

User-based Aspects All documents are stored in a Peer Search Memory which are relevant in a special context for the user. The *search context* is specified by the query. When a user selects documents from the result list, he can flag these documents from the search result as relevant which have satisfied his information need. This conforms to *explicit relevance feedback*. For each query and document for which relevance feedback has been provided, a is stored. We use the following notation to represent explicit feedback:

- $\mathcal{R}_u$: set of documents for which relevance feedback has been provided by user u
- R_u: number of documents in $\mathcal{R}_u$
- $\mathcal{Q}_u$: set of all user queries
- $\mathcal{R}_{u,q}$: set of documents with relevance feedback for query q

Each document in a Peer Search Memory is a Web page that is represented by a URL. We applied no similarity measure to find synonymous links. For the prototype, all query-document associations are stored in a local database. The detailed representation of the database scheme is discussed in the implementation chapter (see Section 7.1.1).

Content-based Aspects In a second step, a document representation determines both what terms (i) are included and how often they occur (tf_i). Term statistics within the corpus are used to compute a term weight (w_i) for each term i. We use the full text of documents for this task. In a pre-processing step, the language of the document is guessed in order to remove all language specific stopwords. Each Peer Search Memory maintains stopword lists of 11 different languages (Danish, Dutch, English, French, German, Italian, Norwegian, Portuguese, Russian, Spanish, Swedish). Language guessing is performed by comparing each stopword list with the document terms. This language is assigned to a document if its stopword list has the highest similarity to the document. We did not apply a minimal similarity threshold, because the prototype is designed to work only with these languages. We extend the notation of the Peer Search Memory in the following way in order to enhance the user representation:

- $\mathcal{T}_u$: set of terms extracted from all documents in $\mathcal{R}_u$
- $\mathcal{R}_{u,i}$: set of relevant documents for a user u that contain the term i
- $r_{u,i}$: number of relevant documents for a user u that contain term i

With relevance information, the weight for each term can be calculated by modifying BM25 (Sparck Jones et al., 1998), a probabilistic weighting scheme. (Teevan et al., 2005b) investigated a modification of BM25 for Web search personalization. They pursued techniques that leverage implicit information about the user's interest. Their personalization algorithm can significantly improve on current Web search. In traditional relevance feedback the term weight is calculated with the Robertson-Sparck-Jones formula (Robertson and Sparck Jones, 1976):

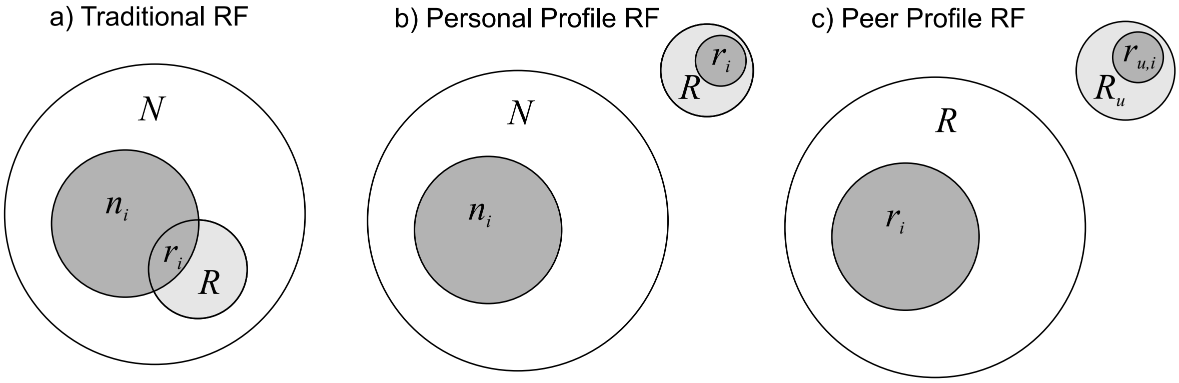


Figure 4.9: Traditional relevance feedback (a) uses relevance information from the corpus. According to (Teevan et al., 2005b), profiles are derived from a personal store (b) with $N' = (N + R)$ and $n'_i = (N_i + r_i)$. In our approach, all peer stores (c) of the users are considered, so $N' = (R + R_u)$ and $n'_i = (r_i + r_{u,i})$.

$$w_i = \log \frac{(r_i + 0.5)(N - n_i - R - r_i + 0.5)}{(n_i - r_i + 0.5)(R - r_i + 0.5)} \tag{4.1}$$

R is the number of documents for which relevance feedback has been provided, and r_i is the number of these documents containing the term i. Figure 4.9a shows that the relevance information (R,r_i) comes from the corpus. N is the number of documents in it, and n_i is the number of documents in the corpus that contain term i. Figure 4.9b shows the concept according to (Teevan et al., 2005b) where relevance information is maintained outside the Web corpus. For the BM25 weighting, they represented the extended corpus with $N' = (N + R)$ and $n'_i = (N_i + r_i)$ in order to include the outside documents. In a similar manner, we use the corpus maintained by all users of our system (see Figure 4.9c).

Both values (R and r_i) can be efficiently estimated within a peer-to-peer network. In Section 2.2.3, Bloom filters (Bloom, 1970) were discussed as a popular technique for an efficient representation of information collections. PlanetP (Cuenca-Acuna and Nguyen, 2002) uses Bloom filters for a disseminates of a peer's index. When the statistics of the network are calculated, we extend the notion of the corpus for the purpose of BM25 to include the documents of all other users. We use $N' = (R + R_u)$ and $n'_i = (r_i + r_{u,i})$ to represent the corpus. These values are substituted in Equation 4.1. After a simplification, we get the following equation for term weights:

$$\begin{aligned} w_i &= \log \frac{(r_{u,i} + 0.5)(R + R_u - (n_i + r_{u,i}) - R_u + r_{u,i} + 0.5)}{(n_i + r_{u,i}) - r_{u,i} + 0.5)(R_u - r_{u,i} + 0.5)} && (4.2) \\ &= \log \frac{(r_{u,i} + 0.5)(R - n_i + 0.5)}{(n_i + 0.5)(R_u - r_{u,i} + 0.5)} && (4.3) \end{aligned}$$

We use these term weights for a personalized search of internal information collections as presented in the next section.

Usage-based Aspects In addition to term weights, we applied a weighting approach that considers all user's repeated accesses to known documents. A user's interest may change over time and documents viewed more recently may give a better indication of a user's current interests. After the first storage of a query-document association, all repeated user accesses to this document are recorded. The number of accesses shows a user's personal significance. The access weight combines the importance of a document with the up-to-dateness of the interest. To calculate the access weight, three parameters of a document are used: (1) number of days when document d has been selected (e), (2) frequency of document d in the access statistics (df), and (3) number of days since the last access of the document d (l). For the design of the access weight, we considered the following characteristics:

- The importance of a document is characterized by how often a user repeatedly viewed (df) the document in proportion to its age ($\frac{1}{e+1}$). Thus, documents which have been viewed very often on the same day ($e = 0$) have a higher importance as documents which have been viewed often in the past. The importance value does not consider that the interest of a user changes.
- The up-to-dateness gives a document a higher weight if it has been viewed lately. The importance of a document, that has been viewed frequently in the past, is reduced by a factor $\frac{1}{\log(l+2)}$. We calculated $l + 2$ in order to avoid that the denominator is zero.

By a combination of both values, we get the following equation for the access weight w_d:

$$w_d = df \cdot \frac{1}{e+1} \cdot \frac{1}{\log(l+2)} \tag{4.4}$$

To summarize, the user profile is build on documents and queries with explicit relevance feedback. In addition, term weights are computed with respect to the document corpus which represents all users' information collections. Access weights are calculated in order to observe how the user works with already known documents. The user profile needs no training phase and is directly applicable for a personalized ranking.

4.2.3 Personalized Ranking

Based on the user profile, a personalized ranking is only applied for repeated interests. The search behavior analysis shows that users selected known links and/or new links for repeated queries in grouped search sessions. We assume that a user pursues one of the following goals:

- *Finding a New Link*: If a user wants to find new results relevant for a former information need, the maintenance of all prior relevant documents enables a selection of all new documents found by an external service. The dynamic changes of this service can be made explicit to the users, when all known documents are marked. This pre-selection enables the user to focus only on new documents.

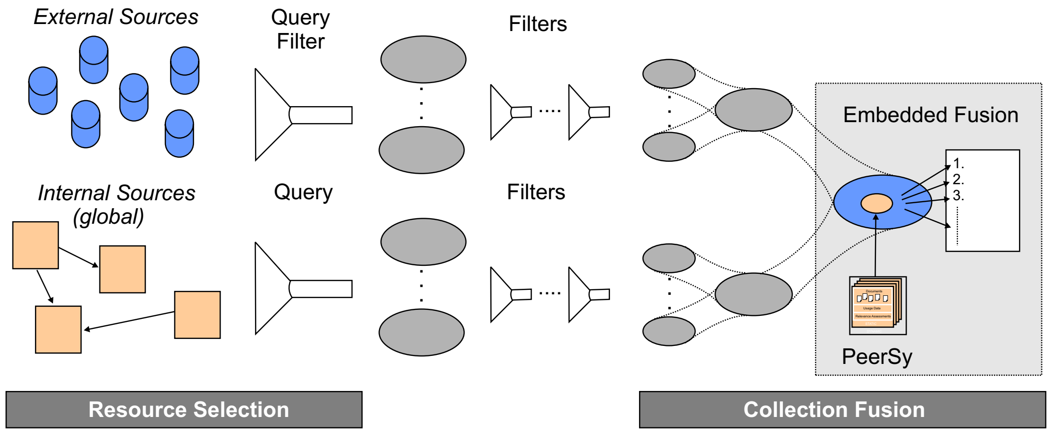


Figure 4.10: General Ranking Process based on Personalized Information Collections

- *Recovery of a Known Link*: If a user wants to recover a known link, the local storage provides a persistent access to known documents. The Peer Search Memory is independent of the dynamics of a Web search index or the ranking of documents. All links are associated with a query that has been used to formulate the information need. The description of this need is very short (2.41 on average) and it can be assumed that the user is able to re-formulated his information need very similar. Thus, the personalized ranking scheme considers all documents for which relevance feedback has been provided and ranks them according to their access weight.

Owing to all former relevance judgements, the user is brought into the focus of the system, and the ranking is adapted to his prior search sessions. Figure 4.10 depicts the general ranking process based on personalized information collections. The resource selection of external information providers is done by the user who selects his favorite Web search engines. For internal information providers, the model of communities is developed for a selection of relevant peers for a query (see Chapter 5). The main attention in this section is paid to collection fusion.

The local retrieval method of the Peer Search Memory is independent of the type of information need as classified previously. The ranking of documents consists of three steps: (1) selection of all documents matching with the query, (2) reduction of the set of documents by several filters, and (3) assignment of a relevance score to all documents. The second step is optional and depends on the system design. In practice, Web search engines provide several advanced search features in order to filter the first result set, although these features are not used widely (Jansen, 2000).

Collection fusion works in two phases. As depicted in Figure 4.11 the collection fusion of both internal and external information sources takes place in two phases. Each phase accomplishes fusion and sorting tasks. In particular, the results are merged by picking up items from the top of the result lists in a round-robin fashion (first item from first list, first item from second list, ..., second item from first list, and so on). After each fusion phase the result list is sorted. First, duplicates are detected by their URL and a checksum. For each result, the number of duplicates is counted. Second, all results are ordered by their number of duplicates in decreasing order. Once all external and all global internal information are

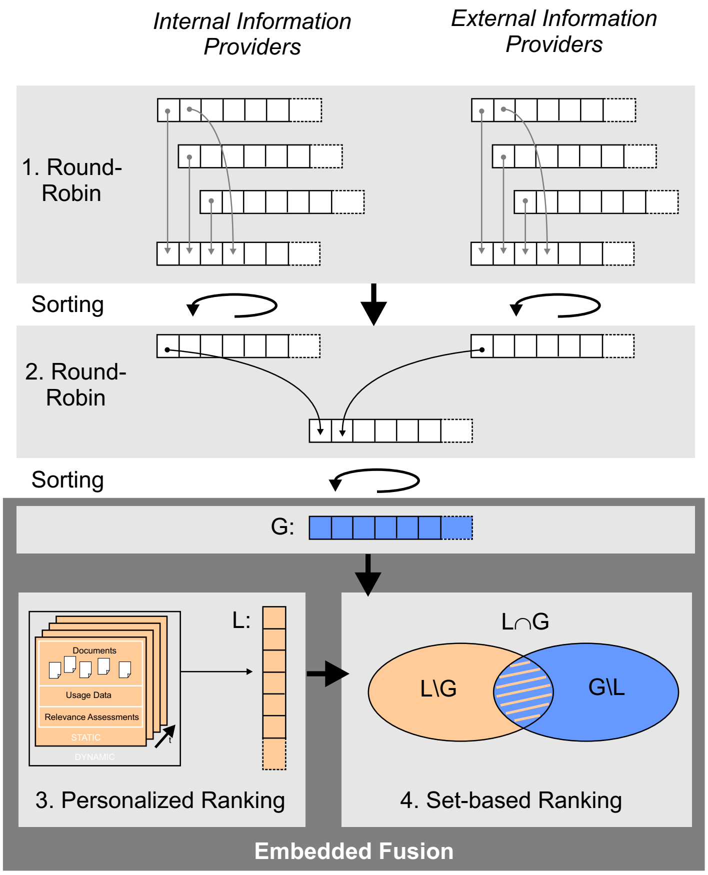


Figure 4.11: Collection Fusion Module

individually fused, in a second phase both results are merged by round-robin and are finally sorted. The result of the global collection fusion is a ranked output G. Owing to the sorting phases, the first ranking positions are assigned to documents commonly found by both types of information collections. In an embedded fusion, this output is merged with the ranked output L of a personalized ranking scheme. The personalized ranking scheme is a tripartite process:

[1st Step] All documents are selected which have been relevant for a former query that is similar to the actual query. The user profile maintains query-focused sets of documents $\mathcal{R}_{u,q}$ for all user queries. The analysis of the search behavior shows that all queries are rather short (on average 2.41 terms). If a query is repeated, we also observed in the Weblog corpus that the user formulates it in a similar way. In order to assist this behavior, all documents are retrieved that have at least one query term in common, because we assume that a user formulates his repeated information need very precisely. The query's score is given by the ratio of the identical terms in q and q^* to all of the terms in q^*:

$$score(q) = \frac{|q \cap q^*|}{|q^*|} \tag{4.5}$$

We select all sets of relevant documents $\mathcal{R}_q$ with a query score $score(q) > 0$. To construct the ranking, all sets of documents $\mathcal{R}_q$ are ranked by the $score(q)$. This ranking does not yield a completely ordered list. Thus, for each document of $\mathcal{R}_q$ the access weight is considered in the second step.

[2nd Step] In case a set $\mathcal{R}_q$ matches with an actual query q^*, all documents of this set are ranked according to their access weight (see Equation 4.4). All duplicates of a document are removed from sets with a lower query score.

[3rd Step] Finally, a document's score for all local documents is calculated by summing over the query terms ($\mathcal{T}_{q^*}$), the product of the query term weight (w_i) and the normalized query term occurrence in the documents (tf_i). We do not normalize the query term occurrence. In analogy to the BM25 ranking function (Sparck Jones et al., 1998), we get the following equation:

$$score(d) = \sum_{i \in \mathcal{T}_{q^*}} \frac{tf_i}{\frac{dl}{avdl} + tf_i} \cdot w_i \tag{4.6}$$

where dl is the document length, and $avdl$ is the average document length in the corpus. All documents are ranked according to this score. This ranking list is appended to the previous ranking list resulting from step 1 and 2. Similar to step 2, duplicates are removed if they also occur at lower ranking positions.

The result of the personalized ranking scheme is a ranked output L of documents which are relevant for a user, and which have been found in prior search sessions. After all matching documents have been identified in local query-document associations, these documents are merged with results of G. The fusion of a local PeerSy and other providers is characterized by three sets: (1) $L \cap G$: a set of documents found by PeerSy and other sources, (2) $L \setminus G$: a set of documents found only by PeerSy, and (3) $G \setminus L$: a set of documents found only by all other sources. The first two sets are ranked according to our personalized ranking scheme of L. For the last set, the ranking order of G is used, because no re-ranking is applied. All sets are presented to the user in a set-view style as discussed in Section 7.4. A user can individually focus on this set of documents he indented to search. For example, a user can easily see which new documents have been found in the document collection since his last search.

4.3 Evaluation

This scenario evaluates the impact of a local feedback storage. The evaluation scenario is designed to answer the question: *Does PeerSy support an effective search for repeated queries by personalized ranking?* The evaluation task reflects the advantage of a persistent storage of explicit feedback. We simulated a personalized ranking with the Weblog corpus.

4.3.1 Known-Item Retrieval Settings

The evaluation task has the goal to measure the effectiveness of a personalized ranking for repeated information needs. The advantage of a local feedback storage is measurable for the single user when a query is repeated and a known document is recovered. The effectiveness of the recovery is measured with a known-item search. For this task, the Weblog corpus was preprocessed in order to collect all grouped sessions of a user that have repeated queries and links. We collected pairs of succeeding grouped sessions which have been classified according to their number of repeated links:

Single Known-Item Retrieval The first example shows that for a repeated query, only one link is selected repeatedly in grouped sessions. For example, the query `tour de france` has been repeated three times, and at each repetition the same link was selected. We conjectured that the information need was very specific and only one link exists which has a high relevance for the user.

Multiple Known-Item Retrieval The second example represents a class of queries where several links were relevant for the user, because they are selected repeatedly. For example, when the query `mp3 download` was submitted the first time, four distinct links were selected. Three days later, the user repeated the query and only three of the previous seen links were chosen again.

In total, we extracted 879 repeated queries of both known-item retrieval classes from the Web log. 34% of all users have asked at least one repeated query. In the set of all grouped sessions, 75% of all sessions belong to the first class and 25% to the second. Both classes enabled us to define different evaluation tasks:

- For single known-item retrieval, we evaluated how the dynamics of an information collection assists individual needs. A baseline is computed for all grouped sessions of this class. In comparison, a Web search engine has been requested in order to investigate the dynamic change of results within the first 20 results.
- For multi known-item retrieval, we evaluated the effectiveness of the personalized ranking. The original selection behavior is compared with a simulated ranking. We simulated the personalized ranking scheme and a vector-space ranking.

We applied different evaluation metrics for both tasks.

4.3.2 Evaluation Metric

The effectiveness of known-item retrieval depends on two aspects: the ranking of the search engine and the mental ability of a user to preprocess the result list in order to detect relevant information. Usually, known-item retrieval is evaluated in a user-independent setting. The evaluation setting simulates a user who processes successively a result list in order to find the relevant document. The search process is not efficient the more nonrelevant documents must be viewed before finding a relevant one. Standard measures like precision and recall are not applicable for the known-item search. If a user repeats a query, all associated documents to this query will be retrieved that are judged previously as relevant by the user in a Peer Search Memory. Instead, we selected two different measures for each evaluation task:

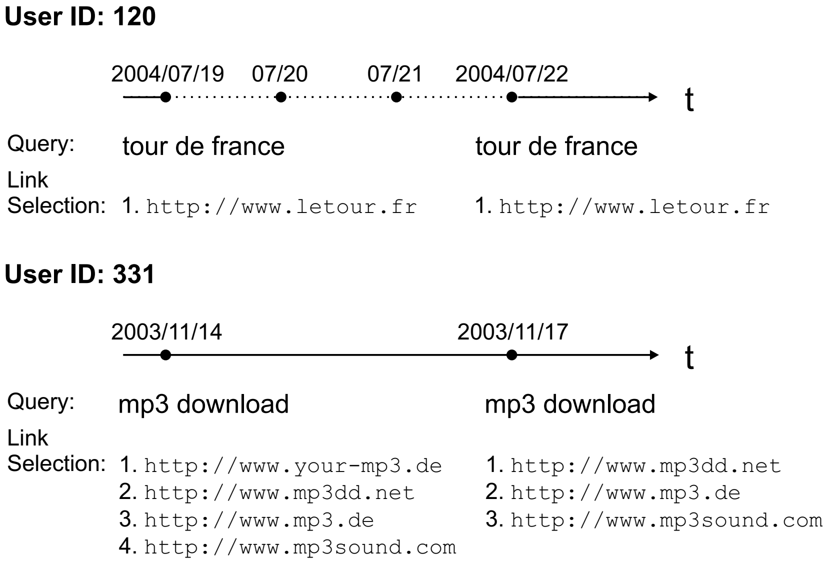


Figure 4.12: Two Examples for Duplicated Queries and Links

Metric for Single Known-Item Retrieval Performance of the single known-item retrieval setting is measured by the rank at which the desired link appears in the list of viewed links. The *average rank* for a set of queries $q_1, \ldots, q_n$ and relevant links $l_1, \ldots, l_n$ is

$$\overline{\text{rank}} = \frac{1}{n} \sum_{i=1}^{n} \text{rank}(l_i) \tag{4.7}$$

Another measure is the *inverse average inverse rank* that is the harmonic mean of the rank at which the desired document occurs. This measure is defined as

$$\text{IAIR} = \frac{n}{\sum_{i=1}^{n} (\text{rank}(l_i))^{-1}} \tag{4.8}$$

Both measures, average rank and inverse average inverse rank, score 1.0 for perfect retrieval. These values increase if the system returns the desired document late in the result list.

Metric for Multi Known-Item Retrieval Average rank and inverse average inverse rank cannot be applied to the multi known-item retrieval setting. For each query that belongs to this setting, a set of links have been viewed repeatedly by the user. The order of the links is defined by the timestamp of the search session. This ranking represents the real selection behavior of the user. We examined how similar this ranking is to s personalized ranking. The similarity of rankings is measured with the *Kendall tau distance* (Adler, 1957). It is the number of pairs of links that appear in opposite order in the two rankings. We normalize the measure with the maximum possible disagreements. The Kendall tau distance is 0 when the two rankings are exactly the same, and it is 1 when the rankings are in reverse order. Two random lists have a distance of 0.5 on average (Teevan et al., 2005b).

Table 4.4: Results of Single Known-Item Retrieval.

method: date (range):	Baseline 2003/19/07 - 2005/08/04	Google 2005/09/15
IAIR	1.081	3.296
$\overline{rank}$	1.274	14.753
no. queries (rank =1)	70.74%	24.50%
no. queries with link not found	-	67.32%

4.3.3 Evaluation Results

Results of Single Known-Item Retrieval: We used 661 queries that have been requested repeatedly. In total, 2288 grouped sessions are found for all repeated queries and the inverse average inverse rank is computed for each repeated link. In our data, we observed the following aspects:

- In Table 4.4, the baseline results show that the user is very effective in recovering a known-item. Although, this observation cannot be assigned as a quality measure for the Web search engine. The user may not have viewed documents according to the original ranking. Nevertheless, the combination of a Web search ranking and a human selection behavior is very effective at the time of the repetition.
- In comparison to the baseline, we measured the dynamics of a Web search ranking (in our case Google) independent of a human preprocessing. All queries of the single known-item retrieval task are processed by Google in order to get the rank of a relevant link in the actual result list. If the document was not found in the first 20 results, the rank was set to 21. On average, the known link is now found on the 14. ranking position. The result shows that the user has to switch in 70% of all cases to the second result page in order to recover the known link.
- The increase of the IAIR value of the Google setting shows that the ranking of the Web search engine might have changed due to the continuous update. At the time of a repetition in the Weblog corpus, the ranking of the Web search engine was more optimized for the individual needs of a user. Over the time, the information collection of the Web search engine is highly dynamic, and its ranking will have been optimized for the majority of users who favored offer documents.
- For a more user-centered evaluation, we explored queries which occurred in many grouped sessions. Table 4.5 (left columns) lists five queries with the highest number of repetitions. We observed a low average rank for those queries. They consist only of one popular term or phrase. The results show that during all repetitions the information need could be satisfied rapidly. We conjecture that results for these queries do not depend on the dynamic shifts of the information collection. All search services are optimized for a majority of users searching for the most authoritative pages. We expect that there is a manageable number of known links, for example for route planers, on the first result page. It depends on the users individual choice which is his favorite service.

Table 4.5: Selection of repeated queries. On the left side, queries are ordered by the number of grouped sessions (GS) in which they occur. On the right side, queries are ordered by their average rank in descending order.

query	no. GS	$\overline{rank}$	query	no. GS	$\overline{rank}$
`routenplaner`	39	1.08	`ferienhaus plattensee`	1	10.00
`ebay`	10	1.08	`tribulus terrestris`	1	9.00
`freenet`	10	1.00	`geschichten`	1	8.00
`telefonbuch`	9	1.00	`lineare optimierung`	1	7.00
`tour de france`	9	1.00	`model bewerbungsformular`	1	7.00

Table 4.6: Results of Multiple Known-Item Retrieval.

method:	VSM	PeerSy
avg. τ	0.61	0.08
no. queries (τ =0)	28%	87%

- In Table 4.5 (right columns), queries are investigated that have a high average rank. For example, a user has viewed 9 different links until he recovered the well-known link for the query `ferienhaus plattensee`. All queries with a high average rank are repeated only once. On one hand, the user might not explicitly tried to recover a known link. He was searching for new relevant results, and he did not remember a already seen document. On the other hand, for queries that are not frequently requested or all users viewed a diverse set of links, the Web search engine is not able to assist individual needs due to its dynamic collection.

Results of Multiple Known-Item Search: If several links have been viewed by a user repeatedly, we identified a multiple known-item search. The personalized ranking of these documents is compared with the order of the real user's selection behavior. We explored 345 queries and the personalized ranking of all repeated links in comparison to a baseline method:

- The baseline method ranks all known documents according to a vector-space search. The results in Table 4.6 show the average Kendall-tau distance for the baseline and the PeerSy ranking. The distance between the real selection behavior and the baseline ranking is 0.61. The result for the PeerSy ranking, $\tau = 0.08$, shows that it is closer to the individual choice of the user.

- In a second step, we analyzed how often both rankings are exactly the same ($\tau = 0$). For the vector-space search, in 28% of all cases its ranking matches with the real selection behavior. Instead, for 87% of all queries, the personalized ranking is exactly the same as the user's selection.

- For example, the personalized ranking for the sample query `mp3 download` on November 17th, 2003 has an optimal match with the real selection behavior. We conclude for all queries with an Kendall-tau distance greater 0 that it is feasible for the user to compensate a reverse order. The result sets from the Peer Search Memory are small compared to Web search engine results. It can be assumed that once a document is selected as relevant, this judgement changes over time. A document that was viewed in the last session may have a higher relevance than an older one. Such aspects cannot be derived from the Web log data. The personalized ranking scheme is only affected by usage data which is collected due to click-through data.

4.3.4 Discussion

Whether a known-item refers to a relevant document cannot be analyzed without explicit feedback. This is a limitation of the Weblog corpus, as well as for each Web search engine that analyzes usage data. The repeated selection of well-known links was used to derive implicit relevance information. It was applied to a local known-item retrieval task in order to measure the effectiveness of personalized ranking for repeated queries. The evaluation task confirmed the assumption that a persistent storage of feedback supports a constant ranking of the user's prior interests. The combined retrieval of local feedback and a Web search index allows an effective pre-selection of documents for users according to their actual needs. In addition, personalized ranking accommodates the individual selection behavior because the number of viewed nonrelevant documents is decreased. The IAIR value of the baseline method for single known-item retrieval shows that the user's strategy to recover a relevant document is very effective on average. It depends on the type of information need, if a document can be easily recovered on the first result page. We expected that this ranking is not the same as it was at the original request time. We observed the impact of a human selection and the stability of a ranked list over the time. PeerSy has a high stability in its ranking because of a persistent storage of explicit relevance feedback. The goal of a PeerSy is an individual optimization. Each single user must be supported to satisfy his information needs. Thus, the whole user group is involved in this task. When particular queries are asked by a user for the first time, a support can only be facilitated if the whole user group is considered. The test corpus shows that exact query matches between users are rare. It was difficult to find two users with very specific interests, e.g. paperfolding, due to the geographical restriction. In such a case, it can be assumed that they know each other personally.

It is a limitation that for particular long-term interests of a user no support can be achieved by the user group. This restriction of the evaluation scenario does not reveal new insights about the impact of collaboration among users. For this purpose, a community concept is described in the next chapter.

4.4 Summary

This chapter designed a dynamic information collection based on explicit feedback. The model of personalized information collections comprises both static and dynamic features. The analysis of the search behavior indicated that a user asked on average 101 queries during a 536 day period. 42% of all queries were asked only once. We conjectured that many queries are very specific, due to the low number of viewed links. The data showed that only one or two documents have been viewed for a query in 62% of all cases. All click-through data is stored as usage patterns in a Peer Search Memory in order to provide a personalized access. This information is enriched by explicit relevance feedback. Owing to a temporal organization of usage data, a stream of documents which have been found in the time between two particular search sessions, can be offered to a user. This offset is managed locally for each user in order to accomplish a dynamic information collection. An integrated representation of both dynamic and static features assists the information retrieval process. The local search engine applies a user profile to perform a personalized ranking of Web pages. The impact of personalized ranking was evaluated with known-item retrieval and implicit feedback. We selected all search sessions with repeated queries and links. The baseline was defined by the combination of a human result selection and a ranked output of a Web search engine. A repeat of each query showed a marked impairment of the retrieval effectiveness. If multiple documents have been viewed for a repeated query, the results showed that the personalized ranking performs nearly optimal. We conjectured that this effect will be even more pronounced with a test collection based on explicit relevance feedback.

The Peer Search Memory is an essential part of the Congenial Web search concept. All users are able to maintain a local information collection with validated results. With a Peer Search Memory, each user takes the role of an information provider. The user profile enables a personalized ranking for repeated information needs. New documents can be easily detected in the result set of information providers. The local user collection does not advances the search for new information needs. Queries that are previously asked by other users can be exploited for an advanced retrieval. Previous validated results can be considered to organize users with common interests. The model of communities addresses this necessity to further enhance the concept of individual information collections.

5 Modelling Communities

This chapter represents the second pillar of Congenial Web Search, and it describes the implicit collaboration strategy. Collaboration among users can be initiated based on individual user feedback. In order to discover similar search interests, common judgements are grouped by context. A Virtual Knowledge Community is the result of a context grouping, and an automatically supervised expansion of queries, links, and their associations. Users can explicitly join a community which facilitates advanced search and filtering processes. Evaluation settings are designed to measure the quality of a grouping of queries and links.

5.1 Conceptual Requirements

The process of modelling communities is based on a decentralized storage of explicit relevance feedback. The Peer Search Memory is a local information collection for each user. A personalized ranking assists repeated information interests locally and in case of new information needs, a user relies on external providers with a large index or on internal providers with relevant query-link associations. The overlay architecture of the peer-to-peer network is exploited in order to retrieve documents from other users. In general, it is not efficient to select all resources to process a single query. Observations of search engines showed that the frequency of popular queries conforms to the Bradford distribution (Brookes, 1977). Hence, there will be few topics requested by a huge number of people, and numerous topics are requested very little, if at all. Based on this assumption, a resource selection process must be designed to work with the Bradford distribution as discussed by Bates (2002). In particular, such a process must handle all queries distributed in the network equally. The routing of information requests should not be dominated by information needs of the majority. On one hand, general requests are very popular for a majority of users. On the other hand, information needs with a high specificity are requested only by a minority of users. In order to handle both types of requests, a central structure of all local interests is required.

Requirement 1: All local interests must be prepared for a global access in order to structure common interests. Each peer retains the autonomy of the maintenance of its local associations. A component is necessary that synchronizes all local interests efficiently. This process must be independent of the availability of single peers.

The first requirement defines a global structure which is essential for the detection of common search interests. A synchronization in a decentralized manner possesses a storage overhead for each peer and is difficult to maintain. Hence, a hybrid architecture combining centralized and decentralized entities is designed in order to facilitate the synchronization of all public search interests. The result of the synchronization enables a detection of *collaborative information needs* which are requested frequently of a group of users. They allow an effective resource selection if peers are organized in peer groups. In addition, these groups enable a self-organization of peers for a topic-driven query routing. For this purpose, a highly-flexible process for the detection of common interests based on the global set of explicit relevance feedback is required.

Requirement 2: Explicit feedback information must be processed to detect similar interests. The discovery process has no training phase in order to be independent from the number of users. Each local user feedback is treated as a subjective assessment depending on the actual state of knowledge at the time of request. The ambiguity of relevance feedback must be handled before users are assigned to similar search interests.

