Journal of Web Engineering, Vol. 3, No. 3&4 (2004) 252–280

THREE PILLARS FOR CONGENIAL WEB SEARCHING
Continuous Evaluation for enhancing Web Search Effectiveness

MELANIE GNASA, MARKUS WON, and ARMIN B. CREMERS
Institute of Computer Science III, University of Bonn, Römerstrasse 164
53117 Bonn, Germany
gnasa, won, abc@iai.uni-bonn.de

Received July 17, 2004
Revised December 23, 2004

In the context of large homogeneous retrieval systems, metrics have been established to evaluate the effectiveness with precision and recall. By contrast, measuring Web search effectiveness is a new challenge due to the heterogeneity of high-dynamic Web content. Currently, users select a Web search engine by their individual preferences, and the evaluation of effectiveness is a subjective measure defined by the user. Since there are different emphases for each single user, those user-defined measures cannot be quantified in a global way. Therefore, we propose a new Web search system, where the effectiveness is continuously evaluated by explicit user feedback in terms of a personalized ranking matrix. These local rankings can be evaluated according to different goals. First, accumulation leads to a wider base of ranked and validated results. Second, the aggregated ranking lists can be used to identify topics, as well as communities of interest. Finally, together with social aspects for community support, a framework for Congenial Web Search is defined.

Keywords: Web Information Retrieval, Personalization, Web Mining, Community-based Filtering, Peer-to-Peer
Communicated by: A Spink & C Watters

1 Introduction

The Web is extremely large 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 this may not be interpreted as an indication for their optimal effectiveness. Today, the retrieving process is optimized for the processing of several thousand queries per second. In this regard, Web search engines do a good job. New system architectures should not claim to exceed existing systems in their coverage of the Web, and their efficiency. From the user's perspective, efficiency is part of his overall subjective impression of a Web search engine. However, regarding the effectiveness, no statistical evaluation can represent the quality of a search result for each individual user. Even identical queries enunciate different information needs, because each user differs in his social environment. Recently, Web personalization strategies have been developed to overcome this shortcoming. These strategies can be defined as actions to adapt information, or services, provided by a Web site, to the needs of the individual user [21]. For this task,

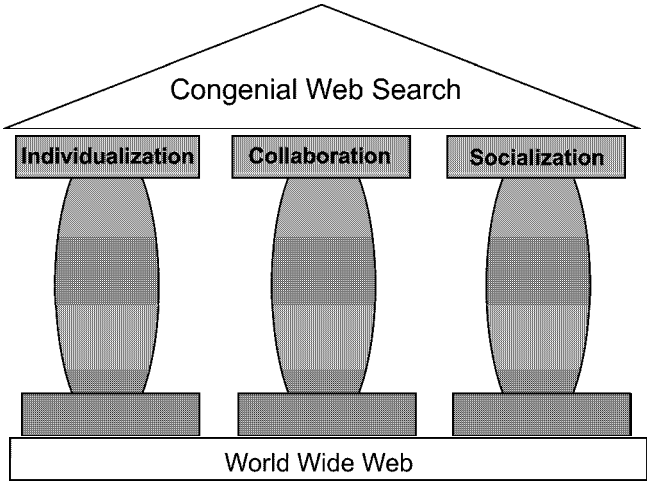

Fig. 1. "Three Pillars" for Congenial Web Search

navigational behavior and individual interests are taken into account. Thus, the main goal of Web personalization is the determination of relevant information without an explicit request [36]. An evaluation by Joachims [28] shows that search engines could be optimized by using clickthrough data. This data is available as query logs in abundance, and can easily be used to learn retrieval functions. Nevertheless, an exploratory study by Khopkar et al. [31] shows that despite the high level of interest in this topic, most Web search engines currently offer no, or just limited personalization features.

So, our approach concentrates on a local personalization strategy for Congenial Web Searching, which is independent from a central server or a Web search engine. The adjective "*congenial*" should clarify the fact that new advantages for a user-centered Web search are to be gained through continuous evaluation of explicit user feedback. Based on the integration of recent Web search engines, the evaluation is done explicitly by the user, and implicitly by the system by which ranked results are weighted. At the same time, it is known that Information Retrieval is always a social process [51]. In traditional knowledge gathering, a user first searches within his local community (e.g. colleagues or family members) before other trusted sources are requested. The annihilation of the user-anonymity in the Web is a crucial challenge to encourage the formation of trusted groups, where users can confirm their information needs. For this purpose, we suggest the automatic proposal of Virtual Knowledge Communities [22], and the classification of common users and experts according to their standard of knowledge. In order to avoid a strict classification into group members and non-group members, degrees of socialization are also proposed regarding an additional community support with our approach. Summarized within three pillars of Congenial Web Searching (cf. Figure 1), our user-centered framework will be described in the next chapter. No average effectiveness measure must be defined, because the system optimizes the effectiveness for each user, individually. With ISKODOR there exists a first prototype of the proposed system; a brief evaluation will be given in Section 8.