The analysis of the search behavior showed, that users have similar interests which are formulated with repeated queries. In addition, link are viewed repeatedly for the same or a new query. The ambiguity of a user's assessment results from the query similarity, as well as from the document similarity. With each feedback, an overlap of two search sessions can be measured. If two users provided feedback for the same document that has been found for the same query, we assume that they have similar interests. For the discovery of these interests, the ambiguity must be handled before users are informed about a community of interest. The grouping of users with respect to collaborative information needs requires a transparent organization in order to gain the acceptance of the users.

Requirement 3: Collaborative information needs require a transparent organization of all associated users. No implicit grouping can be performed, because a user needs to acknowledge his interest. Each user must explicitly commit a membership to a community which represents common interests of a user group. The proposal of a community must be transparent for a user in order to recognize his contribution to this group.

The explicit commitment of a community membership preserves the autonomy of each peer. In particular, identifying a user by his specific memberships must be prevented. Once a community has been established, all future assessments of the user must be organized with respect to existing communities. On one hand, queries which are related to a community must be associated to it in order to complement the stored associations. On the other hand, new memberships must be offered to users who show interest in the topic of a community.

In order to summarize all requirements, a model of communities is required that is based on relevance feedback stored in a decentralized manner. First, all search sessions are synchronized and mapped by a central entity to collaborative interests. The discovery of these interests leads to a self-organization of peers into peer groups. Second, each group requires the confirmation of all associated users. With the confirmation of a community membership, a user provides explicit feedback that the topic of the community is of long-term interest for him. The basic characteristics of a peer-to-peer network such as transparency, self-organization, and autonomy support all conceptual requirements.

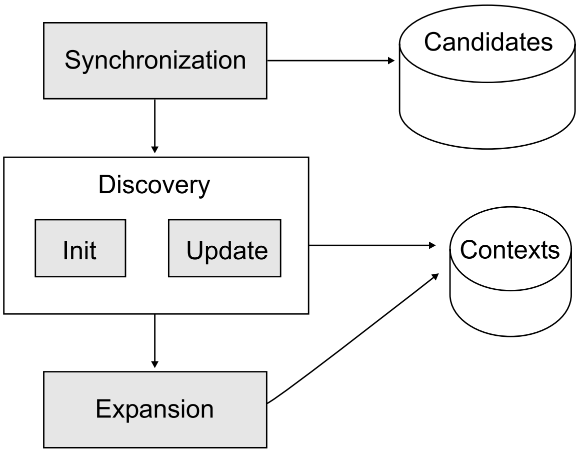


Figure 5.1: Discovery Process of Collaborative Search Contexts

5.2 Discovery of Collaborative Search Contexts

The discovery of collaborative search interests in all internal sources is an automated process in order to select communities. To do so, similar information needs are discovered by an accumulation of all Peer Search Memories. Such a strategy requires a high flexibility due to the dynamic of all internal information providers. In particular, similar search interests comprise three important characteristics:

1. They capture overlapping interests of users by assigning a context to each search session.
2. They identify relevant documents for a set of query terms with a high commitment by a group of users.
3. They build the basic structure for collaboration in personalized information sources.

The process of discovery is divided into three processes. First, a synchronization process initiates a preprocessing of all distributed search sessions. Second, the main processing phase is the analysis of synchronized feedback information. This phase initializes a new context or it updates an existing one. Third, a context is expanded with similar collaborative information needs in a final postprocessing phase. All processes are described in the succeeding sections.

5.2.1 User Feedback Synchronization

The first step in order to discover similar search interests is the synchronization of user feedback. The model of personalized information sources includes no central log file due to privacy reasons. Each user decides which search sessions are available to other users. As all internal information sources underly a continuous update depending on the activities of the peer, a synchronization technique must fulfill three requirements: (1) management of the temporal order of all search requests, (2) uniform representation of all search requests, and (3) dynamic update of a previous synchronization. The synchronization strategy can be developed with respect to the characteristic of proxy logs. Each log entry consists of a timestamp, user id (IP address), and click-through data (HTTP request). A post-processing of such a log file extracts query sessions including all judged documents for a particular query. Owing to the distributed storage in the personalized search model, such log entries are extracted from each Peer Search Memory locally. The user identification can be assembled by a unique peer id. In particular, each query and the documents which have been assessed as relevant by a user define an individual *search context*.

For the management of the temporal order of individual search sessions, we do not apply a time synchronization at all peers. After a user provided relevance feedback for a document, the query-document association is sent to an external community discovery service as discussed in Section 3.2.2. It assigns each association an actual timestamp. The concurrency control of the central entity organizes the successive processing of each user-driven feedback stream. The enhanced context is denoted as *timestamped context*.

Definition 5.1 *A timestamped context of a user u is a 4-tuple (q, l, s, u) where q is a query with a relevant Web page (URL) l at time s.*

All timestamped contexts are pre-processed by a user-based query expansion. For each user, the 10 last distinct query terms are maintained by the service. A new query q^* of a timestamped context is expanded in order to detect previous similar search interests in the individual set of former query terms. This step only considers the similarity based on string matching without any semantical similarity in order to notice small changes in the spellings (e.g. spelling mistake) of a query term in previous sessions of a user.

Similarity between query terms is determined by the Levenshtein distance that is a measure of the similarity between two strings, which we will refer to as the source string (s) and the target string (t). The distance is the number of deletions, insertions, or substitutions required to transform s into t. The more different the strings are the greater is the Levenshtein distance. In our case, we use the maximal length of the two terms to divide the Levenshtein distance in order to constrain the value within the range of $[0, 1]$. The similarity of two strings is inversely proportional to the normalized Levenshtein distance:

$$sim_{lev}(s,t) = 1 - \frac{Lev\_distance(s,t)}{max(|s|,|t|)} \tag{5.1}$$

For all terms of a query $q^* = \{s_1, \ldots, s_m\}$, we compute the normalized Levenshtein distance to the last 10 distinct query terms $\mathcal{T}_{10,u} = \{t_1, \ldots, t_{10}\}$. The expanded query q^*_{exp} is represented by all terms of the query q^* and the set of new query terms that are similar based on a query expansion threshold δ:

$$q^*_{exp} = q^* \cup \{t \in \mathcal{T}_{10,u} | \exists s \in q^*, sim_{lev}(s,t) > \delta\}$$

If no similar terms in the previous sessions of a user are detected, the expanded query q^*_{exp} is identical to the original q^*. After the expansion, a timestamped context is parsed into its elements. The community discovery service maintains all timestamped context that are not grouped in a community. All remaining contexts are collected in a *candidate set* $\mathcal{N}$. It is updated at time unit s_n with a new timestamped context (q^*_{exp}, l^*, s_n, u).

$$\mathcal{N}_{s_n} = \mathcal{N}_{s_{n-1}} \cup \{(t, l^*, u) | \forall t \in q^*_{exp}\} \tag{5.2}$$

For each term of the expanded query q^*_{exp}, an association with the relevant link and the user is stored in the candidate set. Stopwords have been removed from the query, because a frequently used term allows no discrimination among contexts. The set characteristic of $\mathcal{N}_{s_n}$ allows no duplicated storage of identical associations by a user. Repeated judgments do not initiate a feedback re-synchronization.

5.2.2 Context Instantiation

The model of personalized information sources regards three main items expressing the context of a search session. First, the *query context* is described by assessed documents (links). Second, query terms that are used to retrieve relevant documents describe the *link context*. Third, assessed documents with associated queries describe a *user context*. Cross-references between context types are used to measure the similarity of two timestamped contexts. For each incremental update of a context, three similarity principles are considered.

Principle 1 Two queries p and q are similar at time s_n, if identical links have been assessed.

Principle 2 Two links l and m are similar at time s_n, if identical queries have been assessed.

Principle 3 Two users u and v are similar at time s_n, if identical queries or links have been assessed.

All three principles are used to define a collaborative search context. It depends on the periodicity of an information need when similar queries, links, and users are grouped. A new collaborative search context is initialized if a minimal agreement of a user group with the new timestamped context (q^*_{exp}, l^*, u) is observed. The initialization is restricted by two parameters:

- ρ: minimal number of identical query terms that are associated with the same document l^*
- σ: minimal number of users who have provided feedback for the same document l^*

Both parameter, ρ and σ, are used to define the minimal of agreement in a user group. They restrict the number of collaborators in the candidate set with respect to a timestamped context. These users and their shared interests initialize a collaborative search context.

Definition 5.2 *For all users $\mathcal{U}$ in the candidate set $\mathcal{N}$, a subset of users $\mathcal{U}_c$ with collaborative search interests to a new timestamped context (q^*_{exp}, l^*, u) of user u is defined by*

$$\mathcal{U}_c = \{v \in \mathcal{U} \mid \exists \mathcal{T}^* \subseteq q^*_{exp} : |\mathcal{T}^*| \geq \rho \wedge \forall t \in \mathcal{T}^* : (t, l^*, v) \in \mathcal{N}\}$$

If the set $\mathcal{U}_c$ of collaborators for a timestamped context (q^*_{exp}, l^*, u) is greater than the threshold σ ($|\mathcal{U}_c| > \sigma$), it is used to initialize a collaborative search context. Otherwise, no further process to discover a collaborative search context is initiated.

Definition 5.3 *A collaborative search context C is a 4-tuple ($\mathcal{T}_c, \mathcal{L}_c, \mathcal{U}_c, \mathcal{A}_c$) with*

- *$\mathcal{T}_c$: a set of query terms*
- *$\mathcal{L}_c$: a set of links*
- *$\mathcal{U}_c$: a set of users with $\forall u \in \mathcal{U}_c$, $t \in \mathcal{T}_c$, $l \in \mathcal{L}_c$, and $(t, l) \in \mathcal{N}$*
- *$\mathcal{A}_c$: a set of associations based on a 3-tuple (t, l, u) with $t \in \mathcal{T}_c$, $l \in \mathcal{L}_c$, and $u \in \mathcal{U}_c$*

Each initialized collaborative context is a subset of the candidate set $\mathcal{N}_{s_n}$ at time s_n. All tuples that are used to initialize a collaborative search context C are removed from the candidate set. For each new timestamped context, its similarity to collaborative search contexts is measured for the purpose of updating.

5.2.3 Incremental Context Update

The initialization of a collaborative search context is the first step towards a community. The goal of an incremental update is a merge of a timestamped context of user u at time s_n with already found collaborative search contexts. We defined two measures to calculate the similarity between a timestamped context and a collaborative search context. In absence of any information other than the timestamped context, the measures consider the following aspects:

1. *Query*: Similarity of queries can be computed in different ways based on the representation of the query content: keywords, phrases, or word order. A context C is similar to an expanded query q^*_{exp} if the query terms are similar to the context specific terms $\mathcal{T}_c$.
2. *Link*: Link similarity can be computed either based on the content or the link structure. A link can be relevant for several contexts. In particular, links with a low specificity belong to multiple contexts, and links with a high specificity are constrained to one.
3. *Users*: Similar users are characterized by common interests and habits. A user has a high context similarity if he already contributed to this context. Two users are similar if an overlap of their local information collections is measured.

4. *Request Time*: The probability that two queries are asked at the same time can be either unconditional or conditional. Depending on the periodicity of an information needs, some queries are likelier to be requested at the same time than others. The request time cannot be used as a single measure of temporal context similarity.

The previous list shows that a user similarity and a temporal similarity are not independent of a query similarity and/or a link similarity. We developed two different similarity measures in order to evaluate their impact on the community process.

Term-based Similarity This measure considers a term-based similarity between the set of terms $\mathcal{T}_c$ of a context C and the expanded query q^*_{exp} of the tuple (q^*_{exp}, l^*, u). In analogy to the vector space model, the content of a context C is represented by a term vector space TVS$_c = \mathbb{R}^{|\mathcal{T}_c|}$. A context C is assigned a term-based context vector $\overrightarrow{ct} = (w(\mathcal{T}_c, t_1), \ldots, w(\mathcal{T}_c, t_{|\mathcal{T}_c|})) \in$ TVS$_c$. In addition, a query vector

$$\overrightarrow{q^*_{exp}} = (w_q(q^*_{exp}, t_1), \ldots, w_q(q^*_{exp}, t_{|\mathcal{T}_c|})) \in \text{TVS}_c$$

is assigned to the expanded query of the timestamped context. Since all terms of a context and queries share the same representation, the cosine of the angle between the query vector and the term-based context vector is used to determine the similarity:

$$sim_T(ct, q^*_{exp}) = \frac{\overrightarrow{ct} \bullet \overrightarrow{q^*_{exp}}}{\|\overrightarrow{ct}\| \times \|\overrightarrow{q^*_{exp}}\|} \tag{5.3}$$

As a weighting function for the term-based context vector, we applied the following weighting scheme

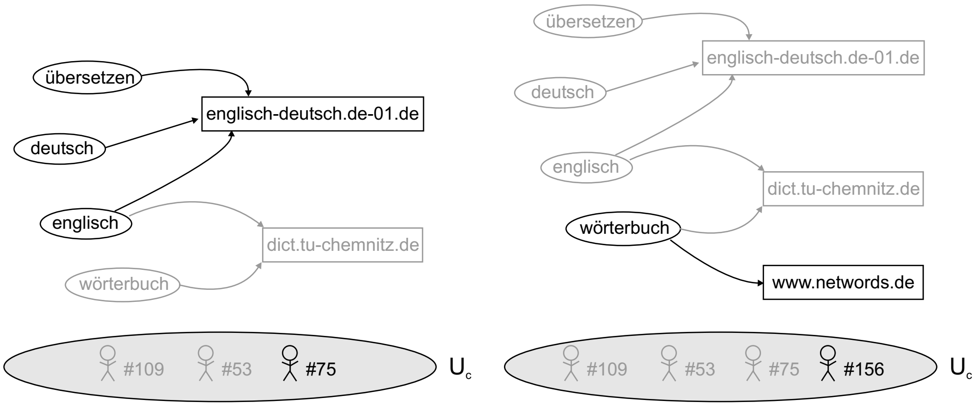
$$w(\mathcal{T}_c, t) = tf(t) \cdot iqf(t) \tag{5.4}$$

where the term frequency $tf(t)$ is the number of times the term t occurs in the set of associations $\mathcal{A}_c$. $iqf(t)$ is the *inverse query frequency*

$$iqf(t) = \log \frac{|\mathcal{Q}|}{q_t} \tag{5.5}$$

where $|\mathcal{Q}|$ is the total number of queries that have already been processed by the community discovery service, and q_t is the number of times term t is used in a query. Within the personalized system design, we assume that an actual query is not always independent of the individual search history. Former queries of a user may help to discriminate ambiguous query terms. Thus, we incorporated the individual user history $\mathcal{T}_{10,u}$ (see Section 5.2.1) into the weighting function of the query vector

$$w_q(q^*_{exp}, t) = \begin{cases} 1 & \text{if } t \in q^*_{exp} \\ 0.2 & \text{if } t \in T_{Q,u} \land t \notin q^*_{exp} \\ 0 & \text{otherwise} \end{cases} \tag{5.6}$$



(a) 1st Update (b) 2nd Update

Figure 5.2: Example of a Term-based Context Update

The collaborative search context with the largest similarity to the timestamped context is updated, if it exceeds the update threshold ε. In Section 5.4, we investigate the grouping behavior for different thresholds. For example, Figure 5.2 shows two updates of a collaborative search context with $\varepsilon = 0.5$. Both updates enhance the set of terms, links, and users. The first update is initiated by the similar context term `englisch`. The second update is performed with the context term `wörterbuch`. The samples show that a context can only be updated with a new user if identical terms are requested. The method is independent of the link is associated to the query. The term-based similarity function does not consider cross-references between terms if they are associated with the same link. In order to incorporate cross-references between links and terms, we developed a community-based similarity function.

Community-based Similarity The combination of similarity measures considers query terms and the link of a timestamped context. A linear combination of a query, link, and user similarity can fail due to three problems:

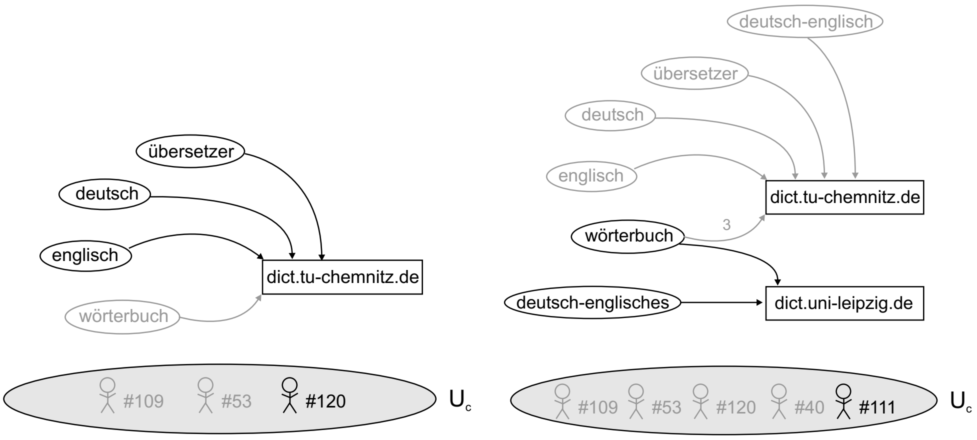
Problem 1 A high similarity of query terms between two contexts without a similarity between links can be a sign for a query term ambiguity or shift of interests.

Problem 2 A high similarity of links between two contexts without a query term similarity can be a sign for a *weak link*[1] or synonymous query terms.

Problem 3 A high diversity in a set of users with respect to common queries or assessed links can be a sign for specific user communities or distinct query contexts.

All three problems result from a high diversity of possible contexts. Each context update has to consider a user's short-term and long-term search interests. On one hand, for short-term interests, specific contexts have a high similarity based on queries and links, but more so on user similarity. On the other hand, long-term information needs are characterized by a

[1] Weak links refer to Web pages which are relevant for several distinct topics (e.g. online newspapers).



(a) 1st Update

(b) 4th Update

Figure 5.3: Example of a Community-based Context Update

high query similarity and user similarity, but not by a link similarity. In general, a context gains from long-term, as well as short-term queries and should be updated independent of the user group. Hence, we define a community-based similarity measure that combines the term-based similarity measure with a link-based similarity measure as follows:

$$sim_C = \alpha \cdot sim_T + \beta \cdot sim_L \tag{5.7}$$

The community-based similarity parameters α and β are difficult to determine in advance. They can be adjusted over the time when the system is in use.

The link-based similarity is defined in analogy to the term-based similarity. All context links are represented by a link vector space LVS$_c = \mathbb{R}^{|\mathcal{L}_c|}$. A context C is assigned to a link-based context vector $\overrightarrow{cl} = (w(\mathcal{L}_c, l_1), \ldots, w(\mathcal{L}_c, l_{|\mathcal{L}_c|})) \in$ LVS$_c$. In addition, a query vector for a link $\overrightarrow{l^*} = (w_l(l^*, l_1), \ldots, w_l(l^*, l_{|L_c|})) \in$ LVS$_c$ is assigned to the timestamped context. We compute the cosine similarity of a link-based context vector and a link vector:

$$sim_L(cl, l^*) == \frac{\overrightarrow{cl} \bullet \overrightarrow{l^*}}{\| \overrightarrow{cl} \| \times \| \overrightarrow{l^*} \|} \tag{5.8}$$

A weighting function is given by

$$w(\mathcal{L}_c, l) = \frac{lf(l)}{max(lf)} \tag{5.9}$$

where the link frequency $lf(l)$ is the number of link occurrences in associations that are assigned to the context C. The weighting function of the link vector is similar to the weighting function for query terms (see Equation 5.6). Instead of a term history, we consider the individual link history of a user. We applied the community-based similarity to the sample context of Figure 5.2. We observed in total 22 updates in a set of 14,000 timestamped contexts. In Figure 5.3, we depict the context after the first and the fourth update. The combination of a term similarity with a link similarity results in a set of terms that are all related to the topic ‘translation’. Community-based similarity allows to group users with similar terms but who assessed totally different links.

Both types of context similarity measures are based on intra-type similarities. None of these approaches incorporate cross-references between different types of items. Queries, as well as links are the essential features of a context. In order to exploit cross-references between terms and links, we need an expansion process after a context is initialized or updated. Former associations which failed the similarity principles on their processing time can now be added to a context.

5.2.4 Context Expansion

After each context initialization or update, a collaborative search context is expanded with similar users of the candidate set $\mathcal{N}$. It includes all old (former) timestamped contexts which could not be grouped. The reason for two contexts failing to be grouped might be that two associations have identical queries but different links, or an identical link but different queries. The initialization of a new context only based on identical queries or links is excluded because the similarity of single items is too weak due to the heterogeneity of possible contexts. We expand only the set of users and associations based on a term and link expansion.

(1) Term-based Expansion: Through the synchronization of all timestamped contexts, it is possible that identical query terms occur in different queries (see Section 4.1). The synchronization of all local feedback information splits up each query in its terms. An expansion based on a single term increases, as well as will decreases the specificity of a context. In the first case, the context specificity increases if an expansion is based on a specific term, for example the term `ebay`. In the second case, if a context is expanded based on an unspecific term like `definition` semantically unrelated links might be associated with this topic. The second scenario leads to an uncontrolled expansion which must be prevented in order to focus on the narrow context structure. Thus, we applied two limitations to the term-based expansion process:

- The set of context terms $\mathcal{T}_c$ which are used for an expansion is limited to terms requested by more than one user.

$$\mathcal{T}_c' = \{t \mid \exists u, v \in \mathcal{U}_c, l_1, l_2 \in \mathcal{L}_c : (t, l_1, u) \in \mathcal{A}_c \wedge (t, l_2, v) \in \mathcal{A}_c \wedge u \neq v\}$$

- A context gets a new associated member if his association consists of a context term $t \in \mathcal{T}_c'$ and a context link $l \in \mathcal{L}_c$.

Both limitations prevent an uncontrolled growth. A term-based expansion selects new users who are associated to the context. Experiments without the second limitation showed that the expansion process failed, because single contexts evolved to a heterogeneous collection of several thousand links. For example, a community degenerated because is has been expanded with links that are associated with the term `definition`. Both limitations ensure that associations of the candidate set are only added if they are based on known context terms and links. The expanded set of new associated users is defined by:

$$\mathcal{U}_{exp} = \{u \in \mathcal{U} \mid \forall t \in \mathcal{T}_c' \; \exists l \in \mathcal{L}_c : u \notin \mathcal{U}_c \wedge (t, l, u) \in \mathcal{N}\} \tag{5.10}$$

For all users of the expanded set $\mathcal{U}_{exp}$ and all known context users $\mathcal{U}_c$, associations of these users are located in the candidate set that are based on context terms and links. These associations are added to a collaborative search context $C = (\mathcal{T}_c, \mathcal{L}_c, \mathcal{U}_c \cup \mathcal{U}_{exp}, \mathcal{A}_c \cup \mathcal{A}_{exp})$ with

$$\mathcal{A}_{exp} = \{(t, l, u) \in \mathcal{N} \mid t \in \mathcal{T}_c' \wedge l \in \mathcal{L}_c \wedge (u \in \mathcal{U}_{exp} \vee u \in \mathcal{U}_c)\} \tag{5.11}$$

(2) Link-based Expansion:

In analogy to a term-based expansion, the set of context users is based on context links. The expansion process does not enhance the set of terms, because specific links might be relevant for different topics. If all terms associated with a link are used for an expansion, the context will get to diverse. For example, a Web site of a large newspaper can be associated to several different terms which address different user interests. Thus, two limitations have been applied to the link-based expansion process in order to regulate the set of new users:

- The set of context links $\mathcal{L}_c'$ which is used for an expansion is restricted to links assessed by more than one user.

  $$\mathcal{L}_c' = \{l \mid \exists u, v \in \mathcal{U}_c, t_1, t_2 \in \mathcal{T}_c : (t_1, l, u) \in \mathcal{A}_c \wedge (t_2, l, v) \in \mathcal{A}_c \wedge u \neq v\}$$

- A context gets a new associated member if his association consists of a context term $t \in \mathcal{T}_c$ and a context link $\mathcal{L}_c'$.

For the link-based expansion, the set of associated users is expanded with a set of new users who share context terms and links

$$\mathcal{U}_{exp} = \{u \in \mathcal{U} \mid \forall l \in \mathcal{L}_c' \; \exists t \in \mathcal{T}_c : u \notin \mathcal{U}_c \wedge (t, l, u) \in \mathcal{N}\} \tag{5.12}$$

The expanded set of users and the set of associated users allow to remove associations from the candidate set $\mathcal{N}$ in order to add these associations to the context $C = (T_c, L_c, U_c \cup U_{exp}, A_c \cup A_{exp})$ with

$$\mathcal{A}_{exp} = \{(t, l, u) \in \mathcal{N} \mid t \in \mathcal{T}_c \wedge l \in \mathcal{L}_c' \wedge (u \in \mathcal{U}_{exp} \vee u \in \mathcal{U}_c)\} \tag{5.13}$$

In general, we apply first a term-based expansion and then a link-based expansion. It depends on the context if one or both context expansions result in an update. Neither the update process nor the expansion process limit the number of associations of a user to contexts. All user interests contribute to different contexts if similar interests are discovered. Thus, both processes, update and expansion, define the evolution of a context which grows with its terms, links, and users. The final evolution step of a context is a community.

5.3 Tripartite Community Approach

The general process of suggesting communities is an automatic processes. Communities are the final evolution step of local search interests. They grow from single information needs of an individual user to collaborative search contexts which are classified as communities if they have a stable growth. Finally, the announcement of a community is based on three phases:

1. *Proposal*: For each community candidate an advertisement is generated which is a summary of the context, and all associated users are selected who get the community advertisement.
2. *Verification*: Each user selects from community advertisements by explicitly selecting his memberships.
3. *Confirmation*: All confirmations of memberships are collected in order to perform the announcement of a community.

The tripartite community approach performs no automatic grouping of users. Each user can explicitly join a community or not. The result of the final confirmation are *Virtual Knowledge Communities* with explicit user memberships.

5.3.1 Community Candidates

All items of a context are assessed in order to measure its evolution. The *evolution* of a context reflects the growth stability based on a *redundancy* measure for terms, links, and users. It is defined by the number of replications of query-link associations. In addition, the evolution of a contexts reflects its *lifetime* which is defined by the age of the context since it has been initialized at a particular time unit. During the lifetime of a context, the context can evolve to a community candidate.

Community Candidate The evolution of a community candidate shows a high redundancy over a particular time. For the lifetime of a context, the redundancy ratio reflects the temporal history of repeated queries, links, and users. The age of a context is not essential for the candidate criteria. ’Young’ contexts, as well as ’old’ contexts are equally good community candidates.

The growth stability of a context c is measured based on replicated terms and links of users who contributed to it. The redundancy ratio expresses the significance of a user’s query-document association which reflects previous assessments of already associated context users. It is measured at each update step i by an accumulation over all previous update steps (n). For each context update with a timestamped context (q^*_{exp}, l^*, u), we calculate three item-specific redundancy ratios of the context c:

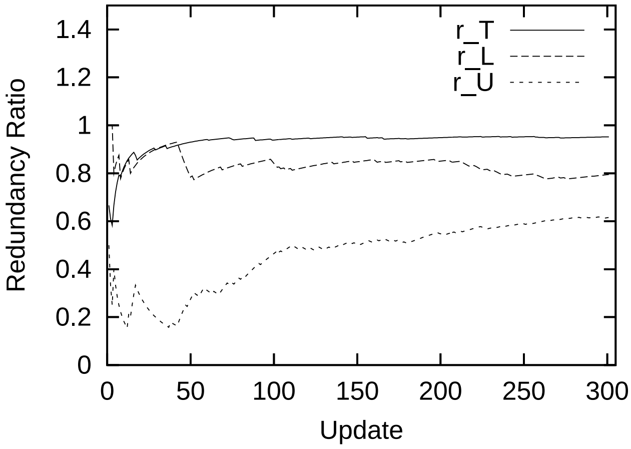


Figure 5.4: Redundancy Ratios of a Sample Community

- Term-based Redundancy Ratio:

$$r_T(c) = \frac{1}{n} \sum_i \frac{kt_{c,i}}{|q^*_{exp}|} \tag{5.14}$$

- Link-based Redundancy Ratio:

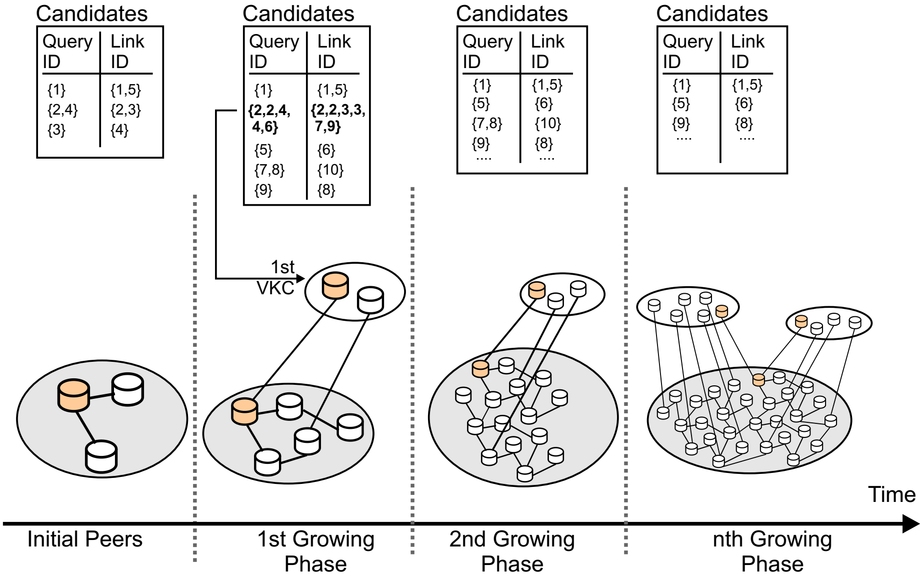
$$r_L(c) = \frac{1}{n} \sum_i kl_{c,i} \tag{5.15}$$

- User-based Redundancy Ratio:

$$r_U(c) = \frac{1}{n} \sum_i ku_{c,i} \tag{5.16}$$

where $kt_{c,i}$ is the number of known context terms of the actual query q^*_{exp}, $kl_{c,i}$ is the number of known context links, and $ku_{c,i}$ is the number of known associated users to the context. All redundancy ratios have a value between 0 and 1. In Figure 5.4, we show the redundancy ratios at each update step of a sample context. After the initialization of the community, we observed a decrease of all three redundancy ratios. In the first update step, we computed a r_T of 0.67, a r_L of 1.0, and a r_U of 0.5. Over the time, r_T increased, and the number of repeated links slowly decreased. In addition, the number of users contributing repeatedly to the community increased over the time.

All redundancy ratios should not fall below certain thresholds. We defined three thresholds γ_T, γ_L, γ_U which control the growth stability of all redundancy ratios. The sample context shows that it is difficult to find a minimal threshold γ_U for the user-based redundancy ratio. It underlies large changes during the first updates, because a context naturally will grow based on new users who commit existing associations. Community candidates are only selected if a minimal threshold for a term-based and link-based redundancy ratio has been exceeded. For these candidates, an advertisement process (see 7.2.2) is initiated in the peer-to-peer network to announce a Virtual Knowledge Community.



Candidates

Query ID	Link ID
{1}	{1,5}
{2,4}	{2,3}
{3}	{4}

Candidates

Query ID	Link ID
{1}	{1,5}
{2,2,4, 4,6}	**{2,2,3,3, 7,9}**
{5}	{6}
{7,8}	{10}
{9}	{8}

Candidates

Query ID	Link ID
{1}	{1,5}
{5}	{6}
{7,8}	{10}
{9}	{8}
....	....

Candidates

Query ID	Link ID
{1}	{1,5}
{5}	{6}
{9}	{8}
....	....

Figure 5.5: Growing of Virtual Knowledge Communities

5.3.2 Virtual Knowledge Communities

A community candidate is denoted as *Virtual Knowledge Community (VKC)* if users confirm their memberships. The core of a Virtual Knowledge Community is a collaborative search context. We define a Virtual Knowledge Community as follows:

Definition 5.4 *A Virtual Knowledge Community of a context C is a 6-tuple* $V(C, E, r_T, r_L, r_U, c)$ *with*

- *C: collaborative search context (Definition 5.3)*
- $\mathcal{E}$*: set of confirmed members*
- r_T*: term-based redundancy ratio*
- r_L*: link-based redundancy ratio*
- r_U*: user-based redundancy ratio*
- *c: confirmation rate*

The confirmation rate represents how many users confirmed their membership. In general, Figure 5.5 depicts the growing of the community structure. The *Candidates* table summarizes query ID's and link ID's for each phase. User ID's are not visualized in the example. As time continues, the *Candidates* table grows with more links and queries, but also the number of users joining the peer-to-peer network increases. As soon as the redundancy ratios of a context exceed certain thresholds, a Virtual Knowledge Community is announced. In the given scenario in Figure 5.5, an advertisement for vkc_1 is computed in the 'First Growing Phase', after the collaborative search context has been initialized. The user group is based on all confirmed members contributing with query-link associations to the *Candidates* table. Once a VKC is created, the particular context is removed from the table *Candidates*.