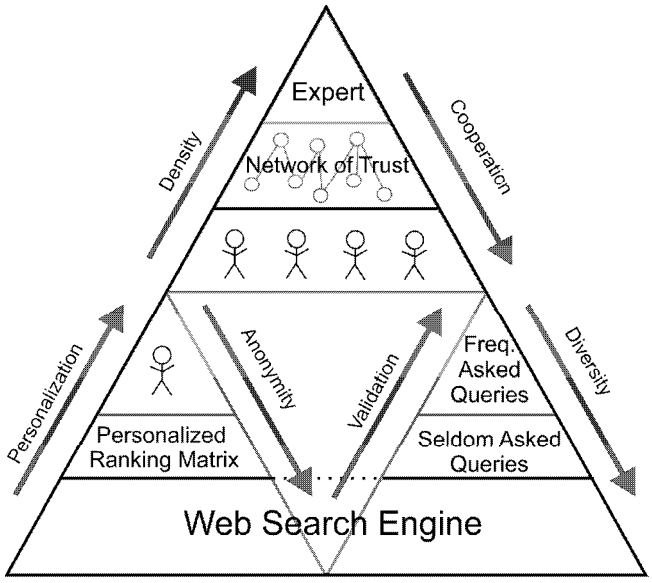

Fig. 2. Feature Pyramid for Congenial Web Search

2 "Three Pillars" - A conceptual overview

The Congenial Retrieval Model supports a framework for document representations, queries, and their relationships. Users and their information needs are represented by queries. A correlation between queries and a user is not represented with traditional (Web) retrieval systems. Web Search engines generally treat search requests in isolation [33]. Hence, each query is assumed as individual *context* information of a user. This assumption is similar to the SearchPad approach by Bharat [13]. In order to support optimal effectiveness, the user, as the central part of a retrieval system, is integrated into a new framework for a congenial Web search. For this task, three concepts build the pillars for the new framework: individualization, collaboration, and socialization.

With Figure 2, we give an overview about all of the concepts and their features. Each concept is visualized in a pyramid, in order to explain the features of a Congenial Web search system. The first pyramid on the left side represents our individualization issues. The traditional Web Search is enriched by locally stored transaction logs (matrices). This user-specific matrix allows to keep track of a search context, explicitly. Furthermore, we build personalized ranking matrices, which include an access relevance for all prior viewed documents. A personalized search is achieved by filtering of documents according to prior information needs. Hence, each user manages a local collection of relevant information, which is used for a continuous evaluation of the entire system. For each user a maximum anonymity can be guaranteed by means of local filtering.

In the second pyramid on the right, all personal transaction matrices are used to group the access information by topics. The problem with current Web search is that all users get the same result for identical queries, and they do not know about each other. Because of this, we propose building a network of interests. Concerning local Personalized Ranking Matrices,

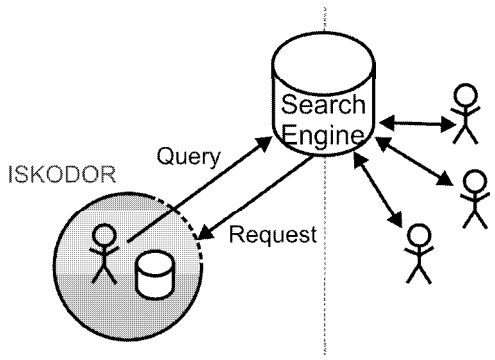

Fig. 3. ISKODOR vs. traditional search engine communication

results can be ranked individually by their level of validation for each user. Results can be validated by the group (several users rank the same documents) or by using external web directories. Unlike Web search engines, we can apply user feedback to distinguish between Seldom and Frequently Asked Queries. We believe that users with similar queries share a common interest. Hence, queries and marked answers are analyzed with this assumption. If such similarities can be found, the corresponding users might belong to the same community. The option of joining a community can be presented automatically, whereas the act of joining has to be committed explicitly. Information on a shared community can be used to get better validation. Furthermore, ratings of community members can be ranked higher than those of others.

Instead of a Web search engine, the users build the foundation for the top pyramid. Our system assists the cooperation of users, in order to be aware of new information in the network. We assume that all users build a network of trust, which occurs by a continuous increase of members who recommend others. In this social network, different degrees of relationships between users and communities can be identified. We can take advantage of the increased cohesion within the community to determine the experts of a community.

3 First Pillar: Individualization

Individualization can be achieved by a personalized filtering of retrieval results for each user. Previously, the user directly interacted with a search engine as depicted in Figure 3. In difference to traditional search requests, ISKODOR facilitates personalized filtering of a Web search result without losing anonymity. Hence, we define fundamental associations between queries and relevant results.

3.1 Explicit Relevance Feedback

Inspired by the concept of search trails pioneered by Vannevar Bush in his fictive system Memex [15], all search processes are recorded, and edited for future access in a *Peer Search Memory*, called PeerSy. Unlike traditional search histories, all documents are stored, which are relevant in a special context for the user. The context is specified by the query. The evaluation of the relevance is being carried out by the user, who flags all documents from the search result that answer his information need. This conforms to *explicit relevance feedback*

(cf. [42], [43]), and is used for a personalized filtering of search results. PeerSy is modelled as a transaction matrix, which stores all fundamental associations between query terms (concepts) and documents.

Definition 1 *Let $I = \{i_1, ..., i_m\}$ be a set of literals, called term items. Let $D = \{d_1, ..., d_n\}$ be a set of literals, called document items. Let Q be a set of query transactions, where each transaction T_Q is a set of term items, such that $T_Q \subseteq I$. Let L be a set of link transactions, where each transaction T_L is a set of document items, such that $T_L \subseteq D$. A $|Q| \times |D|$ transaction matrix M is defined for all fundamental associations between a query transaction q and a document d*

$$M[q,d] = \begin{cases} 1, & q \in Q, d \in T_L, \text{ } d \text{ is relevant for query } q \\ 0, & \text{otherwise} \end{cases}$$

A crucial factor of our system is the dependency of the transaction matrix on information, which has been explicitly given by the user. Usually, users would expect some kind of incentive for their intellectual work. Within the Web context, the best incentive for a user would be better search results in an effective and efficient manner. A study [30] shows that bookmarks are widely used as a strategy for locating information. Hence, users already try organizing their personal information, and it has been observed that this collection grows linearly with time [1]. In our system ISKODOR, we want to exploit this motivation by keeping the interaction with the system as simple as bookmarking.

Summarizing, we collect user feedback in a direct way to build fundamental associations. This explicit relevance feedback allows for a transparent access to personal information during retrieval. Unlike present personalization attempts (e.g. Google Personalized, eurekster.com or Amazon A9), our strategy is only one component of our Congenial Web search system. According to the taxonomy of Web Mining [18] our system combines Web Usage Mining with an agent-based approach. First, for a continuous usage of all associations, our system performs a new organization of Web-based information by filtering. Unlike traditional retrieval systems, where a user initiates an information pull, it is practical to recommend documents to other users with similar interests. This agent-based recommendation process is part of our socialization attempts described in the third pillar (cf. Section 5). For the conceptual basis of this process, we need to define *implicit relevance feedback*. Second, with Web Usage Mining [46] an independent application of generic data mining such as a discovery of association rules is involved. In definition 1, we already used a data mining terminology to describe the transaction matrix. Generally, the problem of mining association rules is comparable to our goal of finding users with similar search interests. Furthermore, we have to deal with the heterogeneity of a sparse global transaction matrix (cf. Section 4).

3.2 Implicit Relevance Feedback

To avoid the difficulties associated with gathering explicit ratings from users, several techniques have been proposed to capture useful information unobtrusively [37]. Oard and Kim [38] identify different sources of implicit feedback, which are mapped into three broad cat-

egories of potentially useful observations: examination, retention, and reference. According to the classification of observable behavior, each fundamental association based on explicit feedback is an example of such a behavior of the retention category. Additionally, *implicit relevance feedback* is computed for each fundamental association of the transaction matrix M, in order to assign a relevance weight between a query, and an evaluated result. Unlike traditional collaborative filtering approaches, we only have binary ratings for documents (is relevant or not), instead of using a numerical scale, for example 1 to 5 stars. The goal of our approach is computing a rating between zero and one for each association. For this task, we compute a *rating matrix* R, which weights all documents of the transaction matrix M.