Summarizing, Virtual Knowledge Communities groups similar information needs of the entire user group. If a Virtual Knowledge Community is relevant for a current information need, results from this group promise a high effectiveness due to their validation by several users. Evaluation results of the community quality are presented in the next section.

5.4 Evaluating Communities

This section contains results from community experiments on the Weblog corpus, and it provides empirical evidence on how different similarity functions affect the community process. Click-through data is used to simulate the community algorithm. The experiment is designed to measure the quality of the grouping in relation to the set of terms and documents.

5.4.1 Evaluation Setting

The community discovery process is simulated with all 220,506 search sessions of the Weblog corpus. Each search session and its UNIX timestamp is used as a timestamped context. No time synchronization is necessary, because all search sessions are temporally ordered due to the collection by one proxy. The evaluation with the Weblog corpus is limited, because not explicit relevance feedback is available. The click-through data is used as implicit feedback in order to derive relevance assessments.

For all experiments with the Web log data, no standard clusters were available. The results cannot be compared with other approaches running on different data sets. We assume that the Web log data collected by one proxy would define specific constraints for the verification of the community concept. On one hand, the fact that all users belong to the same local area might prevent the grouping of highly specific interests within a community. On the other hand, special regional interests can confirm the verification of the community concept which identifies common professional or personal interests. The high diversity of search interests shows that a regional proxy collection does not limit the evaluation setting (see Section 4.1.2). We were able to verify that it is possible to find common interests among users who live in the same region.

The community discovery process has been simulated with two different settings. They differ in their update and expansion process:

K-Sim This setting of the community algorithm uses the term-based similarity measure. The expansion step of a community applies only the term-based method.

VKC-Sim This setting uses the community-based similarity measure. The expansion step of a community applies the term-based method and the link-based method.

In addition, the following list of parameters must be set for the initialization and expansion of a community:

Table 5.1: Number of communities obtained by varying the similarity thresholds for both evaluation settings.

	No. of Communities	
ε	K-Sim	VKC
0.5	2290	2290
0.6	2286	2292
0.7	2287	2292
0.8	2290	2293
0.9	2291	2292
1.0	2292	2292

- δ: threshold for query expansion (see Equation 5.2.1)
- ρ, τ: thresholds for context initialization (see Section 5.2.2)
- α, β: parameters for community-based similarity (see Equation 5.7)
- ε: threshold for context update (see Section 5.2.3)
- γ_T, γ_L, γ_L: thresholds for growth stability (see 5.3.1)

In all experiments, we varied the update threshold ε from 0.5 to 1.0. Thus, we obtained different proportions and numbers of clustered terms, links, and users. All other parameters of the community settings are set to a constant value:

- $\delta = 0.5$
- $\rho = 2$, $\tau = 1$
- $\alpha = 0.5$, $\beta = 0.5$
- $\gamma_T = 0.5$, $\gamma_L = 0.3$, and $\gamma_U = 0.0$

By varying the similarity threshold, we intended to show how the community characteristics behave. In addition, we expected to observe individual community properties depending on the similarity threshold. The general characteristic of the grouping behavior is shown for the following three aspects:

Size Table 5.1 shows the statistics of the context grouping of 220,506 ratings. Neither the application of different settings nor the variation of the update threshold influences the number of communities significantly. Differences among the communities are observed for the proportions of clustered terms and links. The proportion of clustered users remains constant. We conclude that in general the common interest is detected with each setting, and remains constant independent of the update threshold.

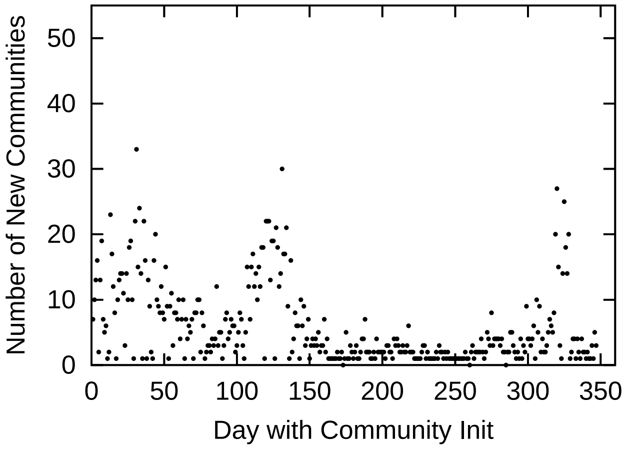
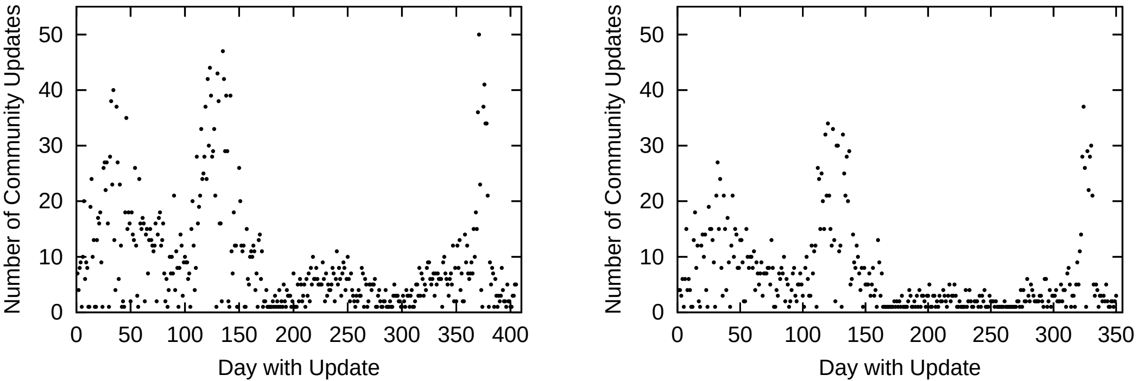


Figure 5.6: Community Instantiation over a 536 Day Period



(a) K-Sim

(b) VKC-Sim

Figure 5.7: Community Updates

Instantiation For both settings, the instantiation of new communities shows that on some days many communities are created. In general, we observed that K-Sim and VKC-Sim have the same creation behavior (see Figure 5.6), due to the fact that they have the same thresholds ρ and τ for the community initialization.

Update In Figure 5.7, the update behavior of the K-Sim and VKC-Sim setting is depicted. We observed fewer updates for VKC-Sim than for K-Sim. The general update behavior of all communities enables no detailed analysis of an individual community. Thus, we selected a sample community, and we analyzed its growth in detail (see Figure 5.4). When a community was instantiated many new users are associated with it. Only over a longer time period a user-specific redundancy ratios can be measured. In addition, the number of days between two updates tended to grow the older the community got. Once the community was initialized, we noticed many updates within short periods of days. Figure 5.8 depicts the increase of days between two updates. In correlation to the dwindling interest, we see that the redundancy ratios increase constantly. Over the time, the community still gets new members, and all members agreed in the same community terms. The characteristic of r_L shows that the community is scarcely updated with new links.

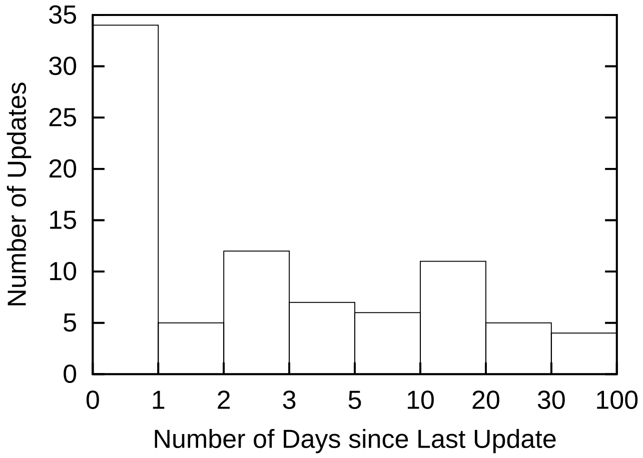


Figure 5.8: Distribution of Days between two Updates

5.4.2 Evaluation Metrics

The overall quality of a community depends on the quality of grouped terms and links. The Weblog corpus provides only implicit feedback. Thus, we conjecture that the community quality is lower than it would be with explicit feedback. On average, each user viewed 3.3 links for a query. All query-document associations are handled equally in our evaluation. In a real search scenario, not all viewed links might be relevant to the user if he provides explicit feedback. The evaluation of such a scenario needs a user study which cannot be contributed within the scope of this dissertation. For this purpose, we measure the quality of the community approach based on implicit feedback. The quality of a community is measured for the associated terms, links, and users. The user-specific quality of the grouping cannot be evaluated with anonymous log data. Without precise information about the identity of a user, we cannot measure whether authorities for this community have been grouped.

Link-based Community Quality To measure the link-based community quality, we compare the community links to a ground truth similarity ordering of links extracted from the Open Directory Project[2] (ODP). The ODP is a human-edited directory of the Web maintaining a hierarchical collection of Web pages. We incorporated the similarity information that is implicitly encoded in the ODP category tree as follows:

- The ODP tree is collapsed into a fixed depth. The leaves contain the classes of documents (URLs).
- A familial ordering is defined by the documents that fall into the same class, a sibling class, a cousin class, etc. (see Figure 5.9).
- We assume that the true similarity of pages decreases monotonically with the familial ordering.

[2] http://www.dmoz.org

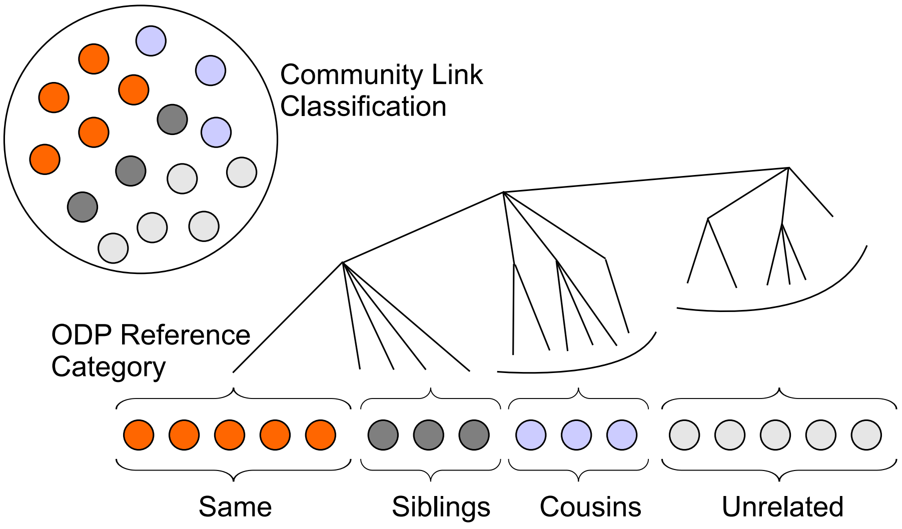


Figure 5.9: Community Classification based ODP Category Tree

We extracted nearly 3.8 million links and 505.514 categories of the ODP. The pre-processing of all communities includes the following steps:

1. We marked each community link with a ODP category if it could be found in our data set. For the comparison of community links and ODP links, we applied an inexact match, and trimmed the links in order to normalize them.

2. We selected all communities with at least 3 ODP links and a proportion of at least 25% ODP links.

After the pre-processing of the test corpus, we analyzed the ODP links of a community. We classified all community links according to a *reference category*. This category contains the largest number of ODP links of the community. If more than one category fulfills this criteria, we randomly selected one of these. The reference category is used to compute four familial distances as visualized in Figure 5.9:

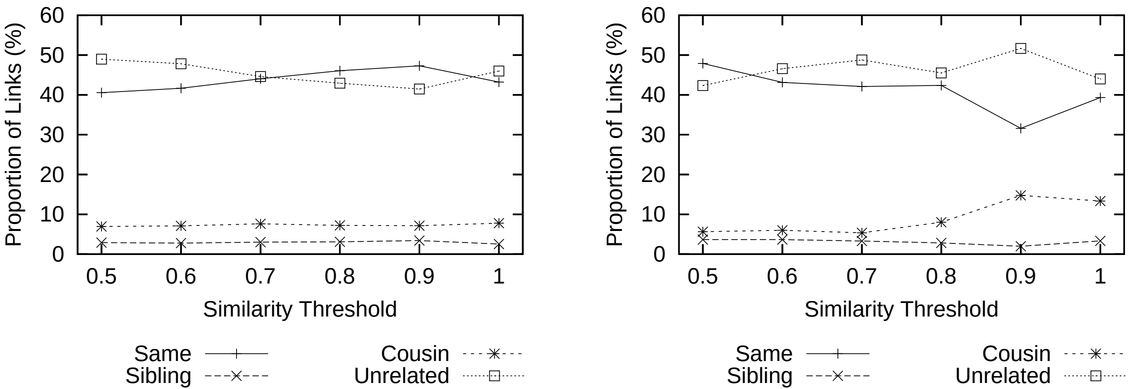
Same All links are counted that are in the ODP reference category.

Sibling All links are counted that are in a sibling class.

Cousin All links are counted that are in the classes which are first cousins.

Unrelated All links are counted that does not fulfill the other familial distance criteria.

The result of the average proportion of links with an assigned familial distance is depicted in Figure 5.10. By changing the threshold, the average proportion of sibling and cousin links is nearly constant for all K-Sim communities. We noticed twice as many cousin links than sibling links. For the VKC-Sim communities, the proportion of sibling and cousin links behaved similar. However, we observed an increase in the proportion of cousin links with a



(a) K-Sim (b) VKC-Sim

Figure 5.10: Familial Distance of Community Links

threshold greater than 0.8. The majority of all ODP links belongs either to the same category or they are unrelated. For all unrelated links, the root is the lowest common ancestor of the document classes. We conjecture that the high proportion of unrelated links results from implicit feedback. Each user might has viewed several links until he found a relevant document. K-Sim communities show an inverted change of the proportion of both types of links. The VKC-Sim behavior is nearly similar: if the proportion of same links decreases, the proportion of unrelated links increases. Only for a threshold greater than 0.8, the decrease of links in the same category leads to a significant increase of cousin links, and a low increase of unrelated links. In general, we classify all links as similar if they have a same, sibling, or cousin category. All similar links are used to measure the quality of grouping links. A weighting of different distance types is not performed due to the low occurrence of sibling and cousin links. Thus, we do not consider a semantic similarity measure in the ODP category tree as discussed by Resnik (1999). We borrow two IR metrics to measure the quality of clustering links:

Link Precision is the ratio of the number of similar links to the total number of ODP links associated with a community.

Link Recall is the ratio of the number of similar links to the total number of all similar links for these ODP links in the current community or in others.

It is difficult to use the recall metric without standard communities. We calculate normalized link recall as follows:

- For both community approaches, we collect the number of correctly grouped links in all communities for a specific update threshold. This number is calculated by multiplying the total number of grouped links with the link precision of the update threshold.
- The total number of correctly grouped links of a specific threshold is divided by the maximum number of correctly grouped links among all thresholds. In our case, this number is 1185 and is obtained by K-Sim when the update threshold is 0.6. Thus, we calculate a normalized link recall in the range $[0, 1]$.

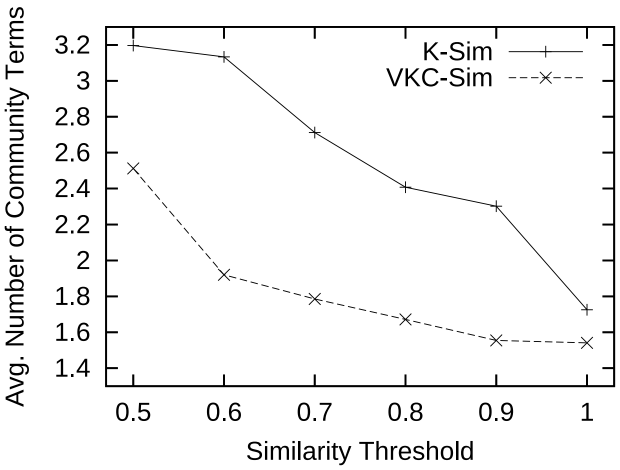


Figure 5.11: Average Number of Community Terms

Term-based Community Quality The cluster purity measure is borrowed as a metric for the quality of community terms. We define the *community purity* of a community C_j as follows:

$$\textit{purity}(C_j) = \frac{1}{|\mathcal{T}_{c_j}|} * rt_j \tag{5.17}$$

where $\mathcal{T}_c$ is the set of terms associated with the community j, and rt_j is the number of relevant terms assigned to community j. The number of relevant terms for each community is measured manually by two assessors. The relevance assessments are based on all community terms and ODP link categories. By combining these two factors, the assessors attempted to guess the search intentions of the associated users. Both assessors worked independent of each other. One assessor worked on a set of 187 K-Sim Communities, and the other one worked on a set of 121 VKC-Sim communities.

The overall purity of community setting is calculated by a weighted sum of individual community purities

$$\textit{purity} = \sum_{j=1}^{k} \frac{|\mathcal{T}_{c_j}|}{|\mathcal{T}|} * \textit{purity}(C_j) \tag{5.18}$$

where k is the number of communities, and $\mathcal{T}$ is the set of all grouped terms.

5.4.3 Term-based Community Quality

In general, we observed that the K-Sim setting groups more terms and links. It is not so strict during the grouping process. The difference between the average number of terms per community in both settings is visualized in Figure 5.11. We observed that an increase of the update threshold leads to a decrease of the terms associated with a community. An exploratory data analysis shows that related terms are grouped in communities. The analysis

Table 5.2: Top 10 VKC-Sim communities ordered by their number of distinct terms. The examples are selected for a threshold of 0.7.

id	no. terms	no. links	no. users	top 5 terms
1433	16	8	2	gradient, flow, vector, dynamic, field
146	15	2	22	fh, bonn, rhein, sieg, augustin
14	14	27	102	wörterbuch, englisch, deutsch, online, english
260	13	35	13	process, statistical, control, sigma, 6
467	13	18	11	bayern, münchen, fc, ag, gründung
199	12	56	24	herr, ringe, arwen, kostüm, gewand
95	11	29	3	robotics, medical, future, robodoc, remagen
114	11	16	22	acrobat, reader, download, adobe, 7.0
58	10	1	65	leo, dictionary, dict, english, org
236	10	6	7	rheinland, pfalz, landessportbund, lsb, dsb

of the top 10 communities, which are ordered by their total number of distinct terms, verifies this observation. The largest VKC-Sim communities are listed in Table 5.2.

To investigate the term-based community quality, we applied the purity measure as an evaluation metric (see Section 5.4.2). We report the purity of our selected communities in Table 5.3, where both community approaches are shown with a update threshold ε varying from 0.5 to 1.0. Furthermore, we differentiate between two sets of terms which are evaluated:

- *Baseline*: We considered all terms which are associated with a community.
- *Subset*: We selected automatically all terms which have been used by at least two different members.

We observed that, in general, the combined similarity approach (VKC-Sim) results in a higher purity than communities which are discovered only based on the term similarity. We observed that the highest purity value that can be reached by VKC-Sim on selected terms is about 0.89. The purity of VKC-Sim is always higher than K-Sim. This observation is correlated with an averagely smaller number of associated terms with VKC-Sim communities. The VKC-Sim approach can take advantage of the evidence of a link-based similarity and expansion. Furthermore, we noticed that the purity is improved for a subset of terms with both settings. In general, this selected set of terms improves the purity significantly for the K-Sim approach. K-Sim communities have a nearly constant term precision when the update threshold is lower than 1.0. The maximum K-Sim term purity is 0.64 for the baseline and 0.73 for the subset when the update threshold is 1.0. The subset of terms improves the K-Sim purity significantly. For a threshold smaller than 1.0, precision is always higher than it is on the full set of terms.

5.4.4 Link-based Community Quality

The quality of community links is measured for 187 K-Sim communities and 121 VKC-Sim communities. We expect that not all community links have been relevant for their members,

Table 5.3: Results of Community Purity. K-Sim and VKC-Sim report results of the baseline set of terms, and K-Sim Subset and VKC-Sim Subset report results of a subset of terms.

ε	K-Sim	K-Sim Subset	VKC-Sim	VKC-Sim Subset
0.5	0.38	0.7	0.64	0.87
0.6	0.37	0.69	0.65	0.82
0.7	0.38	0.63	0.76	0.81
0.8	0.42	0.67	0.77	0.81
0.9	0.44	0.67	0.88	0.89
1.0	0.64	0.73	0.88	0.89

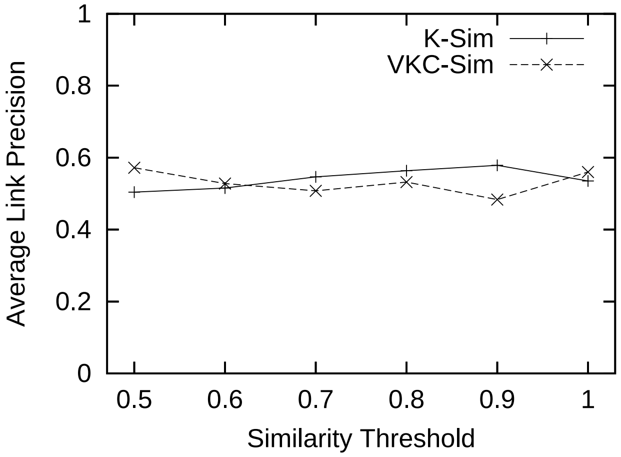


Figure 5.12: Average Link Precision for the Automatic Baseline

and only a subset of links would not have gotten explicit feedback. In order to verify this assumption, we calculate link precision and link recall on three sets of community links. The first set of links includes all links associated with a community. The link similarity is measured with the ODP category as discussed in Section 5.4.2. This method is used as a baseline for relevance assessments. In a second setting, we used all community links and collected human relevance assessments. We investigated if the quality changes due to an automatic or a manual relevance assessment of links. Finally, we applied a selection strategy which narrows the set of community links. We used the automated relevance assessments to measure the quality of the subset of links. For all three settings, we report the average link precision and the average link recall.

Automatic Baseline This settings computes the community quality for all links grouped to a community with the K-Sim and the VKC-Sim approach. The similarity of links is automatically extracted from the ODP category. With the automated process, we investigated the following results:

- Figure 5.12 depicts link precision for both community approaches on the total set of the community links. By changing the threshold, we noticed an increase in link

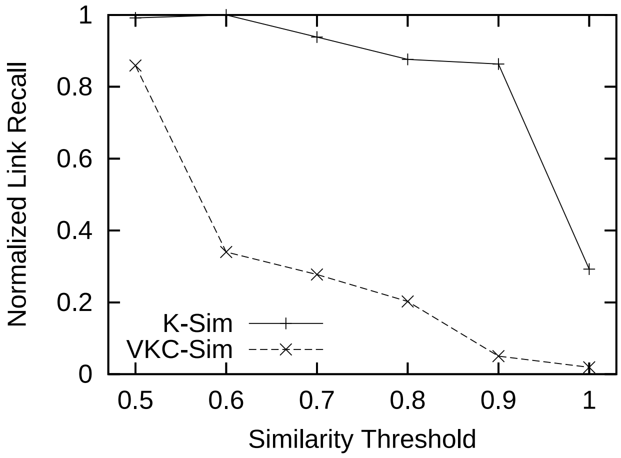


Figure 5.13: Normalized Link Recall for the Automatic Baseline

precision from 50% (threshold 0.5) to 58% (threshold 0.9) for K-Sim. The VKC-Sim approach has its peak at 57% with a threshold of 0.5. For all other thresholds, the link precision decreases. The lowest link precision is observed at a threshold of 0.9 with 48%. In general, the change of precision is not very significant for both approaches. On one hand, we conjecture that our basic community approach captures the salient community topic very well. This change does not influence the quality of the links.

- Figure 5.13 shows the normalized recall for both community approaches. We observed that for all similarity thresholds, K-Sim results in better normalized recall ratios than using VKC-Sim. This shows that the integration of a link-based similarity and expansion has no advantage in comparison to a keyword-based similarity. When the update threshold increases, normalized recall ratios drop quickly for VKC-Sim. In addition, normalized recall ratios of K-Sim decrease slowly when the update threshold is below 0.9. The poor results of the link-based community approach leads to the conjecture that many nonrelevant documents are considered in the click-through data and a special selection of community links can positively influence the link precision.

Manual Baseline For the second setting, two assessors manually judged the relevance of links based on their ODP category. The result of one assessor's judgement is presented in Table 5.4. We provided the assessors only the category description, because it was easier for them to guess the actual intentions of users by taking into account terms and categories instead of links. This example shows, that references between link categories are considered by an assessor, but has not been considered by the automatic category assignment (see Section 5.4.2). The previous results of the automated baseline method did not take into account that categories might be related. The implicit encoded similarity of ODP categories includes relatedness of topics through cross-references in the hierarchy. We did not automatically extract these references. Instead, both assessors manually marked related categories. Table 5.4 shows that one assessor also judged the third category as relevant, which has been classified as unrelated according to the familial distances. Both first topics are classified in a country specific category (Top/World/Deutsch). The third category is also related to the term

Table 5.4: Relevance judgment of VKC-Sim community #1569 based on ODP categories. The most frequently used term of this community is `biodiversität`.

ODP Category	**Rel.**	**#links**
Top/World/Deutsch/Wissenschaft/ Umweltwissenschaften/Biodiversität	+	6
Top/World/Deutsch/Wissenschaft/ Umweltwissenschaften/Biodiversität/Fakultäten_und_Institute	+	1
Top/Science/Technology/Energy/Hydrogen/Storage	+	1

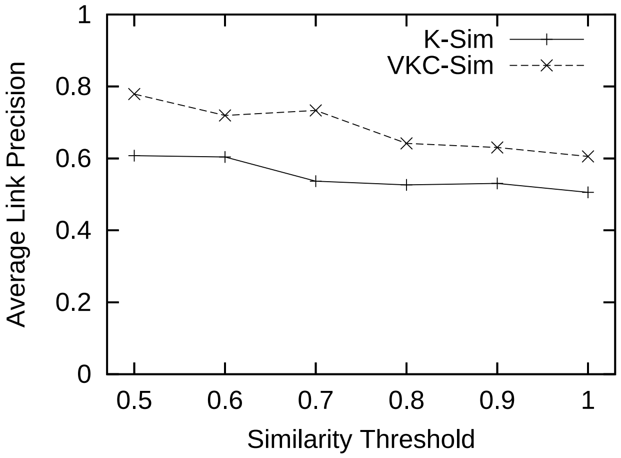


Figure 5.14: Average Link Precision for the Manual Baseline

`biodiversität`, because it is a related English category. The results show that the human assessors considered relatedness across language boundaries. We collected the relevance judgments for each community setting with a threshold of 0.5, and then we applied these relevance judgments to all communities computed with different thresholds. The manual assessed link categories show the following results:

- Figure 5.14 shows the results of average link precision for the manual baseline. We first observed that the manual assessments showed a higher link precision for VKC-Sim than for K-Sim. This observation corresponds to the results of the term-based quality. In general, link precision based on a manual assessment performs nearly constantly when changing threshold. We observed that VKC-Sim is always higher than K-Sim with a margin of about 20%.
- Although the precision of VKC-Sim is always higher than K-Sim, the normalized recall shows better results for K-Sim (see Figure 5.15). We normalized the number of correctly grouped links by dividing it by 411, the maximum number of correctly grouped links which is obtained for K-Sim when the update threshold is 0.5. For a update threshold greater than 0.5, the normalized recall drops quickly for both community approaches. The decrease of normalized recall slows down for a threshold greater than 0.7.

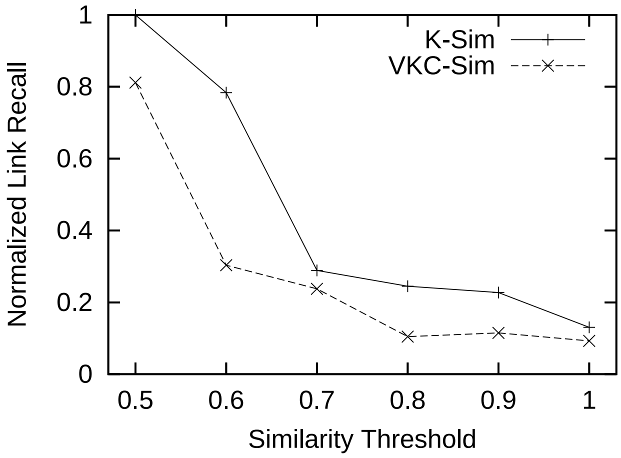


Figure 5.15: Normalized Link Recall for the Manual Baseline

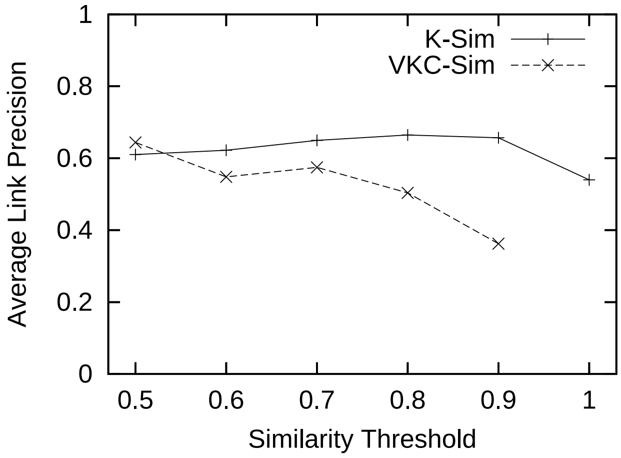


Figure 5.16: Average Link Precision for a Subset of Links

Link Subset This setting evaluates the link-based community quality for a subset of links. We assume that a subset of links might improve the link-based quality, because it narrows the set of links to a link subset that have been commonly viewed by several users. All links are selected which have been viewed by at least two different users. The implicit feedback is now derived from a group of users and not from a single user. Indeed, our data shows that the link quality increases for a subset of community links:

- Figure 5.16 shows the results of this setting. We observed a maximum link precision of 66.44% with a threshold of 0.8 for K-Sim. For all thresholds, K-Sim has a better link precision on a subset of community links. In contrast, VKC-Sim results still do not perform as good as the K-Sim results. For a threshold of 0.5, we noticed a link precision of 64.39%. By changing the threshold, link precision decreased and the values were a good deal worse than the K-Sim values. These results are correlated to the rapid decrease of ODP links for the VKC-Sim communities. For a threshold of 1.0, no community fulfils the ODP selection criteria. Only for a threshold of 0.5, VKC-Sim shows better precision and normalized recall ratios.

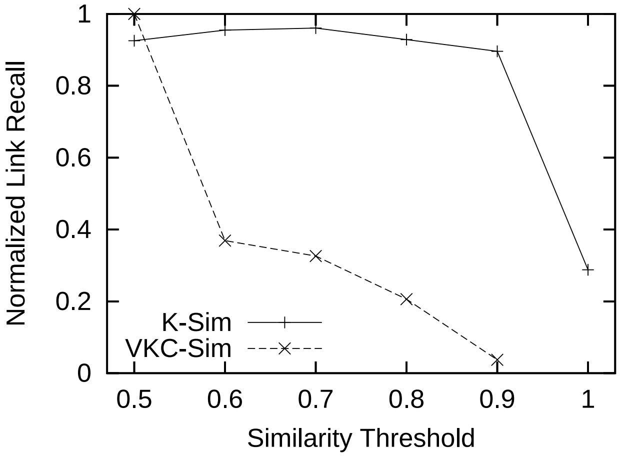
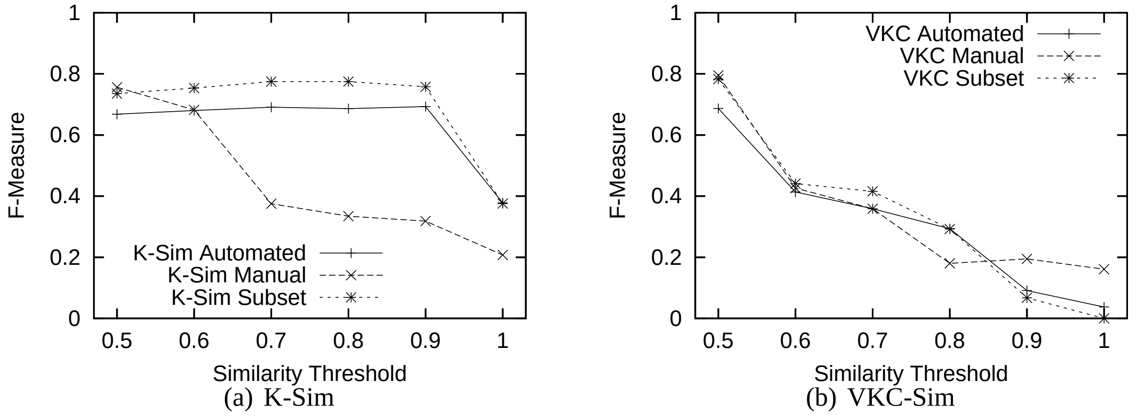


Figure 5.17: Normalized Link Recall for a Subset of Links

- We depict normalized link recall in Figure 5.17. The maximum number of correctly clustered links is 323 and is obtained by VKC-Sim when the update threshold is 0.5. This number is used for the normalization. The automatic selection of community links with a general agreement of at least two members yields no significant improvement for both community approaches. However, precision, as well as normalized recall ratio can be improved for VKC-Sim when the threshold is 0.5. When changing the threshold, normalized recall ratios still drop quickly for VKC-Sim. Hence, the implicit feedback and the automatic link assessment still contain nonrelevant documents.

We conclude that there will be a higher quality of the community approach by an integration of explicit feedback. The link-based community quality is influenced by nonrelevant links which have probably been viewed unintentionally due to a misleading summary or relevance ranking of the Web search engine. Figure 5.18 shows the F-measure of the global link quality for all three evaluation settings. For the K-Sim communities, we noticed that the link-based quality with all implicit feedback information is always higher than the setting which takes into account human assessments. The highest F-measure values are observed for an automatic selection of community links. The number of community links is decreased in order to separate relevant and nonrelevant links. This attempt results in higher F-measure values with a margin of 7%. We conjecture that a more elaborated selection function yields better F-measure values, or that explicit feedback automatically compensates the number of nonrelevant links. The difference between all three settings are not significant for the VKC-Sim communities. The manual assessed communities show F-measure values which drop quickly when increasing the update threshold. We assume that the ODP ordering represents an appropriate ground truth which is similar to human assessments. Also, we were able to notice higher F-measure values for a selection of specific community links. We expect significant higher F-measures if the community approach could be evaluated with explicit feedback. Nonrelevant links influence the community quality significantly. Owing to a high term-based term quality, we observed that the number of ambiguous terms is much lower than expected. Thus, further evaluations are necessary in order to analyze the impact of explicit feedback for the link-based community quality. Such results would deliver insights for an optimization of all community parameters.



(a) K-Sim

(b) VKC-Sim

Figure 5.18: F-Measure Results

5.4.5 Discussion

Both evaluation settings show insights into the qualitative characteristics of communities. The difficulty of evaluating the community approach is similar to a clustering approach without standard test corpora. Our evaluation is concentrated on the combination of evidence from query contents, link views, and user identities, and its influence on the community quality. We observed that communities differ in their number of associated terms, links, and users. This difference does not lead to significant qualitative differences. In related work, Web communities as defined by Flake et al. (2004) are evaluated in an exemplified manner. These communities differ from the Virtual Knowledge Community concept because they are built by link analysis and not by usage data. In Section 5.4.3, we showed several example communities for our approach.

We evaluated the term-based and the link-based community quality separately. The automated evaluation method uses the implicit encoded similarity ordering of links categorized by the Open Directory Project. In addition, two assessors provided relevance feedback for terms associated with a community. We observed a high purity when we considered terms, links, and users in the similarity and expansion function (VKC-Sim setting). The combination shows a higher purity than only a term-based similarity. This observation can be explained with the diversity of the search behavior (see Section 4.1.2). The large information collections of Web search engines often distract the user due to an unmanageable number of results. Our community approach is robust enough to prevent that weak links are used for community expansion. The results of the link-based community quality did not show a high link quality such as observed for terms. As a main reason for this observation, we identified the lack of explicit relevance feedback. Further evaluations are necessary to infer relevance from the time spent viewing a document, or a document suffers a lot of read wear (Hill et al., 1992). If such an advanced analysis of the log file improves the link-based quality of our approach, implicit feedback can act as a substitute for explicit relevance feedback. The viability of interchanging implicit and explicit relevance feedback has been shown by White et al. (2002).

5.5 Summary

This chapter explored a community concept to group terms, links, and users. Initially, it is based on a synchronization of all local information collections. After each search session with a relevance judgement, a timestamped context is defined to update the set of former associations. It is compared to a set of candidates in order to initialize a collaborative search context. The initialization process relies on associations which refer to identical relevance judgments for a subset of query terms. All users who agree on this rating are aligned with the context initialization. The similarity between two contexts can be measured by a term-based approach or a community-based approach. Each update expands a collaborative search context in order to add new similar terms, links, and users. We assigned strict criteria to the update process, as well as the expansion process. On one hand, a threshold is selected to define the minimal similarity between context terms and/or links. On the other, the expansion process is implicitly controlled by users who are already associated with the context. Both criteria avoid heterogeneous contexts. In order to guarantee a stable context growth, redundancy ratios for terms, links, and users are defined which control all updates. Each context is a candidate for a Virtual Knowledge Community. Two evaluation scenarios were set up to provide empirical evidence as to how different similarity functions affect the community results. Both similarity functions identify similar communities which differ in their number of terms, links, and members. We developed an automated evaluation methodology to analyze the global community quality which is defined by the single qualities of grouped terms and links. For the grouping of terms, we investigated that the community-based similarity function shows a higher effectiveness than the term-based method. Instead, the analysis of the link-based community quality reports that the term-based similarity function leads to a better community quality in two out of three scenarios. Only the calculation of link precision by manually assessing links results in a higher precision for the community-based similarity functions. It is conjectured that this is an effect of the data set that lacks explicit relevance judgments.

For Congenial Web Search, the model of communities is essential for an organization of information needs. The Peer Search Memory was primarily developed to organize dynamic information collections which assist individual information needs. In analogy, Virtual Knowledge Communities group common information needs of all users. Each community is a dynamic collection of individual query-document associations, and all members confirmed explicitly their long-term interests. Both types of collections enable a differentiation between a set of known documents and set of new documents since the last usage. All new documents can be used for filtering if a user has stable information needs. These long-term interests are derived from his confirmed memberships. All known documents and their related queries are an effective source for retrieval. The incorporation of the common representation of information needs and collections in information seeking processes is described in the next chapter.

6 Integrated Information Seeking

Explicit relevance feedback provides further information about retrieval items. In addition, Virtual Knowledge Communities are maintained to organize users and their interests. This chapter presents how both aspects are combined with information seeking processes. For this task, the information retrieval process, as well as the information filtering process are integrated in a social network that represents the third pillar of Congenial Web Search. A virtual search network is designed to validate all user interactions. It models a common platform for information consumers and providers. Owing to an interleaving of local usage data, a cooperative pull-push cycle evolves assisting integrated information seeking.

6.1 Integrated Information Seeking Processes

The process of combining components into larger assemblies is called integration. The integration of information seeking processes is essential to retrieve distributed information on the Web from a single point of access. Meta-search engines (see (Howe and Dreilinger, 1997), (Joshi, 2000), (Aslam and Montague, 2001), (Meng et al., 2002)) are used to merge the functional characteristics of separate Web search engines into a comprehensive, interoperable system. For the integration of two distinct information seeking processes, information retrieval and information filtering, the difference between ’process’ and ’system’ is considered. Oard (1997) defines ’process’ as an activity conducted by humans, perhaps with the assistance of a machine. By ’system’, he refers to an automated system that is designed to support humans who are engaged in that process.

6.1.1 Classification of Integrated Information Seeking Processes

The goal of *integrated information seeking* (IIS) is to match highly variable interests with rapidly changing information. In order to achieve this goal information retrieval and information filtering processes must be combined. For a combination, the process specific characteristics are factored to identify stable and dynamic parts of the information collection. We classified two new integrated information seeking processes which incorporate the personalization and the collaboration strategy. The characteristics of each process can be compared in order to extract criteria for a classification scheme. At an abstract level there is very little difference between information retrieval and information filtering. In particular, three characteristics are extracted to differentiate between both processes:

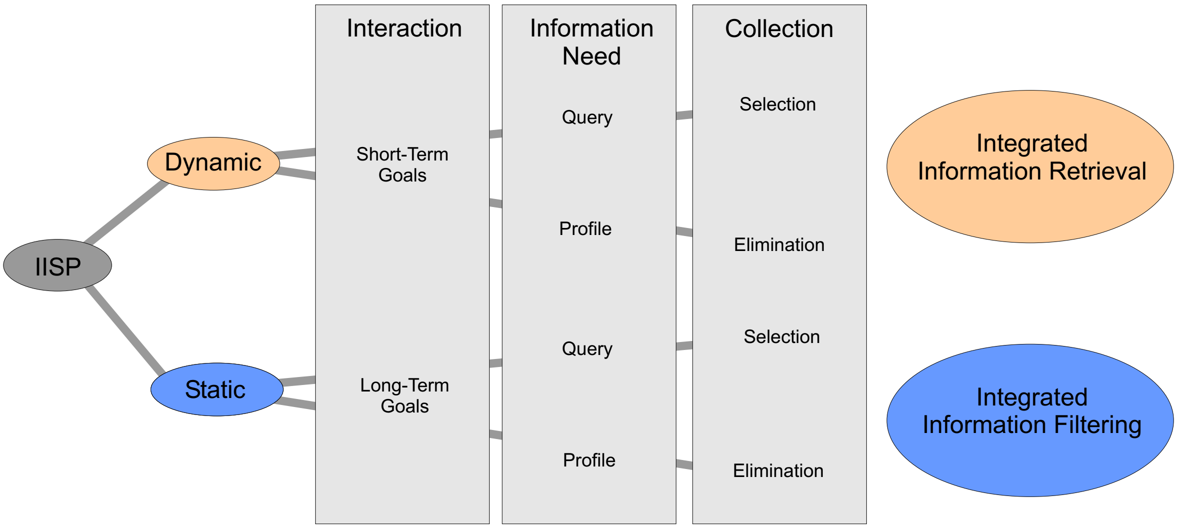


Figure 6.1: Classification of Integrated Information Seeking Processes

Interaction Information needs are distinguished as *short-term goals* or *long-term goals* (Belkin and Croft, 1992). Information retrieval is presently concerned with a single use of a system assisting this process. However, information filtering addresses repeated use of the same system with changes of the information need over a series of information seeking episodes.

Information Need Taylor (1962) classified four types of information needs: visceral, conscious, formalized, and compromised. These types denote the process of moving from the actual information need to an expression of the need that is represented in an information system. A 'problematic situation' (Belkin and Croft, 1987) arises from the situation that the user's goals cannot be attained because his resources or knowledge are somehow inadequate (Schutz and Luckmann, 1973). Such an 'anomalous state of knowledge' (ASK) (Belkin and Croft, 1987) prompts a user to submit a *query* as his compromised information need to an information retrieval system. In information filtering the compromised information need is referred to as a *profile* (Oard, 1997).

Collection The first subtask of an information seeking process is the collection of information sources. Byström and Järvelin (1995) classify the types of information collections as: fact-oriented, problem-oriented, or general-purpose. The collection of different types of dynamic information can be done actively (e.g. agent-based), passively (e.g. with RSS Feed), or as a combination of both (Oard, 1997). Information retrieval is concerned with the *selection* and organization of texts in relatively static databases; information filtering is concerned with the distribution of texts to groups or individuals by *eliminating* texts from a dynamic stream of data.

The differences in interaction, information need, and collection lead to a classification of integrated information seeking processes. In this classification, general information retrieval and information filtering processes are organized in *dynamic* and *static* approaches. This differentiation is a first classification layer (see Figure Figure 6.1) concerning the interaction of users involved in this process. A dynamic interaction of the user is assumed for short-term goals and a static interaction is performed by a user with long-term goals . The new aspect of this classification is the combination of features of the information retrieval process and the information filtering process in order to map them to static information needs and collections. This integration assigns a new combination of features which are organized by three layers in the classification scheme as depicted in Figure 6.1. The result of the feature combination comprises two integrated information seeking processes:

1. The *integrated information retrieval* (IIR) process fulfils the characteristics of a traditional retrieval process such as short-terms goals are expressed by a query which is used to retrieve information from a static collection. The information collection is either an external or an internal provider. External providers offer a large index for traditional Web search. Internal providers offer their search sessions annotated with explicit feedback. These dynamic collections are filtered by mapping the query to static user profiles. Each profile that matches with an actual query allows to investigate a user group organized in a community.

2. The *integrated information filtering* (IIF) process fulfills the characteristics of a traditional filtering process such as long-term goals are represented by a profile which is used to eliminate information from a dynamic stream. Web search engines offer no incremental updates of their search index. Instead, a dynamic stream of individual updates of all Peer Search Memories is located. The local personalization strategy of each peer enables a timestamped logging of all relevance judgements. The static community structure can be exploited in order to retrieve new information for a long-term information need.

Both processes are intensely interleaved and combine in a new way search techniques based on personalization and collaboration. Nevertheless, the integration is modeled in such a way as to allow a factorization of an information retrieval process and information filtering process based on stable information needs and collections. A factorization consists of an initiating process followed by either information retrieval or information filtering, each enhanced with individual information providers and communities. The integrative model of Congenial Web Search is defined in greater details in the following section.

6.1.2 Integrated Information Seeking Concept

A *general model of integrated information seeking* combines existing models for information retrieval and filtering with two advantages: (1) all *process-independent* components are detected which are identical for the information retrieval and information filtering task and (2) *process-dependent* components are identified which have to be adapted in order to perform integrated retrieval and filtering processes. Both aspects of this general model are depicted in Figure 6.2. Two components are customized in order to facilitate integrated information seeking:

Integrated Representation This component enables at a mutual enhancement of individual information collections with collaborative search contexts representing dynamic and static information needs. ‘Search context’ is chosen as a broader term in order to

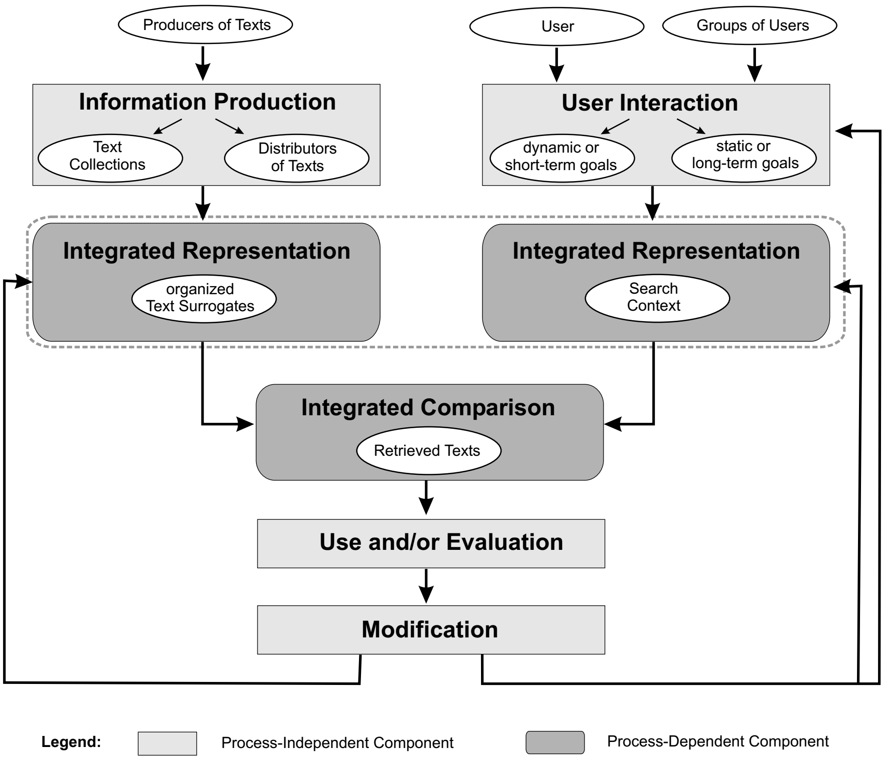


Figure 6.2: A General Model of Integrated Information Seeking

avoid an explicit reference to the term ‘query’ used in the information retrieval process and the term ‘profile’ used in the information filtering process. An integrated representation is achieved by the model of distributed Peer Search Memories and Virtual Knowledge Communities. Peersy is a dynamic information collection that is continuously updated with new relevance feedback. The index logs all updates with timestamps. In addition, all relevance judgments are exchanged among users. This information is used for a personalized ranking, as well as for the discovery of collaborative information needs.

Integrated Comparison This component is process-dependent that means techniques for integrated information retrieval and information filtering are distinguished. The component relies on an integrated representation of information collections and information needs. In general, this component is based on the cooperation of all users. Techniques for an integrated comparison combine the personalization and the collaboration aspects elaborated in the previous chapters. The following sections detail how this integrated representation is exploited for integrated information retrieval and filtering.

6.2 Integrated Information Retrieval

A *cooperative pull-push cycle* is a general interaction model for all users who act either in an active or passive role. In the role of an information consumer, a user is regarded as active, and he formulates a query for his information need. For integrated information retrieval,

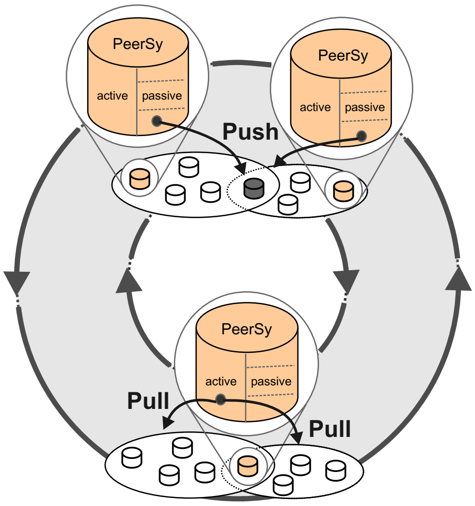


Figure 6.3: Cooperative Pull-Push Cycle

information providers are selected to answer the query. The selection process includes all associations locally stored in other Peer Search Memories or organized in Virtual Knowledge Communities. To do so, the first step of integrated information retrieval is to exploit a virtual search network. After this step, a reputation-based ranking is performed to add a relevance score to documents, users, or communities.

6.2.1 Virtual Search Network

All user interactions during integrated information retrieval are used to define a *virtual search network*. An interaction is either a consuming or a providing task. All Peer Search Memories are vertices (V) in the virtual search network. It is defined as a graph $G = (V, E)$. An edge $(u, v) \in E$ between two PeerSies exists if they successfully exchange information (see Figure 6.4). The success depends on whether the answer document is stored in the Peer Search Memory or not. We calculate the success of all answers I of peer u for peer v with

$$succ(u, v) = \frac{1}{|I|} \sum_{i \in I} upr_i(v), \tag{6.1}$$

The success is the average user-centered precision of peer v. The satisfaction depends on all answers I of peer u. The user-centered precision of peer v for an answer i that consists of a list of ranked documents,

$$upr_i(v) = \frac{rel_i}{S}, \tag{6.2}$$

where rel_i is the number of documents for which the user has provided relevance feedback by storing them in his Peer Search Memory. S is the number of documents that have been viewed for the list of ranked documents. Intuitively, the value expresses the effort of each user to find relevant documents. A high effort correlates with a low user-centered precision.

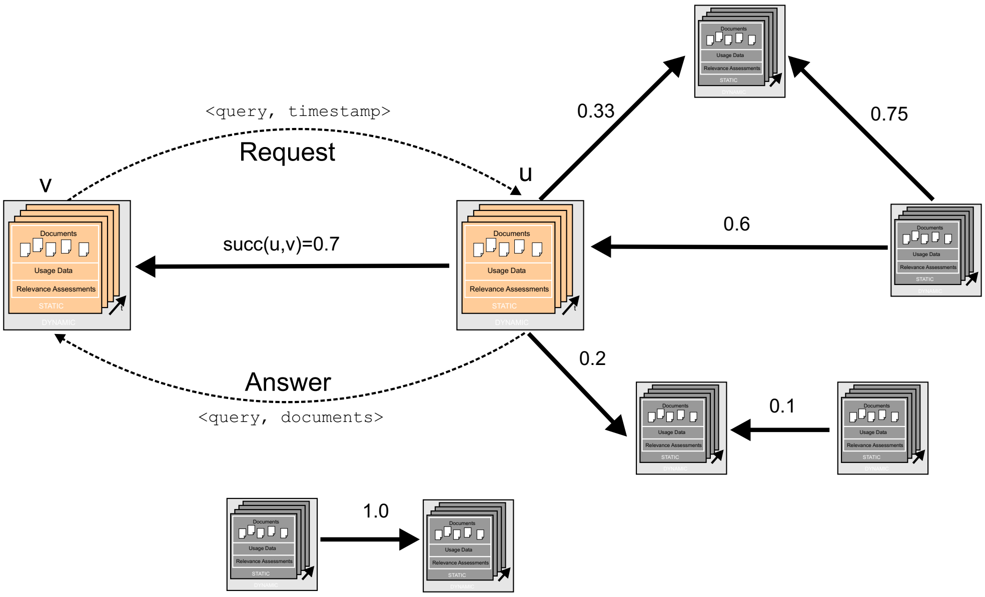


Figure 6.4: User Vicinity of the Access Network

In the virtual search network, all internal information collections are linked in a weighted directed graph. Each vertex has a reputation based on successful interactions. For the computation of the reputation, Web community approaches are considered (see Section 2.3.3) in order to implement a global method. HITS (Kleinberg, 1999) and PageRank (Brin and Page, 1998) are global methods to compute a score for a Web page. The intuitive description of these methods shows that a page has a high rank if the sum of the ranks of its ingoing links is high. This general method can be applied to the virtual search network where each vertex has ingoing $i(u)$ and outgoing $o(u)$ links. In analogy, the reputation of a peer is high if the sum of the reputations of its ingoing links is high. In general, a basic linkage analysis does not consider weighted links. We designed that each edge is annotated with its success. This weight specifies the ratio of all previous successful interactions. Hence, the general PageRank computation is enhanced by this value. We define the reputation recursively with the equation,

$$r(v) = (1-q) \sum_{u \in i(v)} r(u) \cdot \frac{succ(u,v)}{normO(u)} + q \, \frac{1}{|V|}, \tag{6.3}$$

where $i(v)$ are all ingoing edges of a vertex v, q is the probability that an interaction takes place with a random peer. $normO(u)$ is a normalization factor for the success weight. The sum of all weights of outgoing edges yields 1. It is computed with the equation

$$normO(u) = \sum_{v \in o(u)} succ(u,v) \tag{6.4}$$

where $o(u)$ is the set of all outgoing edges of a vertex u. Kamvar et al. (2003) showed that PageRank can be efficiently be computed in peer-to-peer networks. After the calculation of the reputation, each peer has an assigned reputation.

6.2.2 Exploratory Network Analysis

The Weblog corpus does not provide information about user interactions during search. Nevertheless, we can use repetitions of queries, links, and query-link associations in order to simulate interactions. If a repetition is detected, we classify it as an implicit interaction because both users are not aware that they share the same information. It is not possible to infer the success of a simulated interaction without the incorporation of user feedback. In Equation 6.3, the reputation of a peer is computed with an enhanced PageRank formula that uses weighted edges. If the success weight is set to one, the equation degenerates to the traditional formula. Thus, we compute PageRank for a network of users with three settings:

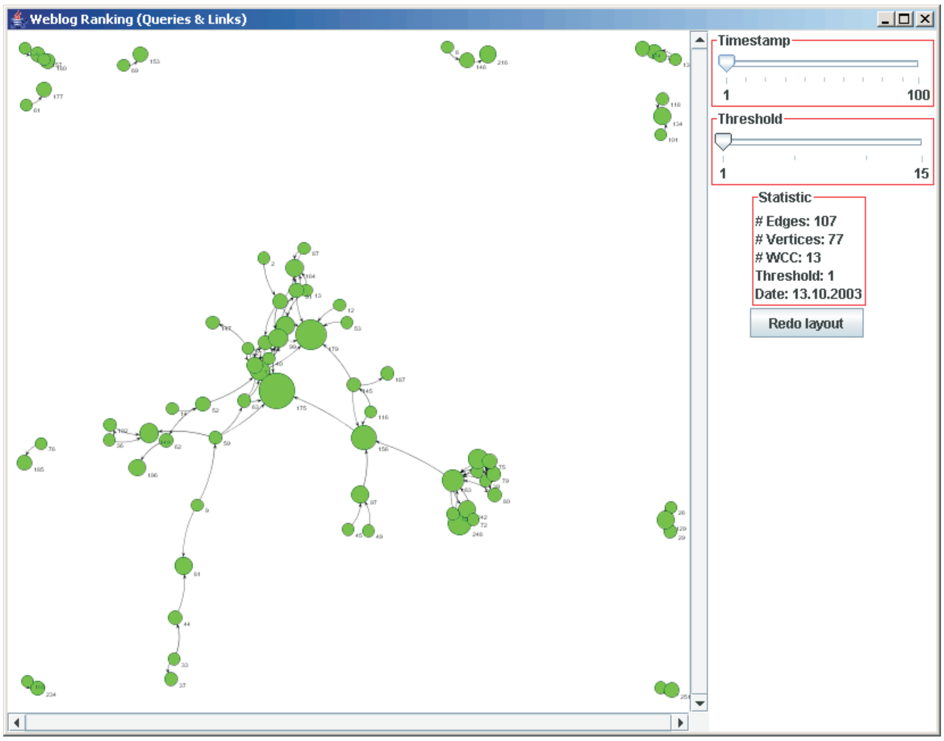
- Two peers are connected with an undirected edge if one peer repeats a query that the other peer has used before.
- Two peers are connected with an undirected edge if one peer views a link that the other peer has also viewed before.
- Two peers are connected with an undirected edge if one peer repeats an association between a query and a link that the other peer has used before.

For each setting, we considered a repetition threshold μ. This parameter defines the minimal number of repetitions between two peers before an edge is added. We started the simulation with a setting where all interactions are considered ($\mu = 1$). We used the PageRank implementation of the open-source Java Universal Network/Graph Framework (JUNG)[1]. The probability q was set to 0.15. At each day of the log data, we computed a PageRank score for a peer of the network.

First, we present the network of simulated user interactions after the first week of log. All users are linked that share at least one identical query-link association ($\mu = 1$). Figure 6.5 shows the network on October 13, 2003. The green color of all nodes shows that the PageRank has grown since the last computation step. We observed 107 edges and 77 vertices. Larger vertices represent larger PageRank values. At the beginning of the simulation, we observed 13 weakly connected components (WCC). A weak component of a directed graph is a subgraph so that the corresponding subgraph in the underlying undirected graph is connected. Over the time, the simulated interaction of users grew. All three settings identify users with different authorities. The PageRank of a vertex is used as authority measure.

On August 14, 2005, we noticed 1018 edges and 295 vertices. In Figure 6.6, we observed only few users who had a large PageRank. The majority of nodes are red colored which means that their PageRank has decreased since the last time unit. For all green colored nodes, PageRank has increased. Blue nodes shows that the PageRank did not changed since the last computation. The plot shows that the network is quite diverse. We present all edges and authority values with a minimal repetition threshold of three. For lower values, the network has 13,097 edges, so that is can not be explored. Despite the high number of edges, the number of weakly connected components is three. This number decreases to two for an identical setting that analyzed only query replications (see Figure 6.7) or link replications (see

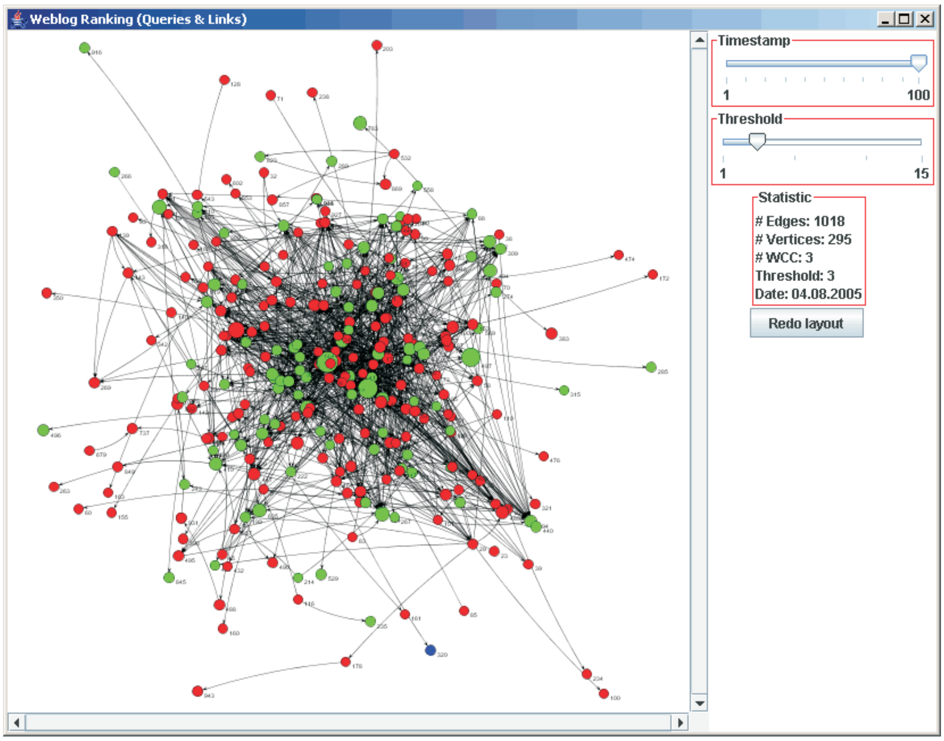
[1] `http://jung.sourceforge.net/`



Weblog Ranking (Queries & Links)

Edges: 107
Vertices: 77
WCC: 13
Threshold: 1
Date: 13.10.2003

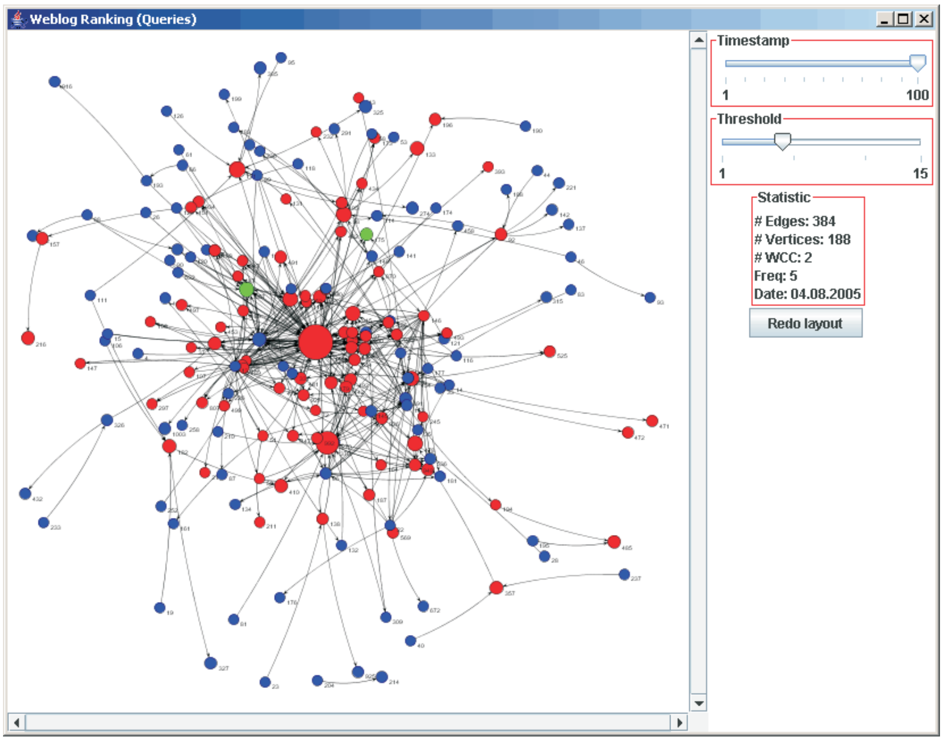
Figure 6.5: Simulated Interaction based on Queries and Links (October 13, 2003)



Weblog Ranking (Queries & Links)

Edges: 1018
Vertices: 295
WCC: 3
Threshold: 3
Date: 04.08.2005

Figure 6.6: Simulated Interaction based on Queries and Links test (August 4, 2005)



Weblog Ranking (Queries)

Timestamp: 1 – 100

Threshold: 1 – 15

Statistic
Edges: 384
Vertices: 188
WCC: 2
Freq: 5
Date: 04.08.2005

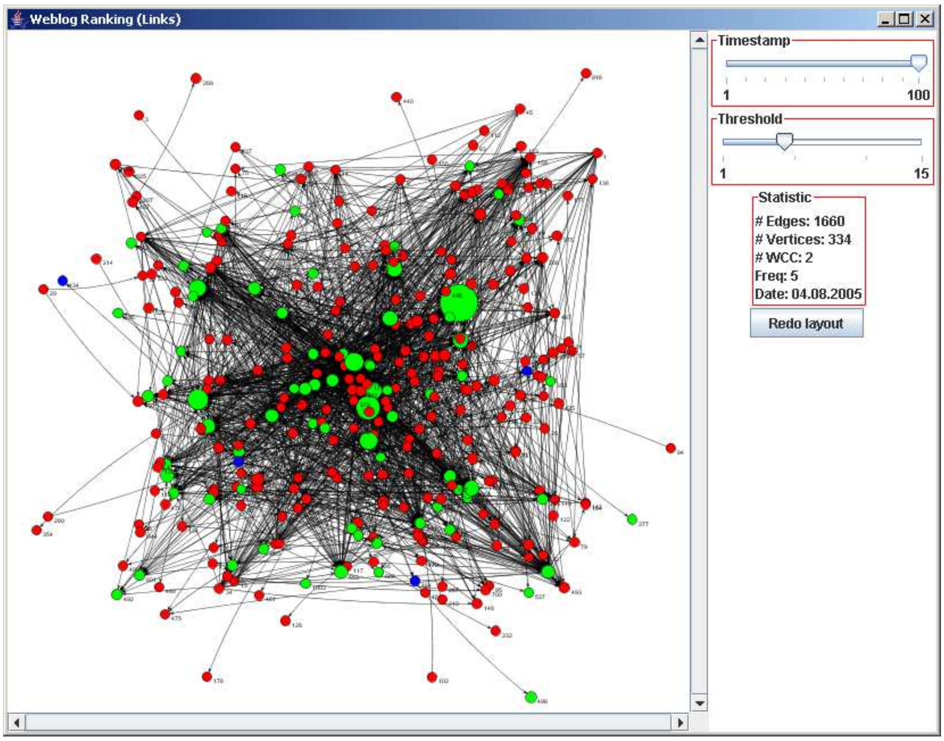
Figure 6.7: Simulated Interaction based on Queries (August 4, 2005)

Figure 6.8). We noticed a moderate increase of edges and vertices if only identical queries were considered. This observation is correlated to the general number of duplicated queries (see Section 4.1.2). We observed a fewer number of identical queries than identical links. Without a repetition threshold, the number of duplicated links leads to a diverse network. We plot the network in Figure 6.8 with a repetition threshold of five. This threshold has also been applied to the query-based network in order to compare the settings. On one hand, only a small number of queries are duplicated and on the other, a strongly connected network results based on link repetitions. In general, all plots show different settings to infer authority measures. For many peers, a reputation can be calculated, and it can be used to discriminate among peer results.