Definition 2 *A rating matrix R is defined for all documents $d \in D$ ($m = |D|$) with*

$$R = diag(r_1, ..., r_m)$$

Each element of R summarizes all implicit feedback information, where r_d denotes the rating of a document d analogously to traditional weighting formulas (cf. [41]). The local rating of a document $d \in D$ is computed for each term $i \in I$:

$$r_d = \frac{1}{|T_{L,d}|} \sum_{i \in I} w_{d,i} \tag{1}$$

Each rating is normalized by the total number of transactions $T_{L,d} \subset L$ which include document d ($d \in T_{L,d}$).

We currently employ a simple retrieval method based on the traditional $tf * idf$ weighting with cosine normalization [44] for the weight $w_{d,i}$ of a relevant result d as regards the term i. It is calculated from the term frequency $tf_{d,i}$ and the inverse term frequency idf_i as well as the query frequency qf_i in the following formula:

$$w_{d,i} = ((1 - \alpha) * tf_{d,i} + \alpha * qf_i) * idf_i \tag{2}$$

The query frequency, qf_i, with regard to a term i is the number of occurrences of the term in all queries. The parameter α allows a differentiation between terms occurring in results and queries. The range of α is $[0, 1]$.

With the computation of the rating matrix R, we could add content-based weighting of documents to the transaction matrix. This weighting considers no individualization for a single user, because we only have a dependency between terms that occur in queries and documents. To avoid that this weighting is identical for users, which use the same query terms for a common set of relevant documents, we log all further accesses to document items: (1) number of accesses to document d per day (z_d), (2) number of days since the first access of the document d (e_d), and (3) number of days since the last access of the document d (l_d). These observations of a user u are summarized as an *access relevance* for each document

$d \in D$.

$$aRel_d = importance_d * lastAccess_d \tag{3}$$

$$\frac{log_2(z_d + 1)}{log_2(e_d + 2)} * \frac{1}{log_2(l_d + 2)} \tag{4}$$

With the access relevance, we define an individualization of each fundamental association. For each user, these observations are summarized in a *personalized ranking matrix.*

Definition 3 *A personalized ranking matrix is defined for all documents $d \in D$ ($m = |D|$) with*

$$PRL = diag(aRel_1, ..., aRel_m)$$

Summarizing all steps, we distinguished content-specific and usage-specific enhancements of the transaction matrix M. Hence, the Peer Search Memory can be defined as a product of all partial matrices.

Definition 4 *The Peer Search Memory is defined as the product of the transaction matrix M with the rating matrix R and the personalized ranking matrix PRM, i.e.*

$$PeerSy = M * R * PRM$$

Only the weighted transaction matrix assists further processes of the next two pillars. Hence, user-driven information is not shared with other users, and the personalized ranking matrix is solely used to assist a personalized ranking.

3.3 Personalized Ranking

For repeated requests, it is possible to personalize the ranking of formerly relevant results in relation to a Web search engine result. Several other systems have already relied on Vannevar Bush's idea, and are designed to personalize Web searching and browsing (cf. [17] and [16]). Unlike related systems, our approach assists the semantic assignment of relevant sites to a query (in contrast to simple bookmarking), as well as the implicit weighting of associations in our system. With our basic approach, we try to bring the user into the focus of the system, and the ranking is adapted to his personalized ranking matrix. This feature does not lead to additional administration tasks for the user. A personalized ranking is performed for repeated queries in three steps:

[**1st step**] A query relevance is used to compute the degree of overlap between the actual query q_{act} and all query transactions of the past. If the degree exceeds a threshold, we identify a repeated request, and we initiate the second ranking step. Otherwise, no further ranking is performed, and the original ranking of the Web search engine is retained.

[**2nd step**] Only if the actual query matches a known context, we select all documents in our transaction matrix for this query. These documents are ranked by the access relevance, which is stored in the personalized ranking matrix.

[**3rd step**] Finally, we use the rating matrix R to perform the ranking of all documents from both preprocessing steps (1st and 2nd step).

After we identify all matching documents from recent query and link transactions, these documents are merged with results of a Web search engine. The intersection between PeerSy and the Web search engine can be characterized by three sets: (1) a set of documents found by PeerSy and the search engine, (2) a set of documents found only by PeerSy, and (3) a set of documents found only by the search engine. The first two sets are ranked according our personalized ranking scheme. For the last set, we take the ranking order of the Web search engine, because no real-time filtering is applied.