6.2.3 Reputation-based Ranking

We apply a reputation-based ranking to retrieve items in the virtual search network. The virtual search network allows to search for documents, users, and communities. The incorporation of the reputation measure takes place in three steps, before a ranking can be accomplished.

[1st Step] We compute the reputation r_i for each vertex i that represents a user and his Peer Search Memory. Several disconnected components may exist in this network. Since they are small compared to the giant component, we expect that they contribute little to the search result. We conjecture that peers which are not in the giant component will only be relevant



Edges: 1660
Vertices: 334
WCC: 2
Freq: 5
Date: 04.08.2005

Figure 6.8: Simulated Interaction based on Links (August 4, 2005)

for very few of the given queries. We use a probability of $q = 0.3$, further ameliorating the problem.

[2nd Step] The reputation score is assigned to documents and communities. Users have already a reputation because the computation is based on their individual Peer Search Memory. The assignment of the score to documents r_d and communities r_c differs in the set of associated users:

- If several user provided feedback for a document, we accumulate the reputation scores of all assessors.
- For a community, we accumulate the reputation scores of all members.

[3rd Step] A modified vector-space model is employed as a relevance measure for documents as regards a query. We use a freely available implementation based on the the Lucene[2] library, an open-source information retrieval library. The main scoring formula Lucene's of modified vector-space model is

$$rel(q,d) = \frac{\sum_{t \in q} \sqrt{tf(t,d)} \cdot idf(t)^2}{\sqrt{\sum_{t \in q} idf(t)^2} \sqrt{\sum_{t \in d} tf(t,d)}} \tag{6.5}$$

[2] http://lucene.apache.org, last visit on 2006/03/01.

where

$$idf(t) = \log \frac{|D|}{df(t)+1} + 1$$

Scores are normalized to fall in a range of 0 to 1.

For a query q, the text retrieval component produces a set of relevant documents, as well as a score for every document. The inclusion of the reputation does not affect the result set. It only influences the ranking of the documents, users, and communities.

There are several models for combining scores with a text retrieval system. A simple method of combining reputation and relevance scores as proposed by (Kirsch, 2005) is applied by

$$r_X \cdot \mathrm{rel}(q,d) \ . \tag{6.6}$$

where X refers either to a document-specific r_d, user-specific r_i, or community-specific r_c score. Both scores are independent of parameters which must be optimized.

6.2.4 Evaluation

This evaluation setting measures the impact a user's reputation for the retrieval process. With the Weblog data, we cannot provide a virtual search network based on explicit feedback. Instead, we applied a social network associating a social rank with each user. A social rank is comparable to our reputation measure, but does not incorporate a dynamic interaction between users. Instead, we use a static network without incremental updates. Nevertheless, we expected first insights about the combination of a global authority measure and a relevance measure. The social rank was computed in analogy to the exploratory network analysis. A automated evaluation methodology was developed in order to measure the influence of a social network on retrieval effectiveness. For such an evaluation method, we identified three prerequisites:

1. Selection of a document corpus incorporating a social network.
2. Providing relevance assessments for a set of queries and relevant documents.
3. Definition of evaluation measures for a user-centered retrieval performance.

Evaluation Methodology

We defined the following automated evaluation methodology which considers all three prerequisites:

1. We selected the mailing list archive 'origami-l'[3] from the years 2000–2005 as a document corpus. The archive contains 44,108 messages written from 1,834 different email addresses. Furthermore, we extracted a subset of the document corpus with messages

[3] `http://origami.kvi.nl`, last visit on 2006/07/22.

from 2004. It is used to compare our experimental results of the full corpus with a smaller set. For all messages, we constructed a full-text index from the message body, after removing quoted parts. These messages defined the content-based part of the evaluation corpus. In addition, a network was constructed based on the linkage information among all messages. This network was defined by two types of vertices and two types of edges, which were identified as follows:

Vertex Each author is a vertex in the social network. We assumed that an email address identifies distinct users. Also each message is a vertex of the network.

Edge The first type of edge links a message and its authors and vice versa. In addition, we identified edges between authors based on how often they respond to one another's messages.

In Section 2.3.3, we discussed several characteristics typical for social network. The characteristics of our extracted social network make it a 'small-world network' (Watts and Strogatz, 1998). On one hand, we observed a high degree of clustering and short average shortest path lengths. On the other hand, 70% of all authors are part of a giant component and the degree distribution follows a power law. For more details of the network analysis see (Kirsch, 2005).

2. We selected appropriate query terms from the subject lines of email messages. They are a good indicator of user information needs. We extracted frequent bi- and trigrams, because we observed that 'real-world' queries in our Weblog corpus have an average query length of 2.41. We filtered all frequent n-grams in order to detect the n-grams which are not highly correlated with the author of the containing messages. This correlation was measured with the mutual information of the occurrence of a specific n-gram in the subject line and the author of the message (Kirsch, 2005):

$$\text{score}(n\text{−gram}) = \frac{I(n\text{−gram}, \text{author})}{\text{df}(n\text{−gram})}$$

All n-grams were sorted by mutual information divided by the document frequency. We used the 10 n-grams with the lowest score as query terms for our evaluation. In case of an overlap between n-grams, the longest n-gram was chosen.

3. For each of the ten queries, one message was chosen as 'known-item'. We restricted the setting only to one relevant message, because we wanted to measure the document-specific changes in the ranking. The most relevant message was selected by an expert (master student) and by a complete novice (author of the dissertation) in the subject matter. Both relevance assessments allow us to evaluate whether reputation-based ranking assists either novice users or experts. We conjectured that novice users expect more general results because they did not know about the author's authority within the specific community. We borrowed a standard evaluation measure in order to evaluate the effectiveness of reputation-based ranking. Owing to the selection of one relevant document, we were able to evaluated the techniques in a known-item retrieval setting and compared them to a baseline technique, in our case a modified vector-space model. The applied evaluation metrics are average rank and inverse average inverse rank (IAIR) (see Section 4.3.2).

Table 6.1: Known-item retrieval on mailing list data. Columns labelled 'VS' contain ranks from vector-space search, and columns labelled 'PR×VS' contain ranks scored by PageRank times vector-space score. Rows '$\overline{\text{rank}}$ change' and 'IAIR change' contain the change compared to the baseline method 'VS' in percent.

method:	VS	PR×VS	VS	PR×VS
searcher:	expert	expert	novice	novice
		on messages from 2004:		
$\overline{\text{rank}}$:	14.75 ± 0.25	17.95 ± 0.05	17.5 ± 0.3	15.2 ± 0
$\overline{\text{rank}}$ change [%]:		$+21.7 \pm 2.4$		-13.1 ± 1.5
IAIR:	7.548 ± 0.032	7.082 ± 0.010	4.670 ± 0.013	4.599 ± 0
IAIR change [%]:		-6.2 ± 0.5		-1.5 ± 0.3
		on messages from 2000–2005:		
$\overline{\text{rank}}$:	24.4 ± 0.3	41.45 ± 0.05	39.35 ± 0.35	39.6 ± 0
$\overline{\text{rank}}$ change [%]:		$+69.9 \pm 2.3$		$+0.6 \pm 0.9$
IAIR:	8.787 ± 0.040	6.697 ± 0.012	4.962 ± 0.013	7.86 ± 0
IAIR change [%]:		-24.6 ± 0.5		$+58.4 \pm 0.4$

In general, the design of a new evaluation methodology requires a high amount of manual work. A known-item retrieval setting reduces this work and it allows a semi-automatic selection of items. We obtained a baseline method for a comparison with our approach in order to be independent of external factors influencing the performance. Thus, only the impact of reputation-based ranking on retrieval performance was measured.

Evaluation Results

Table 6.1 (Kirsch et al., 2006) shows results of the automated evaluation methodology for the full corpus and its subset with messages of 2004. First of all, we present the results for items chosen by an expert searcher. Indeed, the data shows that the combination of PageRank and the vector-space model performs better than the vector-space model alone for four of ten queries on the 2004 corpus. Only in one case, the result is a draw. Furthermore, we observed that the average rank of the found documents increases by 21.7% ± 2.4 for PageRank search and the inverse average inverse rank decreases by 6.2% ± 0.5. The results show that some documents were found considerably later than with vector-space search. In addition, PageRank combined with vector space performs better for those documents in the earlier parts of the result list. On the large document corpus from 2000–2005, this effect is even more pronounced. The average rank increases by 69.9% ± 2.3, but the inverse average inverse rank decreases by 24.6% ± 0.5. The combination of PageRank and the vector-space model performs better for six out of ten queries.

The results of items chosen by a novice searcher are less pronounced. At first, we observed a decrease of both the average rank (13.1% ± 1.5) and inverse average inverse rank (1.5% ± 0.3) on the smaller corpus from 2004. Furthermore, PageRank times vector space performs better for five out of ten queries, with one draw. On the larger corpus, the average rank is unchanged, but the IAIR increases faster (by 58.4% ± 0.4.). We observed for the larger set a better performance for four out of ten queries, also with one draw.

6.2.5 Discussion

In general, integrated information retrieval is based on a social model of all participants. There are two general models to build a social network:

- The current interest in ’social software’ can be exploited in order to collect information about a user’s social environment. Yahoo Communities[4] enables a user to identify his friends explicitly in order to incorporate their annotated search results in his search. The goal to share contacts is realized amongst others by the platform Open Business Club[5] (OpenBC) very successfully. For both services, each user has to abandon his anonymity.

- A relationship network can be generated in a pure content-based manner. The individual information collections of peers can be compared in order to find similarities among users.

In this dissertation, we developed an interaction model that is independent of real social relationships among users. The virtual search network maintains reputations of users based on their interactions. For both cases of building a social network, the quality of the resulting network is crucial. The first evaluation results showed that a poorly formed social network can lead to a failure of the retrieval method. The reputation of a peer may be misleading as regards his authority. In particular, the reputation-based ranking is good at identifying authorities. A larger evaluation setting is needed to verify the significance of the first results.

6.3 Integrated Information Filtering

Integrated information filtering aims at an awareness of documents which have been found by other community members. Each community is continuously updated with new validated documents by its members. The explicit membership is a common agreement that documents are exchanged among all community members according to the cooperative pull-push cycle (see Figure 6.3). No central document index exists for a community. Hence, integrated information filtering provides a decentralized prediction of relevant documents. The privacy and the autonomy of each peer is protected, because only community-specific information is considered for recommendation. In order to differentiate between both functions of a peer, information providing and information consuming, two peer descriptions are introduced: (1) a peer that expects recommendations is called *active peer*, and (2) a peer that recommends documents is called *passive peer*.

[4] `http://myweb2.search.yahoo.com/`, last visit on 2006/03/01.

[5] `http://www.openbc.com/`, last visit on 2006/03/01.

6.3.1 Prediction Process

The prediction process evaluates all recommendations of passive peers according to (Gül, 2004). All community-specific recommendations are organized by the community manager of each client peer. Implementation details of this selection process and the generation of requests and answers are discussed in Section 7.3. Owing to a periodic pull, a community-specific selection of likeminded users increases the awareness of new relevant documents. Information flooding can be avoided, because only a subset of all users is requested. The individual information collections of community members are used for a combination of content-based and collaborative filtering approaches.

From each peer, documents that are associated with a community are weighted by a content-based approach. In addition, they are temporally stored in a *service repository* which collects all documents which are recommended by peers or which will be recommended to other peers. The repository is continuously updated with new documents. The initialization process of the repository is detailed in Section 7.1.3. The main goal of the initialization is the calculation of a document weight by summing over all document terms and their occurrences. Each weight is part of an implicit rating of the document. The Peer Search Memory does not collect an explicit rating of the user who assigns document rates of numerical scale, for example 1 to 5 stars. Thus, identical documents on different peers can have identical weights due to similar data sets of each peer. In order to incorporate the individual quality of a user's recommendation, the pure content-based rating approach is enhanced by the quality of prior recommendations. Thus, differences among users are compensated by a *trust* value. It is an inverse measure to the reputation (see Equation 6.3) of a peer. The reputation represents the global trust of a user group in this peer. The local trust of a peer v in a peer u within a community c is calculated with

$$t_c(v,u) = \frac{arc_c(u)}{\sqrt{\sum_{w \in \mathcal{U}_c} arc_c(w)^2}}. \tag{6.7}$$

and $arc_c(u)$ is the number of accepted recommendations from peer u by the active peer v. As a normalization factor, we consider all accepted results of all community members $\mathcal{U}_c$. The prediction of relevant documents is performed on the active peer after community members have answered. The answer process is detailed in Section 7.3.3.

The community-based filtering approach is a two-step filtering on the local peer, as well as on the community. First, in a decentralized manner all passive peers of a community are filtered by means of a content-specific selection of information. Second, a nearest neighbor algorithm predicts relevant documents for each user individually.

Definition 6.1 *(Community-based Filtering) For an active peer v with a membership to the community c, a prediction for a recommended document $d \in \mathcal{D}$ is computed over all passive peers $\mathcal{U}_c$ of the community c*

$$\mathit{cof}(d) = \overline{w_v} + \frac{\sum_{u \in \mathcal{U}_c} (w(v,u) + t_c(v,u))(w_u(d) - \overline{w_u})}{\sum_{u \in \mathcal{U}_c} (w(v,u) + t_c(v,u))}$$

The cof weight is used for a ranking of all recommendations. In this regard, a high cof weight represents a high relevance of a result d. The similarity of two users is computed with standard similarity measures. Like collaborative filtering approaches, we use the Pearson correlation coefficient (Resnick et al., 1994) by considering the implicit ratings of two peers v and u:

$$w(v,u) = \frac{\sum_{d \in \mathcal{D}} (w_v(d) - \overline{w_v})(w_u(d) - \overline{w_u})}{\sqrt{\sum_{d \in \mathcal{D}} (w_v(d) - \overline{w_v})^2 \sum_{d \in \mathcal{D}} (w_u(d) - \overline{w_u})^2}} \tag{6.8}$$

During initialization all documents associated with a community are rated by a query-document similarity. The average value of a document rating is computed on the active peer. All recommendations of a peer u are considered in order to compute $\overline{w_u}$ with

$$\overline{w_u} = \frac{1}{|\mathcal{R}_u|} \sum_{s \in \mathcal{R}_u} w_u(s) \tag{6.9}$$

where $w_u(s)$ is computed during the initialization of the service repository (see Section 7.1.3). $\mathcal{R}_u$ is set of documents for which relevance feedback has been provided by user u. The local statistics of each passive peer is sent with each recommendation according to the protocol which is defined in Section 7.3.3.

The prediction process is completed after the computation of the cof weight. With the community-based filtering approach, each recommended document is ranked and the result list is presented to the user. The user interface is detailed in Section 7.4. The interface allows the user to provide feedback for documents of the recommendation list. Each document that is stored in the Peer Search Memory updates the local trust of the recommending peers.

6.3.2 Evaluation

This evaluation task measures the quality of recommended documents. The quality is calculated with respect to all community-specific documents collected by a member. An automated evaluation methodology is developed to determine the quality of community-based filtering with respect to a document similarity.

Evaluation Methodology

The automated evaluation methodology is designed to gain insights into the following scenario of integrated information filtering:

A user is member of a community and receives new recommendations from the community-based filtering approach. The quality of the recommended documents depends on his prior search sessions associated with the community.

For this scenario, an evaluation corpus must provide data about communities and their members. In Section 5.4, we showed how the Weblog corpus was used to identify communities.

In addition, this data was used to extract for each member his search sessions associated with the community. We selected the K-Sim community setting with a update threshold of 0.5. Within the set of communities, we chose only communities which had more than two members. For each community, the initialization and recommendation process took place in four steps:

1. *Active Peer Selection:* We selected a community member as the active peer if he had the minimal number of community associations. In addition, we extracted all associations with which he contributed to the community. The Peer Search Memory and the service repository of the active peer were initialized with all community associations.
2. *Passive Peer Initialization:* All members of the community except the active peer were summarized to one passive peer. We initialized the passive peer with a Peer Search Memory that stores all community associations of the grouped members. The unification of all passive peers was done to simplify the evaluation methodology. We did not consider the characteristics of a distributed system architecture in this evaluation task.
3. *Selection:* This process was divided into two phases. First, the active peer computed a set of query terms describing its community-specific information need. Second, these terms were used to query the passive peer. Each passive peer recommended all documents matching the query terms.
4. *Prediction:* During the prediction process, we calculated a score for each recommended document (Equation 6.1). We applied no local trust, because this user-driven parameter is not available in the Weblog corpus.

We processed all four steps for 98 communities. For each active peer, a set of recommended documents was selected. The quality of the recommendations was measured with the ODP data set. For each active peer, we randomly selected a reference category with the largest set of links of the same category. It was used to compute the familial distance (see Section 5.4.4) between the user's personal community interest and all recommended documents. The familial distance classifies recommended documents into four categories: same, sibling, cousin, or unrelated. All documents of the first three categories were counted as similar documents, because the statistic in Section 5.4.2 showed that the majority of communities links are of the same category, and sibling and cousin links have only a proportion of 10%. The recommendation quality is measured with:

Recommendation Precision is the ratio of the number of similar links to the total number of recommendations on an active peer.

Recommendation Recall is the ratio of the number of similar links to the total number of all similar links in the current recommendation list or in others.

The normalized recall is computed in two steps: (1) The number of correctly recommended documents is computed by multiplying the total number of recommended documents and the precision. (2) This value is normalized by dividing it with the maximum number of correctly recommended documents.

Table 6.2: Recommendation quality on community data. Column labelled ‘VS’ contains results scored by vector-space, column labelled ‘CBF’ contains recommendations from community-based filtering, column labelled ‘CBF5’ contains recommendations based on an expansion of 5 search engine results scored by community-based filtering, column labelled ‘CBF10’ contains recommendations based on an expansion of 10 search engine results scored by community-based filtering. The change is compared to the baseline method ‘VS’ in percent.

method:	VS	CBF	CBF5	CBF10
avg. recommendation precision:	0.84	0.85	0.75	0.72
avg. recommendation recall:	0.98	1.0	0.88	0.84
change[%]:		+1.9	-10.6	-14.1

We compared the community-based filtering approach to a baseline method, in our case a modified vector space model. As a baseline, we ran the experiment with a peer that maintains a local central index. To build the index, we crawled 96,438 HTML documents of the Weblog corpus and indexed them with the text search engine library Lucene. For each query generated by the active peer, this method retrieved a set of documents. The baseline considered no community aspects and personal search histories. We compared this baseline technique with the community-based filtering approach on three different settings. Each setting considered different data sets of the passive peer:

CBF The passive peer includes all query-link associations of its associated members. This setting corresponds to the basic initialization of the Weblog corpus.

CBF5 This setting expands the basic initialization of the passive peer. It is updated with the first five hits of a Web search engine, in our case Google. The search engine was asked with query terms generated by the active peers.

CBF10 In analogy to CBF5, we associated additional documents with the passive peer. We initialized query-document associations with the first ten hits of a Web search engine.

The expansion of the passive peer should show insights how a selection of documents without any feedback influences the recommendation quality. We ran the experiments for each community and measured the recommendation quality for an active peer. In order to compare the results of an active peer with different settings, we considered the minimal number of recommendation for the active peer in all settings. For example, if one peer has got only one recommended document with the pure CBF setting, we measured the recommendation quality only for the first recommended document in all other settings (baseline, CBF5, CBF10). The performance of the approach is measured with the average recommendation quality of all active peers.

Evaluation Results

Table 6.2 shows results of the automated evaluation methodology for 98 K-Sim communities (threshold 0.5). With each setting, an active peer received a set of recommended documents.

For the baseline approach, we restricted the maximal number of results to 20. Without a normalization of the recommendation lists, the recommendation quality of two settings is not comparable. We normalized the recommendation lists according to the CBF setting. In 64.5% of all cases, an active peer receives only one recommended document with the CBF setting. The maximal number of recommendations for each active peer was used to limit the result set in all other settings. Without a limitations, an active peer receives 5.1 recommendations on average with regard to the CBF10 setting. The pure community-based filtering approach, recommended 49 document to 31 active peers (on average 1.58 documents). All other peers did not receive a recommendation in this setting. The total number of correctly recommended documents is 41. We applied this value to compute normalized recall. Owing to the normalization of the recommendation list, we observed that the change between average precision and recall compared to the baseline is identical.

The baseline method shows an average recommendation precision of 0.84. We observed that a community-specific collection of user judgments has a marginal higher recommendation precision, as well as recall. Both measures increase by 1.9%. We did not apply a statistical significance test in order to verify if this increase is significant. Moreover, these results show that a community-specific grouping of topics performs as good as a central data collection. This observation is important for peer-to-peer networks, because no central server is available. It shows that a community-specific selection is as effective as standard methods for central indexes. In a distributed network architecture, a community-specific selection of peers would decrease the network load significantly. Community-based filtering recommends documents with a high similarity to the personal user interests. The significance of implicit relevance feedback is even more pronounced if we compare different settings with the baseline and the CBF setting. For both settings with an expansion of the passive peer, we observed a decrease by 10.6% and 14.1% on average recommendation precision, respectively. The effect on normalized recall is similar. We conjectured that the expansion of the passive peer is related to the general community's content, but does not match with the personal interest of the active peer. The poor results show that community-specific documents are not recommended any more on the first ranking positions. The term relevance of the additional documents is higher, but the topic is unrelated to the individual interest of the active peer. Further evaluations are necessary to analyze if the performance of community-based filtering depends on temporal shifts of the community interest. The local index is a self-contained data set to which the user actively contributed and communities have been discovered. The data set has not been updated since August 4, 2005 and new crawled documents are not represented in a community. The evaluation results of the known-item retrieval setting (see Section 4.3) showed the advantages of a persistent storage that can be shared with other users. A Web search index underlies a continuous update process that might influence the ranking. Specific documents get a higher rank due to the optimization process driven by users' click-throughs. This optimization is not done for a specific user. Hence, community-based filtering allows to control the recommendation process for each user individually.

6.3.3 Discussion

Community-based filtering (CBF) and collaborative filtering (CF) can be compared with the following aspects:

Objects: CF generally deals with static objects like movies, books, etc. In addition, CBF considers queries which are associated with links and users.

Ratings: With CF, users rate objects by assigning a numerical scale, for example 1 to 5 stars. Instead, CBF collects only explicit positive feedback. For all viewed documents without a rating, we cannot infer that the document is nonrelevant.

Differences between both approaches lead to a specific evaluation strategy which reveals that a community structure influences the recommendation quality. Each community provides explicit information about long-term search interests and of its members. In general, the quality of a recommendation process must be independent of the system architecture. The storage of ratings in a centralized or decentralized manner should not influence the effectiveness. The evaluation results mirror the effectiveness of a community-based filtering for a peer-to-peer network that is comparable to the effectiveness of a central server. An expansion of a community with new documents of a Web search engine that have no implicit feedback shows a poor performance. Top ranked documents decrease the similarity of recommended documents to the individual reference category.

In general, integrated information filtering depends on the activity of all users. If the community does not change or is not updated, no member enhances his community-specific information set. Only with an active participation of all members, a community will grow and an impact of community-based filtering is measurable. This growth must be maintained on a global and on local level. On a global level, an automatic expansion enhances the set of communities with respect to all terms and documents. A similarity search can be initiated with all community links. On a local level, a community must be expanded in order to consider personal interests. The evaluation results show that a global expansion alone is less pronounced. Hence, further evaluation is necessary to test a combined approach.

6.4 Summary

Integrated information seeking combines both processes, information retrieval and information filtering. A general model is defined that is applicable to both processes. An integrated representation is utilized to match rapidly changing information with variable interests due to a factorization of information needs and collections. The personalization strategy is used to represent all documents of the user group. In addition, the collaboration strategy allows users to organize common information needs. The combination of both techniques factors the problem space into static and dynamic information needs, as well as information collections. It defines a common basis for integrated information retrieval and integrated information filtering. A variety of techniques have been applied to each process.

Integrated information retrieval is based on an associative network model. All search processes define a virtual search network. User interactions are represented by queries and their answers. Vector-space retrieval is used as the underlying text retrieval method. The quality of these interactions is evaluated by a global authority measure defining a peer reputation. A reputation-based ranking combines authority scores with relevance scores from vector-space retrieval. The evaluation was carried out in a mailing list archive simulating an access network. The combination of both measures showed a first improvement.

Integrated information filtering combines content-based and collaborative filtering techniques. The new technique offers awareness of documents which have been found by other community members. According to the general interaction paradigm, the filtering process is based on the cooperative pull-push cycle. A peer generates a query to ask community members. They offer a set of recommended documents to the request. The documents get a relevance prediction considering the individual search history. To do so, the trust of peers in each other is calculated based on previous recommendations. The evaluation is carried out in a community setting extracted from the Weblog corpus. Performance was measured with the similarity between recommended documents and community-specific documents collected by a member. The community-based filtering approach showed the same quality as a baseline method for a central collection of all documents. For efficiency reasons, the pre-selection of information providers of a community is an improvement.

7 Prototype Implementation

This chapter presents the prototype implementation of the Congenial Web Search concept. In Chapter 3, the general system architecture has been introduced which is based on a peer-to-peer network. With JXTA, we use a de-facto standard for peer-to-peer applications. ISKODOR is an integrated application on top of the JXTA system layers. We grouped all system functionalities into four components. The main component is 'Integrated Information Seeking' which manages information retrieval and filtering processes. All techniques implemented in this component have dependencies on three additional application components. This chapter presents implementation details for the 'Data Access', the 'Community Discovery', and the 'Community Management' component. Finally, we discuss the graphical user interface which is a sub-component of the 'Integrated Information Seeking' component.

7.1 Data Access

In order to maintain an advanced access to information on the Web with a Peer Search Memory, three basic components are required (see Figure 7.1). In general, the user requires a search interface, a Web browser, and a local storage device (in our case a database). The complexity to maintain each component depends on the target group of users. The general ISKODOR architecture has a generic structure which is independent of a particular user group. Congenial Web Search aims at an application for the average user. The PeerSy architecture (see Section 7.1.1) provides the main components of a personalized search. It delegates all data requests to specific services. We incorporate data services that are maintained by external providers or other peers. For the local data access, a storage of relevance feedback (see 7.1.2) and a repository of documents (see Section 7.1.3) for recommendations are presented in this section.

7.1.1 PeerSy Architecture

The Peer Search Memory is implemented as a JXTA peer service. The interaction between users is performed by this service. All advanced search facilities attempted by Congenial Web Search rely on the Peer Search Memory. Without the personalization of individual search interests no interactions among users will arise. Figure 7.1 depicts all required components for a personalized Web search:

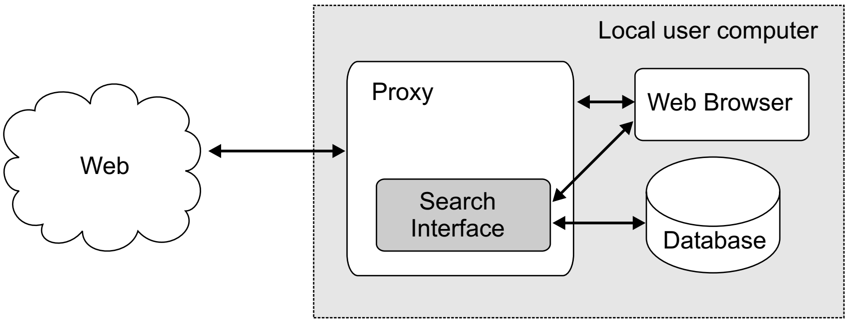


Figure 7.1: General System Components

Web Browser The application is independent of a specific Web browser. Unlike traditional Web search with a browser, a local proxy is used to monitor, edit, and generate HTTP data streams. ISKODOR uses the WBI[1] proxy where applications are added as plugins. The main task of this proxy is the logging of an actual URL presented in the Web browser. In addition, local user feedback can be given in the Web browser.

Search Interface The search engine is implemented as a plugin for the WBI proxy. It is written in Java and a detailed description of the interface design is elaborated in Section 7.4. The design principles rely on a standard user interface design. No specific features for novice or experts are considered.

Database The MySQL database server is used as an open source database for a persistent storage of individual feedback. In addition to feedback information, this database is used as an index for the local search engine. A summary of all meta-data stored during a search session is presented in Section 7.1.2.

All components that run at the local user computer provide access to local data, the Web, and the peer-to-peer network. The local data access is organized as follows:

- A query is processed by a Web search engine and a local search engine. The results of both systems are presented to the user. A Web browser assists the navigation and collects relevance feedback for Web pages. This feedback is stored in a local database, denoted as PeerSy core.
- PeerSy initiates an information push if new recommended documents are available. All recommended documents are stored in a services repository which is continuously updated with new documents. It also collects documents which are recommended to other peers.
- A user can manage his PeerSy in order to delete feedback associations or to administrate personal information.

The representation of all retrieved documents is based on a personalized ranking scheme (see Section 4.2.3). This scheme combines the results of a Web search engine and the local search

[1] http://www.almaden.ibm.com/cs/wbi/index.html, last visit on 2005/08/30.

Table 7.1: Facets and Terms for PeerSy's Local Search Engine

Facets	**Terms**
Conceptual Model	Vector Space Model
File Structure	Inverted Index
Query Operations	Parsing, Boolean, Feedback
Term Operations	Weight, Stopwordlist
Document Operations	Parse, Display, Rank, Field Masks, Assign IDs

engine. The Web search engine is an external provider due to the classification in Section 4.2.1 with no detailed information about the IR system available. The external provider retrieves documents for a user's query, whereas the local search engine retrieves documents with individual relevance feedback stored in the PeerSy core. This integrated IR system is classified by facets and terms proposed by Frakes and Baeza-Yates (1992) as presented in Table 7.1. A probabilistic model is used as a conceptual model. In a first step, all judged Web pages are parsed and terms are extracted which are organized with an inverted index.

7.1.2 User Profile Storage

All documents for which a user has provided explicit relevance feedback are stored in the Peer Search Memory. Each document is associated with the query that has been formulated to retrieve the document. The representation of query-document associations is implemented with a database. All tables and relationships of this database are represented in Figure 7.2.

In general, the tables `Query`, `Document`, and `Association` represent user-based aspects of the Peer Search Memory. All relationships between these entities in the PeerSy core are shown in Figure 7.2. Each query is represented in table `Query` with an assigned identification number (*id*). In this table, a normalized query is stored in order to achieve a common representation for similar queries (for example, 'java api' and 'api java'). The normalization is performed by sorting the query terms and removing identical terms. The original query is stored in table `SearchHistory`. In addition to the normalized query form, the number of index terms for this query is counted with *keywordsCount*. Usage-based aspects of a query are represented by temporal characteristics of an access, such as first (*firstAsked*) and last time of request (*lastAsked*). The total number of repetitions of a query is stored by *askedCounter*.

For each query, relevance feedback is collected by the browser or the search interface (see Section 7.4). The browser is enhanced with an additional button which the user can click if he found a relevant document during browsing. This feedback information relies on an explicit rating of a user. In addition, documents which have been viewed by the user, but without a relevance judgement are stored as implicit feedback in the table `SearchEngineResult`. For each query of the `SearchHistory` asked at a particular day (*askedDate*), all URLs which have been viewed by the user are stored in `SearchEngineResult`. This table models an association between a query, an external search engine, and the particular position of the URL in the search engine result list (*nr*). Documents without explicit feedback are not indexed for personalized access. Instead, all documents for which the user provided relevance

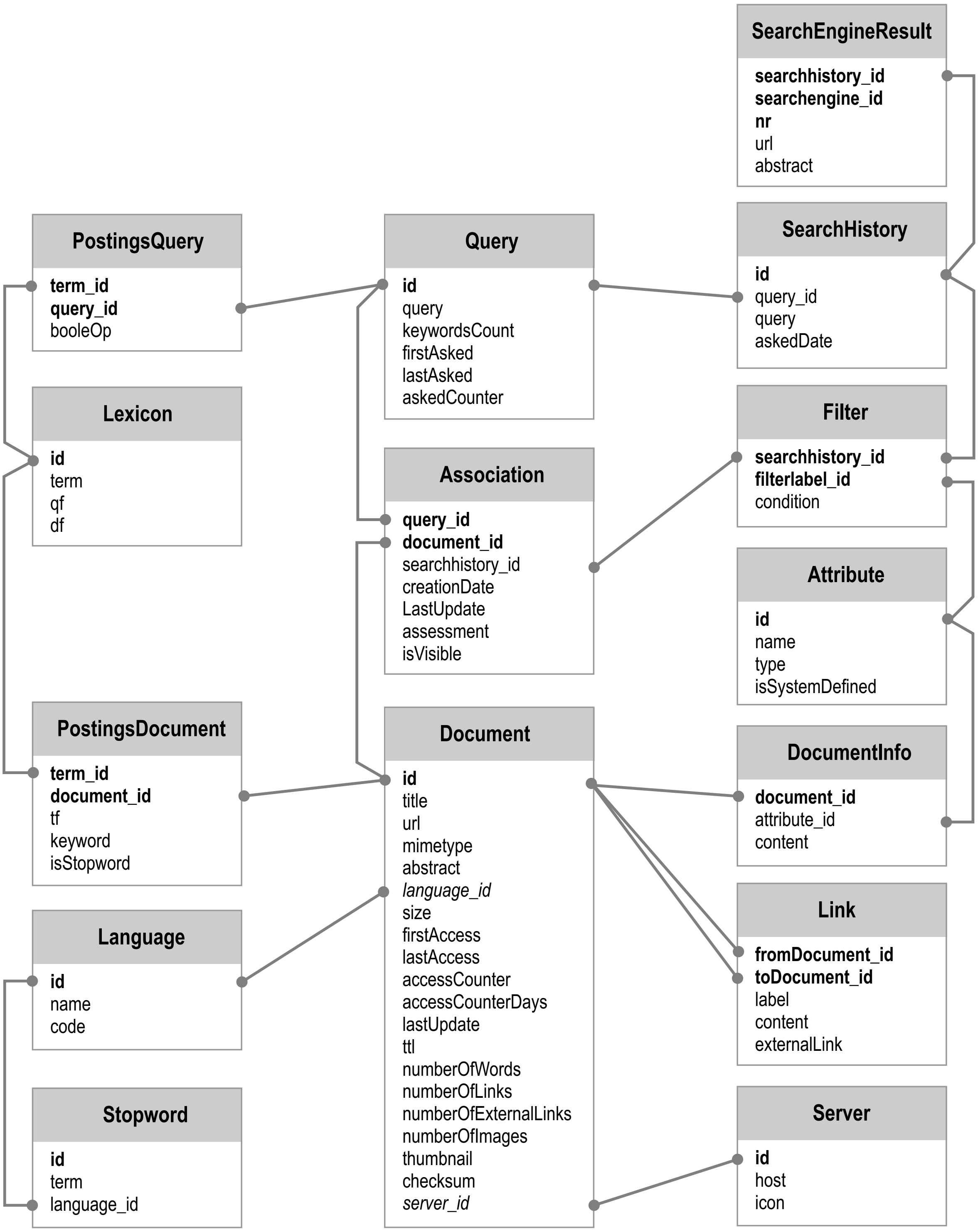


Figure 7.2: Relationship Diagram of PeerSy Core according to (Ruhl, 2003)

feedback get an association with the query in the table Association. The attribute (*isVisible*) manages the availability of the association for other users in the peer-to-peer network. This attribute can be set by the user during the feedback dialog or during administration of associations in his PeerSy.

All documents which are judged as relevant by a user are indexed. In table Document, each document gets an identification number and the URL representing the original location in the Web. In addition, usage-based aspects are added by several attributes. The inverted index is stored in table PostingsDocument. A tokenizer selects all terms of a document and stores them in the table Lexicon. In addition, we store three frequencies of a term:

- *Query Frequency*: The attribute *qf* counts the total number of occurrences of the term in all queries.
- *Document Frequency*: The attribute *df* counts the total number of documents with the term.
- *Term Frequency*: The attribute *tf* counts the frequency of the term in a specific document.

No linguistic models such as stemming are used to modify a term. The inverted index is built by a reference of a term ID and a document ID. Specific terms of a document are labelled as keyword (*isKeyword*) or as stopword (*isStopword*). We selected the 10 frequently used terms of a document which are not stopwords as keywords. We applied lists of stopwords for 11 different languages (Danish, Dutch, English, French, German, Italian, Norwegian, Portuguese, Russian, Spanish, Swedish). Each stopword and its language id is added in the table Stopword. All language ids are managed by the table Language.

7.1.3 Service Repository

The *service repository* is an extension of the PeerSy core. The service repository is designed to be a temporary storage of documents recommended by other peers. Only if a user provides explicit relevance feedback for a recommended document, it is permanently stored to the PeerSy core. Figure 7.3 depicts all tables and their relationships. They are updated continuously with new community memberships. Table PeerSetting summarizes all configuration settings of the filtering approach. The attribute *intervalTime* specifies the interval of the trigger initiating a periodic pull. All tables are initialized with documents which are associated to a community. The initialization accomplishes four steps:

1st Step: For each membership to a community, a peer group advertisements (see Section 7.2.2) is generated. All information about this community are analyzed and stored in the table PeerGroup. In addition, all other community members are discovered which are currently available. Owing to the dynamics of the peer-to-peer network, all actual members of a community must be updated before a broadcast message is sent to all peers. The table PeerProfile collects all peers which have been recognized as community members.

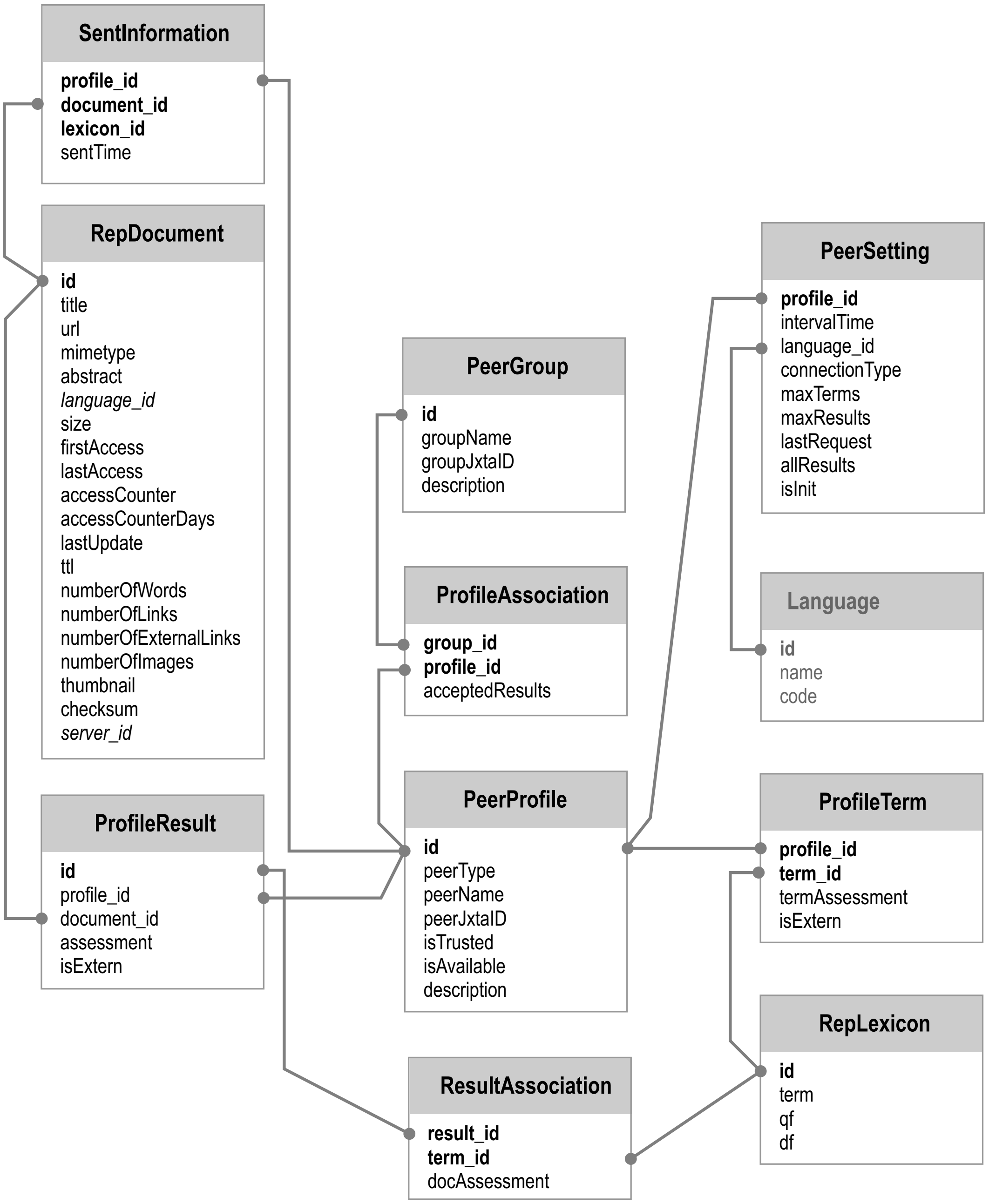


Figure 7.3: Relationship Diagram of the Service Repository according to (Gül, 2004)

2nd Step: All documents, which are associated with a community, are candidates for a recommendation. For this task, the tables `ProfileResult` and `ResultAssociation` represent a temporary preselection of documents from the PeerSy core. The attribute *isExtern* specifies whether the document is locally available, or has been recommended by a passive peer. During initialization only local documents are added to this table.

3rd Step: For all documents listed in table `ProfileResult`, a document's weight is calculated by summing over the document terms ($\mathcal{T}_d$), the product of the term weight (w_i) and the normalized term occurrence in the documents (tf_i).

$$w(d) = \frac{1}{A_d} \sum_{i \in \mathcal{T}_d} w_i \tag{7.1}$$

Each rating is normalized by the total number of query-document associations A_d with document d. It is stored with the attribute *assessment* in the table `ProfileResult`. A simple retrieval method is employed based on the traditional $tf \cdot idf$ weighting with cosine normalization (Salton and McGill, 1983) for the weight w_i of a relevant result d as regards the term i. It is calculated from the term frequency tf_i and the inverse term frequency idf_i, as well as the query frequency qf_i in the following formula:

$$w_i = (tf_i + qf_i) \cdot idf_i \tag{7.2}$$

The query frequency, qf_i, of term i is the number of occurrences of the term in all queries. All frequencies have been measured in the PeerSy core during the storage of explicit feedback (see Section 7.1.2). The rating of a term is represented in table `ResultAssociation`. The attribute *docAssessment* stores the document's score.

4th Step: Each collected peer profile (1st step) is associated with his community memberships in the table `ProfileAssociation`. The number of accepted, recommended documents of a peer is stored with the attribute *acceptedResults*.

Once a peer is initialized, each peer can provide and consume information. All peer service repositories build the platform for integrated information filtering (see Section 6.3). The 'Community Manager' initiates a cooperative pull-push cycle among community members. First, the active peer selects his community-specific interests and initiates a periodic pull from all available community members. Second, each request of an active peer contains recommended documents. Once documents are recommended by passive peers, a prediction process is initiated by the active peer to determine their individual user relevance.

7.2 Community Discovery

The community discovery component is a centralized service that collects all timestamped contexts. This service is part of the hybrid peer-to-peer network. The discovery process detailed in Chapter 5 is implemented on this peer. In addition, a storage device for the set all candidates is required. If a community candidate is detected, a community advertisement is

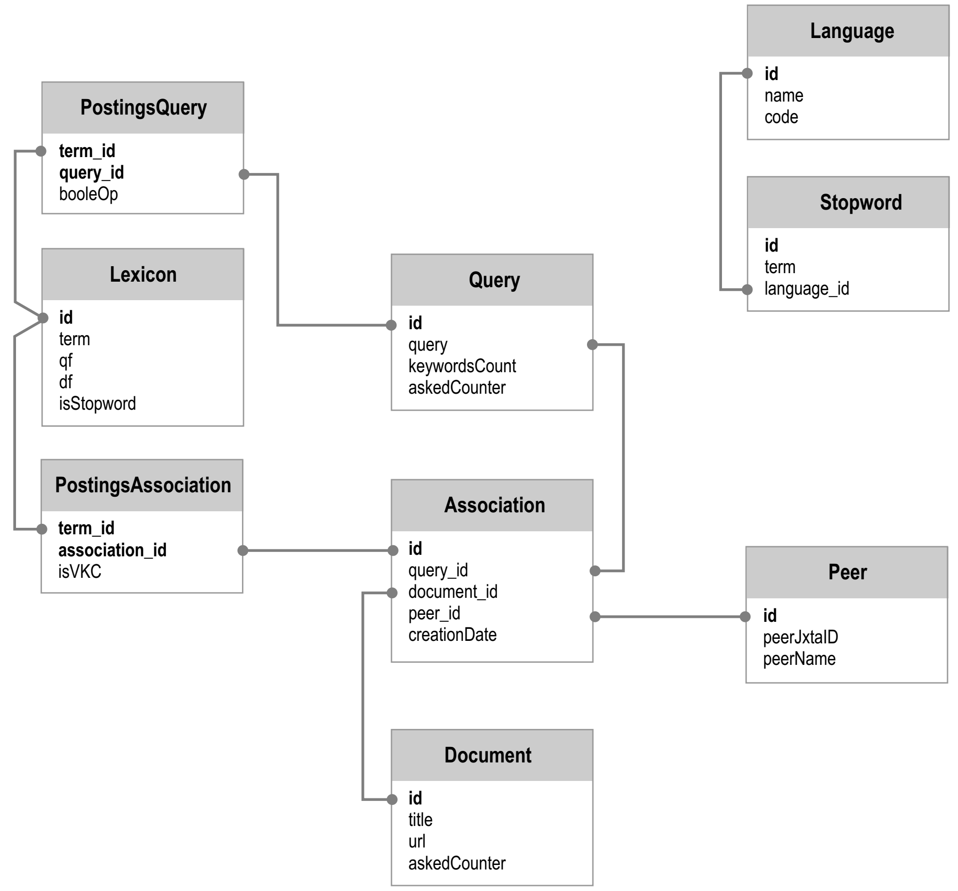


Figure 7.4: Relationship Diagram of SAQ Database according to (Grigull, 2004)

generated. The client peer whose timestamped context initiated the community formation, enables the further organization. Its ‘Community Manager’ maintains the building of a peer group in a self-organized manner.

7.2.1 Storage of Candidates

The prototype stores the candidate set in a database denoted as SAQ (Seldom-asked Queries). All tables and their relationships of the SAQ database are depicted in Figure 7.4. In analogy to the representation of explicit feedback in the Peer Search Memory, the tables `Query`, `Document`, and `Association` store the global feedback of all users. In our prototype, we use the JXTA peer ID and the JXTA peer name to identify a user, to whom an individual entry with this particular information is assigned in table `Peer`. Finally, the time on which the association was made is stored with the attribute `creationDate` in table `Association`.

7.2.2 Community Advertisement Process

The first phase of the tripartite community approach is the proposal of a community. For this task, an advertisement is generated for each community candidate. It is a community-specific

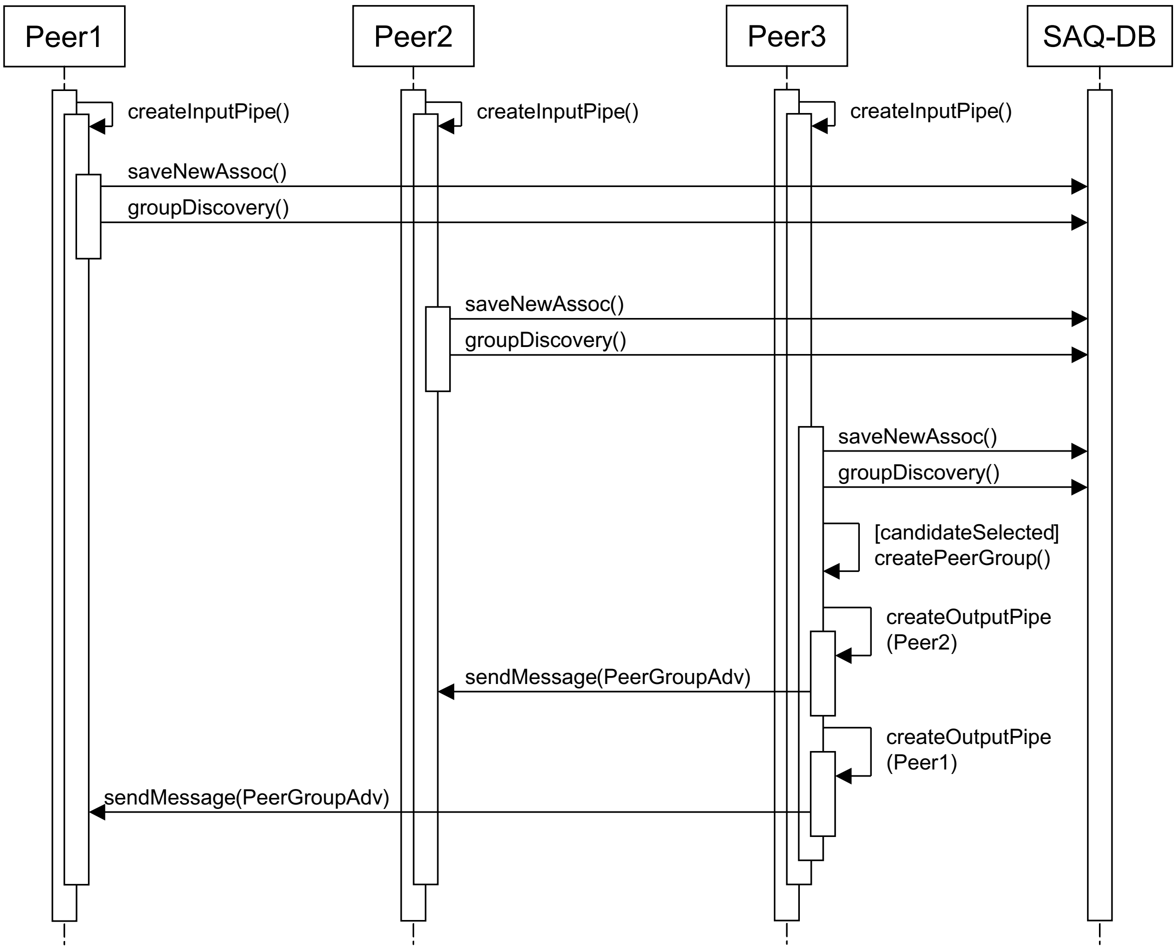


Figure 7.5: Sequence Diagram of Community Formation

summary with a set of initial associations. In the prototype, Virtual Knowledge Communities are represented through peer groups in JXTA. The members share all community-specific associations with other members of the group. In JXTA, a `PeerGroupAdvertisement` is assigned to each peer group, which publishes information about the group in the network. A number of parameters, e.g. name and description of the peer group, can be assigned to an advertisement. `PeerGroupAdvertisements` are published in order to inform other peers in the network about the existence of the peer group. This way, new groups can be discovered throughout the network. A `PeerGroupAdvertisement` is created whenever a new group is formed. The peer group name includes the main query terms of the group. We select the top 5 terms in the set of context terms that are ordered by their term weight $w(\mathcal{T}_c, t)$ (see Equation 5.4). For example, we found a community that is characterized by the terms `ebay`, `tricks`, `tipps`, `deutschland`, `markenrecht`. In addition, we select the most frequently used links of $\mathcal{A}_c$ as a description of the community. Table 7.2 shows an example listing of a `PeerGroupAdvertisement` for a VKC named `ebay`, `tricks`, `tipps`, `deutschland`, `markenrecht`. The `<Desc>`-tag of the advertisement describes all fundamental association of the community. `<GID>` specifies the `PeerGroupID` internally assigned by JXTA, which is associated with the instance of the group. `<MSID>` declares the `ModuleSpecID` that the group uses. This id is used to find a module that references the services of the group.

After an advertisement is generated, potential members are selected in the set of users from the community candidate. The notification of all peers is implemented by point-to-point messages. These messages are initialized by peers because the SAQ table has no trigger concept. Figure 7.5 visualizes an example sequence diagram of a community formation of a community. The example presents three peers instantiating an input pipe („*createInputPipe()*“) at system start. With this instantiation a pipe listener waits for messages on this pipe. Each association, which is not related to a community, updates the candidate set maintained by the SAQ database („*saveNewAssoc()*“). Each peer checks („*groupDiscovery()*“) for the actual association the attribute *isVKC* in relation `PostingsAssociation` (see Figure 7.4). In this example, a Virtual Knowledge Community is discovered after the context has been updated with the association of Peer3. This peer is now responsible for the creation of the advertisement and the selection of potential members („*createPeerGroup()*“). In addition, Peer3 initiates output pipes to the selected peers. For example, pipes are created for peer 1 „*createOutputPipe(Peer2)*“ and Peer2 „*createOutputPipe(Peer1)*“). The output pipe is used for a point-to-point message sending the peer group advertisement („*sendMessage()*“). Peer1 and Peer2 receive this message from Peer3 on their waiting input pipe. The advertisement is processed and presented to the user. Finally, the user decides whether he wants to join the group or not. A user becomes an explicit member if he confirms his membership.

Each JXTA advertisement is published with a lifetime. It specifies the availability of its associated resource. Obsolete resources can be deleted without a centralized control. If no user commits his membership to a community, the advertisement expires, and will not be available any longer. According to the peer group definition of JXTA, a group can consist of at least one member. For the announcement of a Virtual Knowledge Community, no minimal *confirmation rate* is defined in the prototype. In addition to the JXTA advertisement of a community, each peer organizes all local associations which are related to a community. For this task, the PeerSy core database is extended with two new tables as depicted in Figure 7.6. The relation `PeerGroup` stores all communities with a membership of the user. A group is represented by the group name, peer group ID, and a description. The association between query id, document id, and peer id is stored in the table `PostingsGroup`.

7.2.3 Peer Communication

The implementation of the communication within the network occurs through *Pipes*, which are provided by the JXTA framework. Pipes are virtual connections between peers, and can be used as channels between members to support file sharing. It defines an interface for receiving messages of a pipe service. At the same time, an output pipe defines an interface for sending messages of a pipe service. The main action within the pull-push cycle is a continuous searching of group members. These members must be identified during the retrieval process, as well as during the prediction of community-specific recommendations. This task is implemented by a bidirectional pipes (see (Brookshier et al., 2002)). Such a pipe has a communication channel in both directions between sender and receiver. Once an input pipe is initialized, it waits for a request to construct the pipe connection. This pipe uses the pipe

Table 7.2: PeerGroupAdvertisement for Virtual Knowledge Communities

```
 <?xml version =″ 1.0″ encoding =″ UTF − 8″? >
<!DOCTYPE jxta : PGA >
< jxta : PGA xmlns : jxta =″ http : //jxta.org″ >
   < GID >
     urn : jxta : uuid − 35DF64686B64414A9D53F58E7429363602
   < /GID >
   < MSID >
     urn : jxta : uuid − DEADBEEFDEAFBABAFEEDBABE000000010306
   < /MSID >
   < Name >
     iskodor.peersy.jxta.ebay + tricks + tipps + deutschland + markenrecht
   < /Name >
   < Desc >
      < initialAssociations >
         < query > ebay, deutschland < /query >
         < document >
           http : //www.ebay.de/
         < /document >
         < query > ebay, us, bay < /query >
         < document >
           http : //www.ebay.com/
         < /document >
         < query > ebay, crap < /query >
         < document >
           http : //beam.to/ebaycrap
         < /document >
         < query > ebay, auktion, party, frauen, vier < /query >
         < document >
              http : //www.spiegel.de/netzwelt/netzkultur/0, 1518, 273634, 00.html
         < /document >
         < query > ebay, professional, marktanteil, pc, hood < /query >
         < document >
           http : //www.die − auxburger.de/weblog/index.php?
         < /document >
      < /initialAssociations >
   < /Desc >
< /jxta : PGA >
```

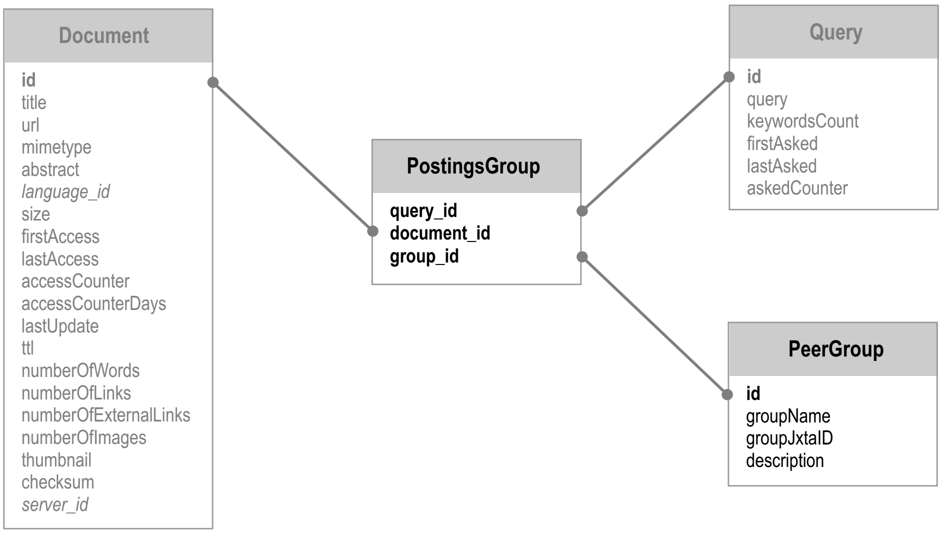


Figure 7.6: Database Extension of PeerSy Core

service for the initial connection to the pipe, and the address of the pipe endpoint is used for the reverse connection. Within the JXTA framework the bidirectional pipe is internally implemented with two unidirectional pipes. Figure 7.7 depicts a typical peer communication to gain other group members. By default, the application instantiates a bidirectional pipe, that afterwards waits for a connection request (*createBiDiPipe()*, *waitForConnection()*). The scenario in Figure 7.7 visualizes a search request of peer 1. In the following step the discovery service is used to search all groups with a membership of peer 1 and their members (*discoveryService.findPeers(ownGroup)*). For example, a connection is built to peer 2 using a bidirectional pipe. Once the connection is established, search and recommendation requests can be handled with this connection.

7.3 Community Manager

This system component is responsible for all community functionalities on a client peer. The ‘Community Manager’ is aware of all community memberships of a user. If a community-based filtering process is initiated by the trigger, this component encodes all answer and request messages of community members . This section presents both types of messages within a cooperative pull-push cycle which are initiated by the ‘Community Manager’.

7.3.1 Selection Process

Figure 7.8 depicts the iterated pull and push phases of the selection process. On a system level, the push is realized as a periodic pull. The information community manager is activated automatically by a trigger. Once the recommendation process is initiated, all possible information providers within the same community are discovered. Standard JXTA protocols are used for the communication between an active peer and a passive peer. No error handling must be considered, because JXTA implements a secure and reliable data transfer. The

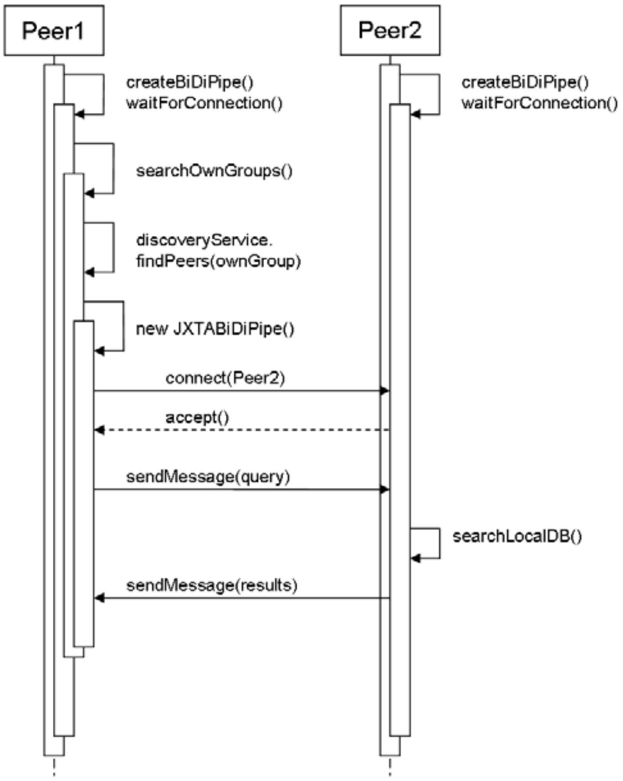


Figure 7.7: Communication of two peers

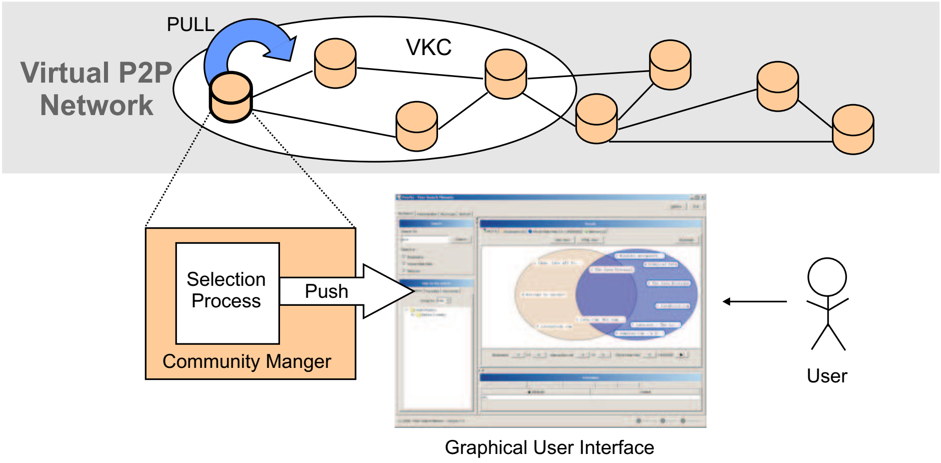


Figure 7.8: Periodic System Pull and User Push

Table 7.3: Description of a Message used by the MyPush Protocol

Message Header			
Element	**Type**	**Length (bits)**	**Description**
versionID	RF	4	MyPush protocol version
messageType	RF	3	MyPush message type (request or response)
memoLength	RF	10	length of memo
memo	RV		
connectionType	RF	4	connection type of sending peer
language	RF	16	language of message
timestamp	RF	64	time of actual message
numberPackets	RF	32	number of message packets (max. $2^{32}-1$)
Message Body			
Element	**Type**	**Length (bits)**	**Description**
packet 1	OV		according to message type
:			
packet n	OV		n = 'number'

messages are encoded according to a specific MyPush protocol (Gül, 2004) and the JXTA Pipe Binding Protocol performs a message broadcast. The MyPush protocol is defined on the application layer of the ISKODOR system architecture (see Section 3.2.2). Each message is binary encoded in order to decrease the network load and for security reasons. Two basic message types are distinguished in the MyPush protocol:

- `MyPushRequest` ($messageType = 0$) and
- `MyPushResponse` ($messageType = 1$).