Finally, according to the individual search requirements, the user locally enriches his set of relevant results, which support an optimal effectiveness for his own information needs. However, this optimal result set cannot improve the effectiveness for new or extended information needs. For these purposes, it is advantageous to take personalized ranking matrices of all users into account, as discussed with the next pillar.

4 Second Pillar: Collaboration

From the viewpoint of an individual user, the result of a Web request only depends on the query, and not on the user. Today, there is no awareness of other users with similar interests. Hence, for an enhancement of search results, identical queries and relevant results collected by all users have to be identified. This leads to a wider base of ranked and validated results. Based on a set of users U, all local query transactions as well as link transactions (cf. definition 1) are combined to achieve a community support through individual user interests. Hence, the global transaction matrix $globalM$ summarizes all local transaction matrices from the users, i.e.,

$$globalM = \sum_{u \in U} M_u$$

Only the content-specific part $(M * R)$ of the Peer Search Memory is used for the identification of common search interests. No personalized ranking matrix is shared with other users. For an efficient sum up of all local matrices, we assign global term ids.

The analysis of the global transaction matrix shows that repeated queries and results are a sign for the distributed occurrence of related information needs. This process is similar to Web usage mining where transaction logs are used to personalize Web sites [21]. One possible technique used during the pattern discovery process is mining association rules. The goal of discovering association rules can be decomposed in two subproblems [3]: (1) finding large itemsets that have transaction support above a minimum support, and (2) use the large itemsets to generate the desired rules. The first step of mining association rules is comparable to our approach as described in the following section. The main difference is the waiving of a support value during the computation of large itemsets. Algorithms like Apriori or AprioriTid

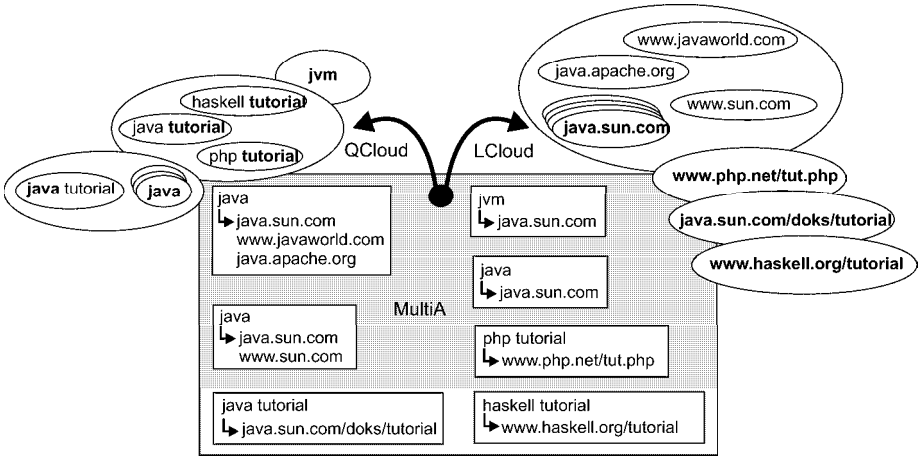

Fig. 4. QCloud and LCloud example

[4] define a minimum support during the computation of large itemsets. For our data set, a minimum support can not be defined, because of the heterogeneity of all query terms. We analyzed a query log with nearly 30,000 queries, where one third of all queries occur only once. Further details of our evaluation are discussed in Section 8. Hence, techniques used for Web usage mining are not applicable in our context. Instead, we developed a comparable approach, where no support value must be defined.

4.1 Aggregation of similar Search Interests

Based on the global transaction matrix, we want to find all sets of items (*itemsets*) that have transaction support greater than zero. For this task, we distinguish between sets of term items (*termsets*) and sets of document items (*documentsets*). Through the combination of all local user interests, two kinds of overlaps can be identified based on all termsets and documentsets.

First, identical query terms are merged, in order to build sets of aligned queries. This set of query associations is called *QCloud* [22]. In difference to the Apriori algorithm, we use multi-sets. This means that identical items can occur multiple times in the same itemset. Based on a sample of associations, the grouping of queries is depicted in Figure 4. In the example, the set *QCloud* consists of three elements. Each element is a termset, which have common query terms. For the example, all four queries of the first element include the term *"java"*.