Both types of messages have a common header that is succeeded by the specific message content. For the description of a message, special abbreviations are used for each element type:

- **R** required element,
- **O** optional element,
- **F** fix length of element or
- **V** variable length of element.

A message according to the MyPush protocol consists of a header and a body. The general description of a message is summarized in Table 7.3. All elements of the message header are independent of the particular message type. At first, the header of a message specifies the version of the used protocol. Furthermore, the message header describes whether this

Table 7.4: Connection Types

ct	Network Type
1	LAN T3
2	LAN T1
3	WLAN (Wireless LAN)
5	Bluetooth
10	Modem

message is a response or a request. A response is sent from a passive peer to an active peer. Each peer can add a short memo to each message. Elements of the message header such as *language*, *timestamp*, and *connectionType* are responsible for a local selection of elements at each peer. The message body consists of several packets of the same type. A packet type distinguishes between a request or a response.

7.3.2 Request Message

The periodic pulling starts with an automatic generation of a request by the active peer. This request encodes meta-information in order to restrict the set of recommended documents:

1. Selection of *languages* for recommended results.
2. Selection of *query terms* describing the information need.
3. Identification of the *connection type* of the active peer.
4. Specification of a *timestamp* to avoid already known results.

The first criteria restricts the set of recommended documents to all assisted languages. See Section 7.1.2 for further details which languages are assisted, and how they are represented in the database. In a second step, the active peer selects a set of terms that are associated with a community. These terms characterize the specific information needs which led to a particular community membership. Each query term t of an active peer associated with a community c is ranked by the weight $w_{t,c}$ with $t \in \mathcal{T}_c$:

$$w_{t,c} = \frac{qf_{t,c}}{|\mathcal{T}_c|} \tag{7.3}$$

$qf_{t,c}$ is the community-specific query frequency. The ranked list of all computed query terms is named $TSorted$.

The maximal number of query terms that are sent to all passive peers are limited by a threshold. This threshold is introduced to optimize the network load. It is computed by means of a connection type representing different types of networks. For this task, we use five categories

Table 7.5: Description of a MyPushRequest Packet

n. MyPushRequest Packet			
Element	**Type**	**Length (bits)**	**Description**
id	RF	32	number n ($1 \leq n \leq 2^{32}$)
termLength	RF	10	number of term tokens
term	RV		

presented in Table 7.4. The connection type influences the transfer rate in the peer-to-peer network. The number of selected query terms is limited by the value $selTerms$:

$$selTerms = \lfloor |TSorted| * \frac{1}{(ct)} \rfloor \tag{7.4}$$

$|TSorted|$ quantifies the total number of ranked query terms. ct defines the connection type as listed in Table 7.4. Finally, the timestamp is selected of the last request in order to avoid already known documents.

For information gathering, all selected terms are used to build a query automatically. The number of terms is specified by the element *numberPackets* of the message header. For each query term, a packet is generated and added to the message body. Each packet of a request consists of several elements listed in Table 7.5. The element *id* is incrementally increased with each term added to the message body. The maximum length of a term is 2^{10} tokens. All packets of a request compose a set of query terms. Each request is sent out to all available passive peers of the community. All peers in the same group with the active peer take over the further processing to answer the request.

7.3.3 Answer Message

In terms of an efficient processing of a request, the service repository of a passive peer is already initialized, and it stores in the tables `ResultAssociation` and `ProfileResult` all pre-selected documents and their implicit assessment. Information is composed at each passive peer according to four restrictions:

1. All results in the `ProfileResult` table are selected, which conform to the selected language of the active peer.
2. From this set of possible recommendations, all results are considered with a more recent timestamp than the active peer.
3. All results are retrieved that match the requested query terms. These hits are ranked by the attribute *assessment*. The ranked output is a set called $RSorted$.
4. The maximal number of recommendations is selected by considering the connection type.

Table 7.6: Description of a MyPushResponse Packet

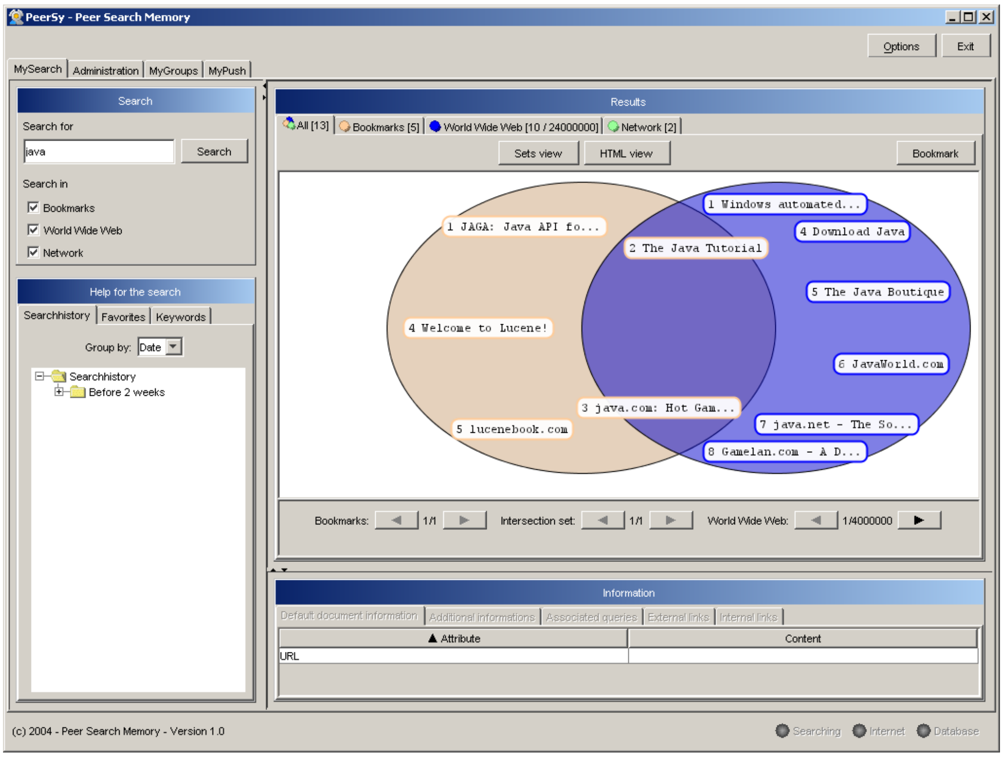
n. MyPushResponse Packet			
Element	**Type**	**Length (bits)**	**Description**
id	RF	32	number n ($1 \leq n \leq 2^{32}$)
titleLength	RF	10	number of tokens in title
title	RV		
urlLength	RF	10	number 'URL' tokens
url	RV		
mimetypeLength	RF	10	number of 'Mimetype' tokens
mimetype	RV		
abstractLength	RF	10	number of abstract tokens
abstract	RV		
date	RF	64	date of first access
assessment	RF	32	result assessment $\cdot 10^7$
numberTerms	RF	32	number of term packets (max. $2^{32} - 1$)
MyPushRequest Packet 1	OV		
:			
MyPushRequest Packet n	OV		n = 'numberTerms'

To compute the number of recommendations of $RSorted$, the communication type is set to the maximum of both values, $selConn = max(AP, PP)$. AP is the connection type of the active peer, and PP is the connection type of the passive peer (see Table 7.4). Analogous to the computation of selected terms (see Equation 7.4), the number of recommendations $selResults$ is computed by

$$selResults = \lfloor |RSorted| * \frac{1}{selConn} \rfloor \tag{7.5}$$

with $|RSorted|$ the total number of ranked results. The number of recommended results of a passive peer is limited by this threshold. In addition, meta-information according to the result weighting is sent to the active peer. All recommended documents are represented by a response packet. The specification of the result is described in Table 7.6.

Each packet gets an identification number which is incrementally assigned. A recommended document is characterized by the elements *title, url, minetype, abstract, date*, and *assessment*. All values are extracted from the PeerSy core. In addition to each document, all query terms associated with this document are attached. These terms are represented by a `MyPushRequest` packet. The attachment of query terms is necessary for the active peer to organize all responses. Recommended documents are stored in the table `ProfileResult` of the service repository. The attribute *isExtern* is set to 1 in order to indicate that no local storage of the document in the PeerSy core exists. This document is only temporary stored in the table `RepDocument` until the user gives his explicit feedback. The relevance for all recommended documents is predicted before a push is initiated for the user. This process includes that already known documents and duplicates are automatically removed.



Search for: java

☑ Bookmarks
☑ World Wide Web
☑ Network

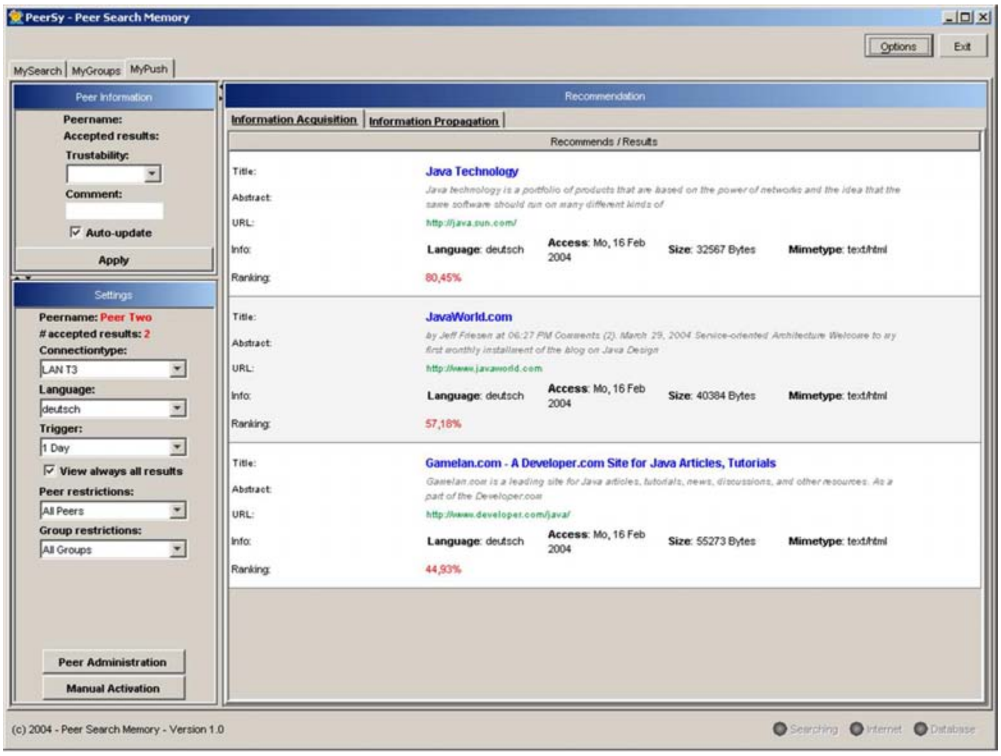
Group by: Date

Figure 7.9: User Interface of MYSEARCH according to (Ruhl, 2003)

7.4 Graphical User Interface

Two main constraints have been defined for the design of the user interface: platform independence and browser independence. In order to implement these constraints, we chose an architecture with a WBI[2] proxy. During the design phase, special attention was paid to the ISO norm 9241-10. These principles describe the design of graphical user interfaces, and a user friendly operation of a system. Figure 7.9 depicts a screenshot of the actual prototype. The user interface is divided into four tabs: ‘MySearch’, ‘MyGroups’, and ‘MyPush’, and ‘Administration’. The ‘MySearch’ tab assists actual information needs. The search interface is designed analogous to common Web search engines. On the left side, a user can formulate his query. According to the users information needs, he can choose different result presentations. We chose a set-view, in order to differentiate between already known results, as well as new results. The orange color marks all known results, which are stored in the Peer Search Memory. They are ranked by the personalized ranking strategy. All blue colored results are new results from external and internal information sources. They are merged for a personalized ranking. If the user finds a new relevant result, he can bookmark this result with the interface or with an additional button in his browser. In addition to the set-view presentation, the interface presents all results according to their source in single tabs which organize results into ‘Bookmarks’, ‘World Wide Web’, and ‘Network’. This differentiation of search results allows the user to select a specific information source. For example, if he wants to find a former relevant document for his information need, he can narrow the set of results to the local information collection. In addition to the search result, additional

[2] http://www.almaden.ibm.com/cs/wbi/index.html, last visit on 2005/08/30.



Peername:
Accepted results:
Trustability:
Comment:
☑ Auto-update

Peername: Peer Two
# accepted results: 2
Connectiontype: LAN T3
Language: deutsch
Trigger: 1 Day
☑ View always all results
Peer restrictions: All Peers
Group restrictions: All Groups

Title:	Java Technology
Abstract:	Java technology is a portfolio of products that are based on the power of networks and the idea that the same software should run on many different kinds of
URL:	http://java.sun.com/
Info:	**Language**: deutsch **Access**: Mo, 16 Feb 2004 **Size**: 32567 Bytes **Mimetype**: text/html
Ranking:	80,45%
Title:	JavaWorld.com
Abstract:	by Jeff Friesen at 06:27 PM Comments (2) March 29, 2004 Service-oriented Architecture Welcome to my first monthly installment of the blog on Java Design
URL:	http://www.javaworld.com
Info:	**Language**: deutsch **Access**: Mo, 16 Feb 2004 **Size**: 40384 Bytes **Mimetype**: text/html
Ranking:	57,18%
Title:	Gamelan.com - A Developer.com Site for Java Articles, Tutorials
Abstract:	Gamelan.com is a leading site for Java articles, tutorials, news, discussions, and other resources. As a part of the Developer.com
URL:	http://www.developer.com/java/
Info:	**Language**: deutsch **Access**: Mo, 16 Feb 2004 **Size**: 55273 Bytes **Mimetype**: text/html
Ranking:	44,93%

(c) 2004 - Peer Search Memory - Version 1.0

Figure 7.10: User Interface of MyPush according to (Gül, 2004)

statistical information of previous searches and favorites is presented.

Besides an active information need, the user interface assists administrative features and an information push. In ‘MyGroups’ existing Virtual Knowledge Communities are summarized, and individual peer membership for a user are administrated. Figure 7.10 shows the ‘MyPush’ tab that organizes recommendations by presenting consumed and provided information. It is always transparent for a user which documents have been recommended to others. All recommended documents are ranked, and the user can provide feedback for a new relevant document that has been recommended. His Peer Search Memory is updated with this document, and all recommenders’ peer relevance is increased. Finally, the ‘Administration’ tab offers the possibility to manage all query-document associations. Associations can be deleted by the user if the relevance of a document to a query is not given any more. In the actual prototype, this deletion of an association is not automatically propagated to a community. The deleted association still belongs to the community context. Future work is necessary to enhance the community administration.

7.5 Summary

This chapter described the prototype implementation of the Congenial Web Search concept. It implements the architecture and the techniques described in the chapters 3 to 6. In general, we tried to employ open-source components for common tasks, in order to reduce development time. ISKODOR is a prototype that can be installed on several client peers. All components are implemented in the Java programming language, using the J2SE 1.4.2 SDK. The

JXTA framework ensures a high-scalability due to efficient routing algorithms for a widely used decentralized application. An open-source database is used to maintain a local index of all prior query-document associations. In addition, it maintains a repository of all pre-selected recommendations for other users and of all recommendations received from other community members. The prototype implements a data access to one external information provider, in our case the Google Web service. For the communication of peers, we implemented JXTA protocols to discover each other, advertise and discover network resources, and communication and route messages. The JXTA peer group concept has been used that peers self-organize into Virtual Knowledge Communities. For the periodic pull within a peer group, two new protocols to request for recommendations and to send recommendations are defined. The main component is the graphical user interface of the Peer Search Memory. It ensures a transparent exchange with all information collections, and it assists an easy selection of a specific information collection that is relevant for the users information need.

8 Conclusion

This dissertation advanced the state of the art in Web information retrieval. To improve the effectiveness of Web search, a new search paradigm is proposed based on local associations between queries and relevant results. The concept is motivated by three conceptual pillars, which are: personalization, collaboration, and socialization. Personalizing search engines promises to be a way to optimize retrieval within huge data sets, taking into account individual interests and preferences. To do so, the first step is the storing of associations between queries and results with respect to a personalized ranking. The individual storage of a search history was shown to be more effective than a non-personalized Web search engine for repeated queries. In a second step, local search transactions are merged in order to group queries and relevant results. A novel technique for community discovery and control was developed for a decentralized search environment. The new technique analyzes individual search contexts in a transparent manner in order to group common interests. Thus, the optimization is not only done for actual trends requested by the majority of searchers. Also small interest groups can be found with this method. The community approach was shown to have a high quality of similar terms and documents. All tests used real usage data to verify the search behavior that led to a discovery of effective Virtual Knowledge Communities. The overall goal of this dissertation is to bring people together with shared interests. This process is directed by query terms and links. Congenial Web Search thoroughly addresses the integration of the information retrieval process, and the information filtering process. All types of interests, long-term as well as short-term, are assisted by integrated information seeking. The new concept of integrated information retrieval was shown to improve the effectiveness of known-item retrieval for experts and novice searchers for a small set of queries. In addition, the effectiveness of the community-based filtering approach was explored with distributed information sources. All techniques are implemented in a prototype denoted as ISKODOR. The remainder of this chapter discusses its limitations, its future work, and its impact.

8.1 Limitations

The main limitation of Congenial Web Search is the need for explicit feedback. Users are not always willing to provide feedback for a document. In addition, the need for explicit feedback also limited the evaluation setting. Integrated information seeking processes show their real impact only in a real interaction scenario. Without active contributions of users, a prototype for Congenial Web Search is only a personalized meta-search engine. Furthermore, the framework is limited in the following ways:

- The scalability of the approach is not analyzed for large and diverse user graphs, or for extended time. The hybrid design of the prototype utilizes a central entity to synchronize all user feedback. The implementation as a centralized service is intuitive, but breaks the original notion of the peer-to-peer paradigm to abandon any kind of central server. This central server can become the bottleneck of the system if the number of users rapidly increases.
- Each peer is autonomous in its decision, to what time and to what extent data is to be shared with the environment. The system functionalities are limited if a peer is not available for participation in an integrated information seeking process. The limitation depends mainly on the collected information of the peer. If a peer profile represents mainstream interests, similar judgements can be found on many other peers. Otherwise, if a peer collected very specific information, no replication of the associations will be found on other peers.
- The discovery of contexts is based on the similarity of query-document associations. No semantic term similarities within and across languages is considered during this process. The update process of a community does not consider temporal correlation of queries or synonyms. Communities group only terms within the same language. In addition, no detection of synonymous links or duplicates within and across languages is applied to the prototype. The summary of all Peer Search Memories builds a multilingual information collection. No assistance for cross-language retrieval exists in the prototype.
- All evaluation settings does not consider statistical significance tests. The significance of the improvements of the retrieval and filtering effectiveness must be validated in combination with a parameter optimization. The parameters for community discovery are not optimized with a training and test corpus.
- Once a community is initialized, its growth depends on all its users. The community expands with highly active members. From the time on a user commits his membership, we expect that the information need assigned to a community either engrosses, remains constant, or fades. The community approach has no administration functions in order to assist an advanced community management. For an established community structure, no support for splitting or joining communities is implemented.

8.2 Future Work

The most important contribution of Congenial Web Search is the new perspective people gain, while they assist each other. The first prototype shows how all concepts can be integrated. It is a first solution which promises more research contributions for future work:

- For all users, the Peer Search Memory is an individual archive of former search interests. They are dynamically updated, and once the user formulates an information need, he accesses a static information collection at the time of his request. Besides individual sources, communities are advanced information sources with an update and expansion history. The lifetime of a community starts with the first membership of a user. The self-organization paradigm of the community assigns each peer group's advertisement a time-to-live that specifies the availability of its associated resource. This lifetime facilitates the deletion of obsolete resources without any central control. If the advertisement of a community is not republished, no interest in the topic is shown by any of its users. Future work has to evaluate an appropriate default lifetime for a community. Because of the self-organization of the peer group, it must be possible to manage each community individually. In case a community lifetime is not extended, it should be added to an archive of former communities. Archival communities can refer to a former state of knowledge of a user group. Once the interest fades by the members, the topic might be useful for a new group of users in the future. Hence, we propose a new retrieval system for an archive of communities.

- The framework of Congenial Web Search assumes that the community structure evolves from all participants and their information needs. With a small number of initial users the evolution of the community structure is rather slow. Future work has to concentrate on the dependency of the number of users, and the effectiveness of the system. In this regard, we have to evaluate whether a small group with similar interests can create a community structure which is more effective than one created by a larger group with heterogenous interests. The generic approach of Congenial Web Search is primarily developed for a large-scale service similar to Web search engines. For smaller and specific groups, an import of communities would lead to a new research direction. In particular, if all users are known in advance, it is possible to incorporate specific information sources. For example, a business network offers specific information about employees and their cooperation in former projects. First experiments showed that an intranet of a company provides usage data and click-through data which can be used to identify search tactics (Gnasa and Harbusch, 2002). After a first automatic discovery, a manual verification step can lead to an import of a preprocessed community structure that avoids a cold-start problem. In addition, for a restricted set of users, a manual administration of relationships promises an effective knowledge management.

- In spite of the general interest in social software, we noticed a trend towards incorporating soft memberships. The concept of community discovery only reflects whether people belong to a group, or not. Enhancing this concept means adding a factor that takes into account how much interest a member has in a topic. Soft memberships can be derived from implicit contributions to a community which can be collected by observing specific user actions. For example, a user has gotten a membership offer, but hesitates to accept the relationship, or a user collects related queries and related documents which do not exactly match with a community topic. For both observations, new similarity measures must be explored for a multi-level ranking that also incorporates closeness among different members of one community. Soft relationships define a new dimension, which must reflect the natural habit that one would pay more attention to a rating by a close friend than that of a foreigner to the community.

- With a community, a set of links is collected which are assessed by community members. These links are a useful source of a community expansion, and to utilize a community-based hypertext structure. A new focused crawling approach can be implemented that is initiated from each community. In addition, a new link similarity can be defined if a community-specific vocabulary and temporal constraints are incorporated. All terms associated with links of the community can be used to define a community-specific vocabulary. It can be expanded with new terms which are temporally correlated. Chien and Immorlica (2005) developed a semantic similarity of query terms using temporal correlation. In analogy to this concept, all timestamped contexts can be exploited in order to find temporally correlated terms and links. In comparison to other link analyses (e.g. Page Rank, HITS), a hybrid approach will combine content-based and link-based measures if the vocabulary similarity between two sites exceeds a threshold. Each community can initiate the link analysis individually, and the peer-to-peer architecture harnesses the computing power of the peers composing the community. It is necessary to evaluate a default threshold and the convergence behavior of the algorithm.

- During the indexing of the test corpus, we noticed a link rot rate of 17%. This observation shows the rapid change of the Web content and its transience. Because of link rot there will be a demand for future expansion of this work. First of all, the influence of link rot on the effectiveness of integrated information seeking must be evaluated. The dynamics of the Web is already a challenge, in order to build test collections with relevance judgments that include documents which are no longer contained in the collection. There are several reasons why a page is no longer contained in the collection. With respect to archival communities, the impact of caching important documents is explored. Two strategies can be attempted in order to deal with link rot. On one hand, a local index caches documents. In this regard, only the textual content of the document needs to be recovered. On the other hand, each local Peer Search Memory initiates a process that crawls already stored documents to detect unavailable links. A similar process can be done with respect to associated community links. Once a temporary unavailable link is detected, this link must be flagged in the search result list. A representative user study needs to evaluate which strategies might be preferred by the users.

- A limitation of Congenial Web Search is the response time of the decentralized architecture. We see the need of a critical mass analysis due to scalability and usability reasons. For small user groups, we noticed the need of imported communities in order to allow users an effective search from the beginning. For a large user group, a critical mass analysis is necessary to evaluate the limitations of the system. Besides technical limitations leading to low response times, the heterogeneity of all users' interests can result in an unmanageable graph structure. Surveys on the Web structure showed that the Web fulfils the conditions of a small world graph. First analyses showed that a simulated social network extracted from a mailing-list archive possesses this characteristics (Kirsch, 2005). Once the characteristics of the full network structure are determined, future work will derive new navigational search strategies for a reputation-based retrieval. A graph structure of all community memberships is necessary to assign a reputation to all of its users.

- The user-centered design of Congenial Web Search is a usefull source for ethnographical studies. Search engines provide only snapshots of the actual user interests. Instead, Peer Search Memories, as well as each community, tracks the interests of all users with more details. In combination with archival communities, it is possible to explore the evolution of the internet society. In particular, former trends can be reconsidered. Moreover, ethnography promises to explore more theoretical results of a long-term search behavior analysis. Congenial Web Search is a platform which offers insights into the future development of the Web and its users searching for information.

- The local user profile on all peers enables a distributed Web search personalization. If a user's information need can not be satisfied with the local information source, his profile is used for a search in external and internal information providers. The approach of (Teevan et al., 2005b) can be integrated to personalize Web search for new information needs. This work focused on a re-ranking of the top search results locally. In future work, such a re-ranking of results can be integrated in our approach. In addition, the set of internal information sources is processed in order to collect validated results from other users. A re-ranking of their results is easier to implement, because the data access service of each peer can provide all necessary document statistics which are necessary to calculate a personalized document score.
- A final aspect for future work is the incorporation of user-centered evaluation measures. For such a setting, we cannot apply system-centered evaluation measures like precision and recall. Congenial Web Search offers optimal user satisfaction, which cannot be measured among all users. Each user has individual needs. A study by Teevan et al. (2005a) showed that people are not good at specifying detailed informational goals. For a user-centered evaluation measure, we need information about the searcher that can be collected in an automated manner. Informational goals and their satisfaction can be inferred implicitly by exploring the user. In a naive way, each user's information need is satisfied when he received relevant information in an adequate search time. The relevance of information and the adequateness of the time that has been spent for search are both subjective ratings of a user. In addition, both factors depend on the type of information need. An unsupervised learning strategy can be developed which learns to classify information needs from the number of documents a user has viewed until he found a relevant document.

8.3 Impact

The Web is extremely vast and heterogeneous with respect to content, structure, and quality. This leads to great difficulty in retrieving documents, and measuring the Web search effectiveness. Traditional Web search engines are popular, even though they may not have optimal effectiveness. Web retrieval is optimized for the processing of several thousand queries per second. New system architectures should not claim to exceed existing systems in their coverage of the Web, and their performance. From the user's perspective, efficiency is part of his overall subjective impression of a Web search engine. At present, the success of a application depends on its ability to address the user's specific information need. This can be done if the users are encouraged to participate by explicitly adding value to the application. Personalization techniques and community discovery are new services which incorporate interaction-enabling technologies. An open application has been developed in this dissertation that serves as a solid basis for future research on harnessing collective intelligence.

Bibliography

Adler, L. M. A modification of kendall's tau for the case of arbitrary ties in both rankings. *Journal of the American Statistical Society*, 52:33–35, 1957.

Aggarwal, C. C., J. L. Wolf, K.-L. Wu, and P. S. Yu. Horting hatches an egg: a new graph-theoretic approach to collaborative filtering. In *Proc. of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, 1999.

Allan, J., C. Wade, and A. Bolivar. Retrieval and novelty detection at the sentence level. In *Proc. of the 26th Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2003)*, 2003.

Alton-Scheidl, R., J. Ekhall, O. van Geloven, L. Kovacs, A. Micsik, C. Lueg, R. Messnarz, D. Nichols, J. Palme, T. Tholerus, D. Mason, R. Proctor, E. Stupazzini, M. Vassali, and R. Wheeler. Select: Social and collaborative filtering of web documents and news. In *Proceedings of the 5th ERCIM Workshop on User Interfaces for All: User-Tailored Information Environments*, 1999.

Amitay, E., D. Carmel, M. Herscovici, R. Lempel, and A. Soffer. Trend detection through temporal link analysis. *Journal of the American Society for Information Science and Technology*, 55(14):1270–1281, 2004.

Arberer, K., F. Klemm, M. Rajman, and J. Wu. An architecture for peer-to-peer information retrieval. In *SIGIR 2004 Workshop on Peer-to-Peer Information Retrieval*, 2004.

Aslam, J. A. and M. Montague. Models for metasearch. In *Proc. of the 24th Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2001)*, 2001.

Baeza-Yates, R. and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, Reading, MA, USA, 1999.

Baeza-Yates, R. and J. A. Pino. A first step to formally evaluate collaborative work. In *Proc. of the international ACM SIGGROUP conference on Supporting group work : the integration challenge*, 1997.

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

Balke, W.-T. *Supporting Information Retrieval in Peer-to-Peer Systems*. Springer-Verlag, Berlin, Germany, 2005.

Barabási, A.-L. and R. Albert. Emergence of scaling in random networks. *Science*, 286: 509–512, 1999.

Bates, M. After the dot-bomb: Getting web information retrieval right this time. *First Monday*, 7(7), 2002.

Baudisch, P. *Dynamic Information Filtering*. PhD thesis, GMD Forschungszentrum Informationstechnik GmbH, Sankt Augustin, 2001.

Bawa, M., R. J. Bayardo, Jr., S. Rajagopalan, and E. J. Shekita. Make it fresh, make it quick: searching a network of personal webservers. In *Proceedings of the 12th international conference on World Wide Web*, 2003.

Belkin, N. J. Information filtering and information retrieval: two sides of the same coin? *Social Science Information Studies*, 4(2 & 3):111–129, 1984.

Belkin, N. J., C. Cool, A. Stein, and U. Thiel. Cases, scripts and information seeking strategies: on the design of interactive information retrieval systems. *Expert Systems with Applications*, 9(3):379–395, 1995.

Belkin, N. J. and W. B. Croft. *Retrieval Techniques*. Elsevier Science Publishers B.V. North-Holland, 1987.

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

Bender, M., S. Michel, P. Triantafillou, G. Weikum, and C. Zimmer. Improving collection selection with overlap awareness in p2p search engines. In *Proc. of the 28th Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2005)*, 2005.

Bender, M., S. Michel, G. Weikum, and C. Zimmer. Bookmark-driven query routing in peer-to-peer web search. In *SIGIR 2004 Workshop on Peer-to-Peer Information Retrieval*, 2004.

Berners-Lee, T. Information management: A proposal. Technical report, CERN Internal Communication, 1989.

Berry, M., Z. Drmac, and E. R. Jessup. Matrices, vector spaces, and information retrieval. *SIAM Review*, 41(2):335–362, 1999.

Berry, M., S. Dumais, and G. O'Brien. Using linear algebra for intelligent information retrieval. *SIAM Review*, 37(4):573–595, 1992.

Bharat, K., B.-W. Chang, M. R. Henzinger, and M. Ruhl. Who links to whom: Mining linkage between web sites. In *Proc. of the 2001 IEEE International Conference on Data Mining (ICDM '01)*, 2001.

Bharat, K. and M. R. Henzinger. Improved algorithms for topic distillation in a hyperlinked environment. In *Proc. of the 24th Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2001)*, 2001.

Bloom, B. H. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426, 1970.

Booch, G., J. Rumbaugh, and I. Jacobsen. *The Unified Modeling Language User Guide*. Addison-Wesley, Reading, MA, USA, 2005.

Bookstein, A. Probability and fuzzy-set applications to ir. *Annual Review of Information Science and Technology*, 20:117–151, 1985.

Breese, J. S., D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proc. of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence*, 1998.

Brin, S. and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Proc. of the 7th International World Wide Web Conference (WWW 1998)*, 1998.

Brookes, B. Theory of the bradford law. *Journal of Documentation*, 33(3):180–209, 1977.

Brookshier, D., D. Govoni, and N. Krishnan. *JXTA: Java P2P Programming*. SAMS, Indianapolis, IN, United States, 2002.

Bruegge, B. and A. H. Dutoit, editors. *Object-Oriented Software Engineering*. Prentice Hall, Englewood Cliffs, NJ, 2004.

Buckley, C. and G. Salton. Optimization of relevance feedback weights. In *Proc. of the 18th Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1995)*, 1995.

Bush, V. As we may think. *The Atlantic Monthly*, 176(1):101–108, July 1945.

Byström, K. and K. Järvelin. Task complexity affects information seeking and use. *Information Processing & Management*, 31(2):191–213, 1995.

Callan, J. *Distributed information retrieval*. Kluwer Academic Publishers, Dordrecht, Norwell, 2000.

Callan, J. Document filtering with inference networks. In *Proc. of the 19th Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1996)*, 1996.

Callan, J. Learning while filtering documents. In *Proc. of the 21th Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1998)*, 1998.

Callan, J., Z. Lu, and W. B. Croft. Searching distributed collections with inference networks. In *Proc. of the 18th Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1995)*, 1995.

Chakrabarti, S., M. M. Joshi, K. Punera, and D. M. Pennock. The structure of broad topics on the web. In *Proc. of the 11th International World Wide Web Conference (WWW 2002)*, 2002.

Chakrabarti, S., S. Srivastava, M. Subramanyam, and M. Tiwari. Memex: A browsing assistant for collaborative archiving and mining of surf trails. In *Proceedings of the 26th International Conference on Very Large Data Bases*, pages 603–606, 2000.

Chawathe, Y., S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker. Making gnutella-like p2p systems scalable. In *SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, 2003.

Chien, S. and N. Immorlica. Semantic similarity between search engine queries using temporal correlation. In *Proc. of the 14th International World Wide Web Conference (WWW 2005)*, 2005.

Cooper, W. S. *Getting beyond Boole*. Morgan Kaufmann, Los Angeles, CA, USA, 1988.

Crespo, A. and H. Garcia-Molina. Routing indices for peer-to-peer systems. In *ICDCS '02: Proceedings of the 22 nd International Conference on Distributed Computing Systems (ICDCS'02)*, Washington, DC, USA, 2002. IEEE Computer Society.

Cöster, R. and M. Svensson. Inverted file search algorithms for collaborative filtering. In *Proc. of the 25th Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2002)*, 2002.

Cuenca-Acuna, F. M. and T. D. Nguyen. Text-based content search and retrieval in ad hoc p2p communities. Technical report, Technical Report DCS-TR-483, Rutgers University, 2002.

De Meer, H. and C. Koppen. *Self-Organization of Peer-to-Peer Systems*. Springer-Verlag, Berlin, Germany, 2005.

Dean, J. and M. R. Henzinger. Finding related pages in the world wide web. In *Proc. of the 8th International World Wide Web Conference (WWW 1999)*, 1999.

Deerwester, S., S. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic indexing. *Journal of the American Society for Information Science*, 41(6): 321–407, 1990.

Denning, P. Electronic junk. *Communications of the ACM*, 25(3):163–165, 1982.

Dieberger, A. and M. Guzdial. *CoWeb - Experiences with Collaborative Web Spaces*. Springer Verlag, Berlin, 2003.

Dumais, S. Improving the retrieval of information from external sources. *Behavior Research Methods, Instruments & Computers*, 23(2):229–236, 1991.

Eberspächer, J. and R. Schollmeier. *First and Second Generation of Peer-to-Peer Systems*. Springer-Verlag, Berlin, Germany, 2005.

Eco, U. Umberto eco über elektronische medien und alphabetisierungskurse. *DIE ZEIT*, 30, 1996.

Eirinaki, M. and M. Vazirgiannis. Web mining for web personalization. *ACM Transactions on Internet Technology*, 3(1):1–27, 2003.

Eisenberg, M. and C. Barry. Order effects: a study of the possible influence of presentation order on user judgements of document relevance. *Journal of the American Society for Information Science*, 39(5):293–300, 1988.

Ellis, D. A behavioral approach to information retrieval design. *Journal of Documentation*, 46(3):318–338, 1989.

Engelbart, D. C. Augmenting human intellect: A conceptual framework. Technical report, Summary Report, No. AFOSR 3233, Contact AF49(638)-1024, Stanford Research Institute, 1962.

Engelbart, D. C. and W. K. English. A research center for augmenting human intellect. In *Proceedings of Fall Joint Research Conference (IFJC)*, pages 395–410. AFIPS Press, 1968.

Erdős, P. and A. Rényi. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci.*, 5:17–61, 1960.

Faloutsos, C. Access methods for text. *Computing Surveys*, 17(1):49–74, 1985.

Faloutsos, C. and D. W. Oard. A survey of information retrieval and filtering methods. Technical report, CS-TR-3514, Univ. of Maryland Institute for Advanced Computer Studies Report, 1995.

Fidel, R., H. Bruce, A. M. Pejtersen, S. Dumais, J. Grudin, and S. Poltrock. Collaborative information retrieval. *The New Review of Information Behaviour Research*, 1(1):235–247, 2000.

Fidel, R., A. M. Pejtersen, B. Cleal, and H. Bruce. A multidimensional approach to study of human-information interaction: a case study of collaborative information retrieval. *Journal of the American Society for Information Science and Technology*, 35(3):66–71, 2004.

Fischer, G. and C. Stevens. Information access in complex, poorly structured information spaces. In *Proc. of the 1991 ACM Conference on Human Factors in Computer Systems*, pages 63–70. ACM Press, 1991.

Fisher, D. *Studying Social Information Spaces*. Springer Verlag, Berlin, 2003.

Flake, G., S. Lawrence, and C. L. Giles. Efficient identification of web communities. In *Proc. of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 150–160, 2000.

Flake, G. W., S. Lawrence, C. L. Giles, and F. Coetzee. Self-organization of the web and identification of communities. *IEEE Computer*, 35(3):66–71, 2002.

Flake, G. W., K. Tsioutsiouliklis, and L. Zhukov. *Methods for Mining Web Communities: Bibliometric, Spectral, and Flow*. Springer Verlag, Berlin, 2004.

Florance, V. and G. Marchionini. Information processing in the context of medical care. In *Proc. of the 18th Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1995)*, 1995.

Foltz, P. W. and S. Dumais. Personalized information delivery: an analysis of information filtering methods. *Communications of the ACM*, 35(12):51–60, 1992.

Frakes, W. and R. Baeza-Yates. *Information Retrieval – Data Structures Algorithms*. Prentice Hall Inc., Upper Saddle River, 1992.

Frants, V., J. Shapiro, I. Taksa, and V. Voiskunskii. Boolean search: Current state and perspective. *Journal of the American Society for Information Science*, 50(1):86–95, 1999.

Gibson, D., J. Kleinberg, and P. Raghavan. Inferring web communities from link topology. In *Proc. of the 9th ACM Conference on Hypertext and Hypermedia*, 1998.

Gül, N. MyPush - Ein kollaborativer Push Dienst für die automatische Informationsbeschaffung in einem Peer-to-Peer Netzwerk. Diploma thesis, Institute of Computer Science III, Rheinische Friedrich-Wilhelms-Universität Bonn, 2004.

Gnasa, M., S. Alda, J. Grigull, and A. B. Cremers. Towards virtual knowledge communities in peer-to-peer networks. In *SIGIR 2003 Workshop on Distributed Information Retrieval*. LNCS, 2003.

Gnasa, M., S. Alda, N. Gül, J. Grigull, and A. B. Cremers. Cooperative pull-push cycle for searching a hybrid p2p network. In *Proceedings of the 4th IEEE International Conference on Peer-to-Peer Computing*, 2004a.

Gnasa, M. and K. Harbusch. Evaluation of search tactics of it-professionals in the framework of a boolean retrieval model. Technical report, University Koblenz-Landau, Computer Science Department, 2002.

Gnasa, M., M. Won, and A. B. Cremers. Three pillars for congenial web searching - continuous evaluation for enhancing web search effectiveness. *Journal of Web Engineering*, 3 (3&4):252–280, 2004b.

Goldberg, D., D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70, 1992.

Goldberg, K., T. Roeder, D. Gupta, and C. Perkins. Eigentaste: A constant time collaborative filtering algorithm. Technical report, UCB ERL Technical Report M00/41, 2000.

Granovetter, M. The strength of weak ties. *American Journal Sociology*, 78(6):1360–1380, 1973.

Gravano, L., H. Gracia-Molina, and A. Tomasic. GlOSS: Text-source discovery over the internet. *ACM Transactions on Database Systems*, 24(2):229–264, 1999.

Greco, G., S. Greco, and E. Zumpano. Web communities: Models and algorithms. *World Wide Web: Internet and Web Information Systems*, 7(1):59–82, 2004.

Grigull, J. Virtuelle Wissensgemeinschaften in Peer-to-Peer Netzwerken. Diploma thesis, Institute of Computer Science III, Rheinische Friedrich-Wilhelms-Universität Bonn, 2004.

Guha, R., R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *Proc. of the 13th International World Wide Web Conference (WWW 2004)*, 2004.

Gupta, D., M. Digiovanni, H. Narita, and K. Goldberg. Jester 2.0: evaluation of an new linear time collaborative filtering algorithm. In *Proc. of the 22th Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1999)*, 1999.

Haneke, E. NewsSIEVE - Ein selbstadaptiver Filter für textuelle Informationen. Technical report, Institut für Informatik III (IAI-TR-2001-1), Universität Bonn, 2001.

Haneke, E. Learning based filtering of text information using simple interest profiles. In *CIA '97: Proceedings of the First International Workshop on Cooperative Information Agents*, 1997.

Haneke, E. *NewsSIEVE - Ein selbstadaptiver Filter für textuelle Informationen*. PhD thesis, Institute of Computer Science III, University of Bonn, 2000.

Hansen, P. and K. Järvelin. Collaborative information retrieval in an information-intensive domain. *Information Processing & Management*, 41(1):1101–1119, 2005.

Haveliwala, T. H. Topic-sensitive pagerank. In *Proc. of the 11th International World Wide Web Conference (WWW 2002)*, 2002.

Henzinger, M. Link analysis in web information retrieval. *IEEE Data Engineering Bulletin*, 23(3):3–8, 2000.

Herlocker, J., J. A. Konstan, and J. Riedl. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Information Retrieval*, 5(4):287–310, 2002.

Herlocker, J. L., J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *Proc. of the 22th Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1999)*, 1999.

Herlocker, J. L., J. A. Konstan, and J. Riedl. Explaining collaborative filtering recommendations. In *Proc. of the 2000 ACM conference on Computer supported cooperative work (CSCW)*, 2000.

Hill, W. C., J. D. Hollan, D. Wrobelwski, and T. McCandless. Read wear and edit wear. In *Proc. of the ACM Conference on Human Factors in Computing Systems (CHI '92)*, 1992.

Hoashi, K., K. Matsumoto, N. Inoue, and K. Hashimoto. Query expansion method based on word contribution. In *Proc. of the 22th Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1999)*, 1999.

Hoashi, K., K. Matsumoto, N. Inoue, and K. Hashimoto. Document filtering method using non-relevant information profile. In *Proc. of the 23th Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2000)*, 2000.

Hofmann, T. *Data Clustering and Beyond: A deterministic annealing framework for exploratory data analysis*. PhD thesis, Institute of Computer Science III, University of Bonn, 1998.

Hofmann, T. Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual ACM Conference on Research and Development in Information Retrieval*, 1999.

Hofmann, T. Learning what people (don't) want. In *Proc. of the 12th European Conference on Machine Learning (EMCL 2001)*, 2001.

Hofmann, T. Collaborative filtering via gaussian probabilistic latent semantic analysis. In *Proc. of the 26th Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2003)*, 2003.

Houseman, E. M. and D. E. Kaskela. State of the art of selective dissemination of information. *IEEE Transaction Engineering Writing Speech III*, 2:78–83, 1970.

Howe, A. E. and D. Dreilinger. Savvysearch: A metasearch engine that learns which search engines to query. *AI Magazine*, 18(2):19–25, 1997.

Huang, Z., H. Chen, and D. Zeng. Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. *ACM Transactions on Office Information Systems*, 22(1):116–142, 2004.

Hust, A., S. Klink, M. Junker, and A. Dengel. Towards collaborative information retrieval: Three approaches. In Franke, J., G. Nakhaeizadeh, and I. Renz, editors, *Text Mining – Theoretical Aspects and Applications*, Advances in Soft Computing, pages 97–112. Springer Verlag, Berlin, Germany, 2002.

Ide, E. *New experiments in relevance feedback*. Prentice Hall, Englewood Cliffs, NJ, 1971.

Ide, E. and G. Salton. *Interactive search strategies and dynamic file organization in information retrieval*. Prentice Hall, Englewood Cliffs, NJ, 1971.

Ikpaahindi, L. An overview of bibliometrics: its measurements, laws and their applications. *Libri*, 35:163–170, 1985.

Ingwerson, P. Cognitive perspectives of information retrieval. *Journal of Documentation*, 52 (1):3–50, 1996.

Janes, J. W. Relevance judgements and the incremental presentation of document representations. *Information Processing & Management*, 27(6):629–646, 1991.

Jansen, B. J. An investigation on simple queries in web ir systems. *Information Research*, 6 (1), 2000.

Jansen, B. J., A. Spink, and T. Saracevic. Searchers, the subject they search, and sufficiency: A study of a large sample of excite searchers. In *Proceedings of the 1998 World Conference on the WWW and Internet.*, pages 828–833, Orlando, FL, 1998.

Jin, R., J. Y. Cai, and L. Si. An automatic weighting scheme for collaborative filtering. In *Proc. of the 27th Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2004)*, 2004.

Joachims, T., D. Freitag, and T. Mitchell. Webwatcher: A tour guide for the world wide web. In *Proc. of the 11th International Joint Conference on Artificial Intelligence*, 1997.

Jones, W., H. Bruce, and S. Dumais. Keeping found things found on the web. In *Proc. of the 10th International Conference on Information and Knowledge Management*, 2001.

Joshi, A. On proxy agents, mobility, and web access. *Mobile Networks and Applications*, 5 (4):233–241, 2000.

Järvelin, K. and T. D. Wilson. On conceptual models for information seeking and retrieval research. *Information Research*, 9(1), 2003.

Kamvar, S. D., M. T. Schlosser, and H. Garcia-Molina. The EigenTrust algorithm for reputation management in p2p networks. In *Proc. of the 12th International World Wide Web Conference (WWW 2003)*, 2003.

Kautz, H., B. Selman, and M. Shah. Referral web: Combining social networks and collaborative filtering. *Communications of the ACM*, 40(3):63–65, 1997.

Kessler, M. Bibliographic coupling between scientific papers. *American Documentation*, 14 (1):10–25, 1963.

Kim, D.-H., V. Atluri, M. Bieber, N. Adam, and Y. Yesha. A clickstream-based collaborative filtering personalization model: towards a better performance. In *Proc. of the 6th annual ACM international workshop on Web information and data management (WIDM 2004)*, 2004.

Kirsch, S. M., M. Gnasa, and A. B. Cremers. Retrieval in social information spaces. In *Proc. of the 28th BCS-IRSG European Colloquium on IR Research (ECIR 2006)*, 2006.

Kirsch, S. M. Social information retrieval. Master's thesis, Institute of Computer Science III, Rheinische Friedrich-Wilhelms-Universität Bonn, 2005.

Kivinen, J. and M. K. Warmuth. Exponentiated gradient versus gradient descent for linear predictor. Technical report, UCSC-CRL-94-16, Baskin Center for Computer Engineering and Information Science, University of California, Santa Cruz, 2003.

Kleinberg, J., R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. The web as a graph: Measurements, models, and methods. In *Proc. of the 5th International Conference on Computing and Combinatorics*, 1999.

Kleinberg, J. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46 (5):604–632, 1999.

Knott, G. D. Hashing functions. *Computer Journal*, 18(3):265–278, 1975.

Knuth, D. E. *The Art of Computer Programming - Sorting and Searching*, volume 3. Addison-Wesley, Reading, MA, USA, 1973.

Konstan, J. A., B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl. Applying collaborative filtering to usenet news. *Communications of the ACM*, 40(3):77–87, 1997.

Kowalski, G. *Information Retrieval Systems - Theory and Implementation*. Kluwer Academic Publishers, Dordrecht, Norwell, 1997.

Kuflik, T. and P. Shoval. Generation of user profiles for information filtering - research agenda. In *Proc. of the 23th Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2000)*, 2000.

Kuhlthau, C. C. Inside the search process: information seeking from the user's perspective. *Journal of the American Society for Information Science*, 42(5):361–371, 1991.

Kumar, S. R., P. Raghavan, S. Rajagopalan, and A. Tomkins. Trawling the web for emerging cyber-communities. In *Proc. of the 8th International World Wide Web Conference (WWW 1999)*, 1999.

Lam, W., S. Mukhopadhyay, J. Mostafa, and M. Palakal. Detection of shifts in user interests for personalized information filtering. In *Proc. of the 19th Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1996)*, 1996.

Lesk, M. The seven ages of information retrieval. In *Conference for the 50th Anniversary of As We May Think*, Cambridge, MA, 1995.

Lewis, D. D., R. E. Schapire, J. P. Callan, and R. Papka. Training algorithms for linear text classifiers. In *Proc. of the 19th Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1996)*, 1996.

Li, L., Y. Shang, and W. Zhang. Improvement of hits-based algorithms on web documents. In *Proc. of the 11th International World Wide Web Conference (WWW 2002)*, 2002.

Linden, G., B. Smith, and J. York. Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, 2002.

Loeb, S. and D. Terry. Information filtering. *Communications of the ACM*, 35(12):26–28, 1992.

Lopez-Pujalte, C., V. Guerrero-Bote, and F. de Moya-Anegon. Genetic algorithms in relevance feedback: a second test and new contributions. *Information Processing & Management*, 39(5):669–687, 2003.

Loser, A., W. Nejdl, M. Wolpers, and W. Siberski. Information integration in schema-based peer-to-peer networks. In *Proceedings of 15th Conference on Advanced Information Systems Engineering*, 2003.

Lu, J. and J. Callan. Content-based retrieval in hybrid peer-to-peer networks. In *Proceedings of 12th International Conference on Information and Knowledge Management*, 2003.

Lueg, C. *Exploring Interaction and Participation to Support Information Seeking in a Social Information Space*. Springer Verlag, Berlin, 2003.

Luhn, H. P. A business intelligent system. *IBM Journal of Research and Development*, 2(4): 314–319, 1958.

Lyman, P. and H. Varian. How much information? Technical report, University of California at Berkeley, 2000.

Malone, T. W., K. R. Grant, F. A. Turbak, S. A. Brobst, and M. D. Cohen. Intelligent information-sharing systems. *Communications of the ACM*, 30(5):390–402, 1987.

Maltz, D. and K. Ehrlich. Pointing the way: active collaborative filtering. In *Proc. of the 1995 ACM Conference on Human Factors in Computing Systems*, 1995.

Marchionini, G. *Information seeking in Electronic Environments*. Cambridge University Press, 1995.

Masinter, L. and E. Ostrom. Collaborative information retrieval: Gopher from moo. In *Proceedings of INET'93*, 1993.

Meng, W., C. T. Yu, and K.-L. Liu. Building efficient and effective metasearch engines. *ACM Computing Surveys*, 34(1):48–89, 2002.

Miller, B., I. Albert, S. Lam, J. Konstan, and J. Riedl. Movielens unplugged: Experiences with a recommender system on four mobile devices. In *Proceedings of the 17th Annual Human-Computer Interaction Conference (HCI 2003)*, 2003a.

Miller, B. N., J. T. Riedl, and J. A. Kostan. *GroupLens for Usenet: Experiences in Applying Collaborative Filtering to a Social Information System*. Springer Verlag, Berlin, 2003b.

Morita, M. and Y. Shinoda. Information filtering based on user behavior analysis and best match text retrieval. In *Proc. of the 17th Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1994)*, 1994.

Mulvenna, M. D., S. S. Anand, and A. G. Buchner. Personalization on the net using web mining. *Communications of the ACM*, 43(8):123–125, 2000.

Munro, A. J., K. Höök, and D. Benyon. *Social navigation of information spaces*. Springer-Verlag, Berlin, Germany, 1999.

Nelson, T. The hypertext. In *Proceedings of the World Documentation Federation*, 1965.

Newman, M. E. J. and J. Park. Why social networks are different from other types of networks. *Physical Review E*, 68(3):036122, 2003.

Ng, A. Y., A. X. Zheng, and M. I. Jordan. Stable algorithms for link analysis. In *Proc. of the 24th Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2001)*, 2001.

Nichols, D. Implicit rating and filtering. In *Proceedings of 5th DELOS Workshop on Filtering and Collaborative Filtering*, 1998.

Oard, D. and J. Kim. Implicit feedback for recommender systems. In *Proceedings of the AAAI Workshop on Recommender Systems, July 1998.*, 1998.

Oard, D. W. The state of the art in text filtering. *User Modeling and User-Adapted Interaction*, 7(3):141–178, 1997.

Ostrom, E. *Governing the Commons: The Evolution of Institutions for Collective Action*. Cambridge University Press, Cambridge, MA, USA, 1990.

Packer, K. H. and D. Soergel. The importance of sdi for current awareness in fields with server scatter of information. *Journal of the American Society for Information Science*, 30 (3):125–135, 1979.

Pennock, D. M., G. W. Flake, S. Lawrence, E. J. Glover, and C. L. Giles. Winners don't take all: Characterizing the competition for links on the web. *Proceedings of the National Academy of Sciences*, 99(8):5207–5211, 2002.

Pennock, D. M., E. Horvitz, and C. L. Giles. Social choice theory and recommender systems: Analysis of the axiomatic foundations of collaborative filtering. In *Proc. of the 17th Conference on Artificial Intelligence (AAAI 2000)*, 2000a.

Pennock, D. M., E. Horvitz, S. Lawrence, and C. L. Giles. Collaborative Filtering by Personality Diagnosis: A Hybrid Memory- and Model-Based Approach. In *Proc. of the 6th Conference on Uncertainty in Artificial Intelligence (UAI 2000)*, 2000b.

Pitkow, J., H. Schütze, T. Cass, R. Cooley, D. Turnbull, A. Edmonds, E. Adar, and T. Breuel. Personalized search. *Communications of the ACM*, 45(9):50–55, 2002.

Pretschner, A. and S. Gauch. Personalization on the web. Technical Report ITTC-FY2000-TR-13591-01, Information and Telecommunication Technology Center, Department of Electrical Engineering and Computer Science, University of Kansas, 1999.

Ratnasamy, S., P. Francis, M. Handley, R. Karp, and S. Shenker. A Scalable Content Addressable Network. In *Proceedings of ACM SIGCOMM*, 2001.

Resnick, P., N. Iacovou, M. Suchak, P. Bergstorm, and J. Riedl. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In *Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work*, pages 175–186, Chapel Hill, North Carolina, 1994. ACM.

Resnick, P. and H. Varian. Recommender systems. *Communications of the ACM*, 40(3): 56–58, 1997.

Resnik, P. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research (JAIR)*, 11:95–130, 1999.

Robertson, S. E. and K. Sparck Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27(3):129–146, 1976.

Robertson, S. E., S. Walker, M. Hancock-Beaulieu, M. Gatford, and A. Payne. Okapi at trec-4. In Voorhees, E. M. and D. K. Harman, editors, *Proceedings of the Fourth Text REtrieval Conference TREC-4*, number 500-236 in NIST Special Publications. U.S. National Institute of Standards and Technology (NIST), 1995.

Rocchio, J. Relevance feedback in information retrieval. In Salton, G., editor, *The SMART Retrieval System: Experiments in Automatic Document Processing*, pages 313–323. Prentice-Hall, Englewood Cliffs, NJ, United States, 1971.

Romano, N. C., D. Roussinov, Jay F. Nunamaker, and H. Chen. Collaborative Information Retrieval Environment: Integration of information retrieval with group support systems. In *Proceedings of the 32nd Hawaii International Conference on System Science*, 1999.

Ruhl, T. Personal Search Memory - Design und Realisierung einer Suchschnittstelle zur kombinierten Suche in früheren und neuen Suchergebnissen. Diploma thesis, Rheinische Friedrich-Wilhelms-Universität Bonn, 2003.

Ruthven, I. and M. Lalmas. A survey on the use of relevance feedback for information access systems. *Knowledge Engineering Review*, 18(2):95–145, 2003.

Salton, G. and C. Buckley. On the use of spreading activation methods in automatic information. In *Proc. of the 11th Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1988)*, 1988.

Salton, G. *The SMART Retrieval System - Experiments in Automatic Document Processing*. Prentice Hall, Englewood Cliffs, NJ, 1971.

Salton, G. and C. Buckley. Improving retrieval performance by relevance feedback. *Journal of American Society for Information Sciences*, 41(4):288–297, 1990.

Salton, G. and M. McGill. *An Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.

Schutz, A. and T. Luckmann, editors. *Structures of the Life World*. Northwestern University Press, Evanston, Ill., 1973.

Shardanand, A. and P. Maes. Social information filtering: algorithms for automating ”word of mouth”. In *Proc. of the 1995 ACM Conference on Human Factors in Computing Systems*, 1995.

Shneiderman, B. Codex, memex, genex: The pursuit of transformational technologies. *International Journal of Human-Computer Interaction*, 10(2):87–106, 1998.

Silverstein, C., M. Henzinger, H. Marias, and M. Moricz. Analysis of a very large web search engine query log. *SIGIR Forum*, 33(1):6–12, 1999.

Singhal, A., C. Buckley, and M. Mitra. Pivoted document length normalization. In *Proc. of the 19th Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1996)*, 1996.

Singhal, A., M. Mitra, and C. Buckley. Learning routing queries in a query zone. In *Proc. of the 20th Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1997)*, 1997.

Small, H. Co-citation in the scientific literature. *Journal of the American Society for Information Science*, 24(4):265–269, 1973.

Smeaton, A. F. Filtering news and WWW pages with the borges information filtering tool. In *Proc. of the Workshop on Practical Applications of Information Filtering*, 1996.

Smith, M. A. *Measures and Maps of Usenet*. Springer Verlag, Berlin, 2003.

Sparck Jones, K., S. Walker, and S. A. Robertson. Probabilistic model of information retrieval: Development and status. Technical report, TR-446, Cambridge University, Cambridge University, 1998.

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

Spink, A., H. Greisdorf, and J. Bateman. From highly relevant to not relevant: examining different regions of relevance. *Information Processing & Management*, 34(5):599–621, 1998.

Srivastava, J., R. Cooley, M. Deshpande, and P.-N. Tan. Web usage mining: Discovery and applications of usage patterns from web data. *SIGKDD Explorations*, 1(2):12–23, 2000.

Steinmetz, R. and K. Wehrle. *Peer-to-Peer Systems and Applications*. Springer-Verlag, Berlin, Germany, 2005.

Tang, C. and S. Dwarkadas. Hybrid global-local indexing for efficient peer-to-peer information retrieval. In *Proceedings of the Symposium on Networked Systems Design and Implementation (NSDI)*, 2004.

Tang, C., Z. Xu, and S. Dwarkadas. Peer-to-peer information retrieval using self-organizing semantic overlay networks. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, 2003.

Taylor, R. S. The process of asking questions. *American Documentation*, 13(4):391–396, 1962.

Teevan, J., S. Dumais, and E. Horvitz. Beyond the commons: Investigating the value of personalizing web search. In *Proceedings of the Workshop on New Technologies for Personalized Information Access*, 2005a.

Teevan, J., S. Dumais, and E. Horvitz. Personalizing search via automated analysis of interests and activities. In *Proc. of the 28th Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2005)*, 2005b.

Tempich, C., S. Staab, and A. Wranik. REMINDIN: Semantic query routing in peer-to-peer networks based on social metaphors. In *Proc. of the 13th International World Wide Web Conference (WWW 2004)*, 2004.

Thor, A. and E. Rahm. AWESOME - a data warehouse-based system for adaptive website recommendations. In *Proc. of the 30th International Conference on Very Large Databases*, 2004.

Tryfonopoulos, C., M. Koubarakis, and Y. Drougas. Filtering algorithms for information retrieval models with named attributes and proximity operators. In *Proceedings of the 27th annual international conference on Research and development in information retrieval*, 2004.

van Rijsbergen, C. J. *Information Retrieval*. Butterworths, 1979.

Watters, C. and M. A. Shepherd. Shifting the information paradigm from data-centered to user-centered. *Information Processing & Management*, 30(4):455–471, 1994.

Watts, D. J. and S. H. Strogatz. Collective dynamics of small-world networks. *Nature*, 393: 440–442, 1998.

Wedeles, L. Professor nelson talk analyzes p.r.i.d.e. *Miscellany News*, 1965.

Wen, J.-R., J.-Y. Nie, and H.-J. Zhang. Query clustering using user logs. *ACM Transactions on Information Systems*, 20(1):59–81, 2002.

White, R. W., J. M. Jose, and I. Ruthven. An approach for implicitly detecting information needs. In *Proc. of the 12th international conference on information and knowledge management (CIKM 2003)*, 2003.

White, R. W., J. M. Jose, and I. Ruthven. An implicit feedback approach for interactive information retrieval. *Information Processing and Management*, 42(1):166–190, 2004.

White, R. W., I. Ruthven, and J. M. Jose. The use of implicit evidence for relevance feedback in web retrieval. In *Proc. of the 24th BCS-IRSG European Colloquium on IR Research (ECIR 2002)*, 2002.

Widrow, B. and M. E. Hoff. Adaptive switching circuits. *IRE WESCON Convention Record*, Part IV:96–104, 1960.

Wilson, T. On user studies and information needs. *Journal of Documentation*, 37(1):3–15, 1981.

Wolff, J. E. *Integration und Individualisierung von Quellen im Internet*. PhD thesis, Institute of Computer Science III, University of Bonn, 2000.

Wolff, J. E. and A. B. Cremers. The MyVIEW project: A data warehousing approach to personalized digital libraries. In *Proc. of the Fourth Int. Workshop on Next Generation Information Technologies and Systems*, pages 277–294. Springer-Verlag, 1999.

Xu, J. and W. B. Croft. Query expansion using local and global document analysis. In *SIGIR96*, 1996.

Yan, T. and H. Garcia-Molina. Sift: A tool for wide-area information dissemination. In *Proc. of the USENIX 1995 Winter Technical Conference*, 1995.

Zachary, G. P. The godfather. *Wired*, 5(11), 1997.

Zhang, Y. and J. Callan. Maximum likelihood estimation for filtering thresholds. In *Proc. of the 24th Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2001)*, 2001.

Zhang, Y., J. Callan, and T. Minka. Novelty and redundancy detection in adaptive filtering. In *Proc. of the 25th Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2002)*, 2002.

Zobel, J. and A. Moffat. Exploring the similarity space. *SIGIR Forum*, 32(1):18–34, 1998.

Glossary

Active Peer A client peer is denoted as active if an automated system process is initiated on it by a trigger or by a user.

Candidate Set All timestamped contexts are stored in a candidate set if they are not similar to a collaborative search context.

Collaborative Search Context A collaborative search context groups overlapping interests of users. It identifies relevant documents for a set of query terms with a high commitment by a user group.

Cooperative Pull-Push Cycle All interactions between information consumers and information providers are based on a cooperative pull-push cycle. Each peer works with other peers for a common purpose by a retrieval of information during the pulling phase, and by a propagation of information during the pushing phase.

External Provider Web search engines are external providers which collect a large amount of information. These services provide no information of how the ranked output has been computed.

Grouped Search Session All search sessions are grouped by users, queries, and days. A grouped search session consists of a user's query and a set of links that have been viewed by the user at one day.

Giant Component The giant component of a network is a connected subgraph that contains a majority of the entire graph vertices.

Integrated Information Seeking In order to match highly variable interests with rapidly changing information, a new process denoted as integrated information seeking is defined.

Internal Provider All users of the ISKODOR system are denoted as internal providers. Each user maintains a local information source that is dynamically updated with each search session.

Passive Peer A client peer is denoted as passive if the user does not need to initiate a system reaction, if the peer is requested. It automatically answers an information request of another peer. The peer itself is not passive only the user status.

Peer Search Memory Each client peer maintains a Peer Search Memory that stores individual relevance feedback.

Reference Category All links grouped by a community can have an Open Directory Project category. The reference category of a community is the category with the largest number of links. If more than one category fulfills this criteria, the reference category is randomly selected.

Search Session Each search session specifies a 4-tuple (q, p, t, u) where q is a query of user u who views a Web page p at time t.

Search Context Each query and the documents which have been assessed as relevant by a user define an individual search context.

Service Repository The service repository is a temporary storage of documents which are recommended by peers or which will be recommended to other peers.

Timestamped Context A timestamped context of a user u is a 4-tuple (q, l, s, u) where q is a query with a relevant Web page (URL) l at time s.

Virtual Knowledge Community All users with common search interests are grouped into Virtual Knowledge Communities. The grouping is an automated process that suggests community memberships. A membership must be explicitly confirmed by a user. Communities are incorporated in the integrated information seeking process.

Virtual Search Network A virtual search network is modelled by all information providers and their interactions.

Weak Connected Component A weak component of a directed graph is a subgraph so that the corresponding subgraph in the underlying undirected graph is connected.