Second, an aggregation of identical results offers the possibility of new relevant results, which are identified by different users in the same context. With the generated set *LCloud* [22] a summary of validated results can be provided for all users. Each documentset is computed in the context of a query transaction. Through the summary all of local transaction matrices, we can identify documents which occurred in different search contexts. Figure 4 shows *LCloud* for our example set. *LCloud* consists of four link sets. Only the first documentset groups different links. The link that is responsible for this grouping, is *"java.sun.com"*.

In contrast to traditional Web search engines, both types of termsets and documentsets facilitate the classification of search requests based on previous queries and relevant results.

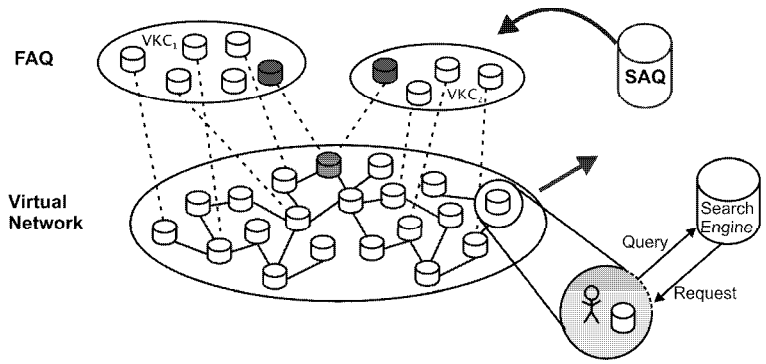

Fig. 5. Virtual network of personalized ranking matrices

Furthermore, a grouping of users with similar interests is achieved.

4.2 Classification of Search Requests

The problem with recent Web search is that all users get the same result for identical queries, and they do not know about each other. It would be a great improvement, if similar interests and feedback information are used to enhance the effectiveness. For the detection of similar interests, the search behavior must be analyzed. Observations of search engines show that the frequency of popular queries conforms to the Bradford distribution (cf. [14]). 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, our system is designed to work with the Bradford Distribution as arrogated by Bates [7]. For this reason, all search requests, which are explicitly evaluated by a user, are automatically classified. As visualized in Figure 5, we assume a virtual network of all local transaction matrices. Based on all local feedback assessments, our strategy classifies queries and relevant results in *Seldom Asked* and *Frequently Asked Queries.* In the first step, all assessments are ranked as Seldom Asked Queries. This set contains most of the requested queries. Only if a representative number of commitments is available, queries and documents are transferred to the set of Frequently Asked Queries. Due to this classification, all users that agree with this topic, have the possibility of joining this community. These memberships maintain a network of Virtual Knowledge Communities (cf. Figure 5). Following, each classification step is explained in detail:

(1) Seldom Asked Queries (SAQ): Due to effective processing, this approach identifies seldom asked questions. These questions can be characterized by a collection of high-quality results, and their corresponding requests besides mainstream topics. While searching, the SAQ set provides the user with assistance for query trend detection. The computation of SAQ occurs in two steps, based on the sets $QCloud$ and $LCloud$. In the first step, for a set of users U, and their fundamental query associations, a set of *Seldom Asked Queries* is represented by

$$SAQ'(MultiA) = \{(s,b)|s \in QCloud(MultiA) \wedge b = LCloud[s]\}$$

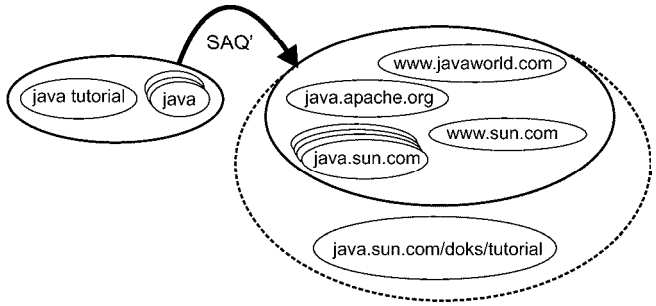

Fig. 6. First SAQ processing step

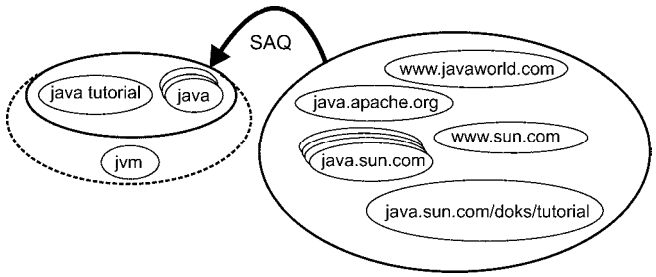

Fig. 7. Second SAQ processing step

Based on the example set in of Figure 4, the first grouping step of SAQ' is presented in Figure 6. This figure exemplifies the building of new query and link associations. Starting with one element of QCloud, all corresponding links in LCloud are aligned. The aggregation of *"java"* and *"java tutorial"* causes the enhancement of the original link set, which was associated with *"java"*. In cases where the same links are related to different queries, the fundamental associations are enhanced with these queries. The intention of the addition of SAQ' is the collection of different descriptions, which lead to the same relevant result. Recapitulating, the final SAQ definition is the following formula:

$$SAQ(MultiA) = \{(s,b)|b \in range(SAQ') \wedge s = \bigcup SAQ'^{-1}[b]\}$$

As depicted in Figure 7, this second step maintains an enhancement of the original query set. A new query (*"jvm"*) is associated with these queries, because of an identical link assessment (*"java.sun.com"*). A study by Jansen and Spink [26] analyzes the usage of common Web search engines. They found that half of the users view only the highly-ranked Web sites on the first page. The advantage of SAQ is that minority ratings are not lost with the system, and can be used as candidates for future growing interests. With this approach, such topics can be propagated without a huge community of interest.

(2) Frequently Asked Queries (FAQ): By the initial increase of the SAQ list, no frequently asked requests can be determined, in order to characterize mainstreams. While

more and more users generate explicit feedback information for Web sites, numerous relevant results are collected. This means that many users request the same query terms, and they assign identical results as relevant. A simple analysis would be identifying all elements in SAQ that have a certain number of queries (x) and associated results (y).

There are two main problem with this accumulation strategy, which can lead to a misclassification of topic trends. First, polysemy of query terms can result in a misleadingly grouping of associations referring to different contexts. For example, the query "sun" can be requested, in order to get information about a company name[a] or our universe. Second, due to synonymous terms (e.g. notebook and laptop), queries are not aligned, even if they are requested in the same context. To overcome these shortcomings, techniques like clustering [53, 50], or latent semantic analysis [12, 25] are applicable. Independent of the integrated technique, we can extract topic trends within the set SAQ. These topic trends are summarized as pairs of query and link sets. After the global classification of requests, locally evaluated results are propagated to support effective searching by all users. The identification of new topics with a high frequency is facilitated by a continuous update of SAQ and FAQ with all local feedback information.

Summarizing, with SAQ and FAQ, the aggregate transaction matrices can be used to identify special topics, as well as communities of interests. This processing is advantageous to confirm relevant results for an information need, even if only a small community validates this result. For this reason, all SAQ elements represent very specific topics. SAQ as well as FAQ are computed by the union of all local transaction matrices. The classification process is based on all summarized associations, and no correlation with the corresponding users is considered. Hence, we propose an automatic community advertisement, in order to group users with similar interests.

4.3 Community Advertisement

In the World Wide Web, many approaches exist supporting Web communities. At this point, a content-based grouping of documents has been developed. Our approach, ISKODOR, provides a community-based support for Frequently Asked Queries, as well as all users interested in these grouped topics. Not all topic trends automatically define a community. Hence, further analysis of topic trends is performed, in order to calculate the *degree of commitment* and the *degree of heterogeneity* in a potential community. With each feedback assessment, a user gives a commitment to a special topic. Only a high degree of commitment facilitates the formation of a community. Furthermore, the *degree of heterogeneity* represents the distribution of all assessments. It is desirable to have a homogeneous distribution, in order to have a popular agreement on the same documents. Otherwise, a high heterogeneity represents controversial rankings of documents within a group of users. In this case, a new clustering of documents must be achieved to minimize the heterogeneity.

If both degrees exceed a certain threshold, the topic trend would be useful for the generation of a community. For the description of the major point of interest, a representative is chosen for each element of FAQ. Currently, we use for the representation the first fundamental associations describing the topic trend. Hence, a *Virtual Knowledge Community* [22]

[a] http.//www.sun.com

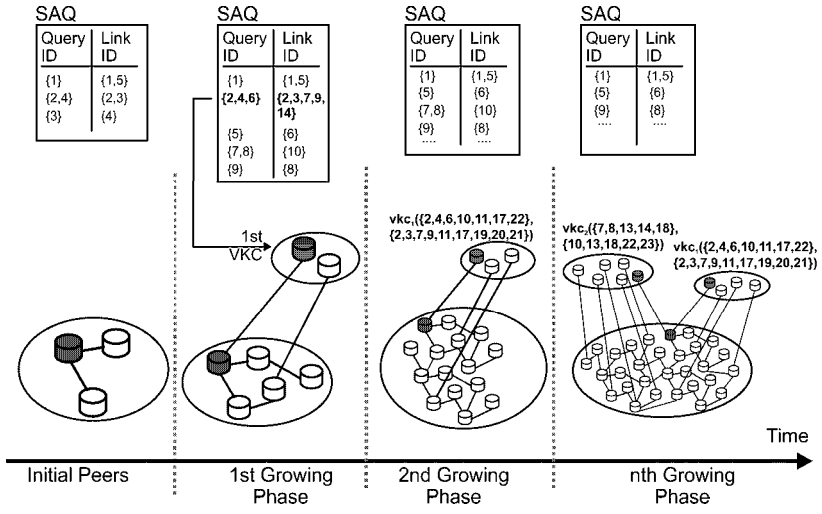

Fig. 8. Growing of Virtual Knowledge Communities

represents a group of users that share common interests, and can be defined on this basis.

Definition 5 *A Virtual Knowledge Community is a 2-tuple $VKC(V, f)$, which is formed by a set of users $V \subseteq U$ and a representative $f \in FAQ$.*

Due to continuous update of SAQ and FAQ, Virtual Knowledge Communities increase their number of users and their associations. As depicted in Figure 8, the SAQ table summarizes query ID's and link ID's for each phase. As time continues, the SAQ table grows with more links and queries, but also the number of users joining the virtual network increases. If a certain number of queries and links is transcended, a topic trend can be identified. In the given scenario in figure 8, a representative $vkc_1 \in FAQ$ is computed in the First Growing Phase, after the virtual network has been initialized. The user group is built afterwards, which contain users that have previously pushed the query-link associations to the SAQ table. The generation of this new user group will then be introduced to all interested users. Once a FAQ is created, the particular representative is removed from the SAQ ($SAQ \setminus vkc_x$).

Summarizing, Virtual Knowledge Communities group different formulations of comparable information needs by viewing an entire user group. Over *QCloud* alternative queries can be reflected, and over *LCloud* well evaluated links are collected. Both functions detect synonymous descriptions. Otherwise, if a Virtual Knowledge Community is found for a current information need, a high quantity of relevant results to a common topic can be result from this group. Aspects of fuzziness and uncertainty in a group of members are still missing, which leads to the definition of the final pillar: socialization.

5 Third Pillar: Socialization

Our idea of relations between users, topics, and documents has been introduced in Section 4. As we have seen, the quality of search results can be significantly improved, if people are grouped according to common interests. The same is done in community support systems

that were derived from the computer supported cooperative work (CSCW) research [29]. In order to form groups, it is essential to be aware of other group members', presence within the community, interests, and work habits [19]. Besides sharing information, one of the main aspects is to offer awareness features. In community systems (i.e. [32]) group processes are described as self-organized work, and are supported by traditional groupware techniques. For instance, the community toolbar, developed in the WiKo project [24], concentrates on the idea of supporting Web communities. People, who share a common interest can participate in a group. The main function is to support sharing, recommending, and annotating Web links within the community. There is very little technical support regarding the ranking of links. Hence, in our system, we propose a personalized filtering (cf. Section 5.2) of all local user associations, in order to be aware of new associations from other users.

Another quality of virtual communities is their fluidity [52]. Normally, such communities are not strictly limited. Instead, there are members, who can be seen as regular users, and others who only participate sporadically. Furthermore, group participation changes over time, as the interests of the members shift. Most of the groupware-driven approaches, so far, only allow for explicit joining in a group. Therefore, the fluidity and the status of other members, has to be identified by the users themselves according to their participation. We believe that the concepts of individualized and collaborative retrieval, as described in the Sections 3 and 4, can be a greatly enhanced for community systems. First approaches integrating fluidity of social factors and personalized user awareness are defined in the next sections.

5.1 Social Relationships

The concept described in Section 4 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 which would define a special group. Furthermore, a similar technique can be used to identify the relationships, and closeness among different members of one community. For instance, one would pay more attention to a rating by a close friend, than that of a foreigner or a "newbie" to the community. For these purposes, we define two degrees of social relationships:

- An *inter-type relationship degree* is defined between a user and a community. This degree is measured by the similarity of a user to the main topics of a community.
- An *intra-type relationship degree* is defined between two users, in order to measure the degree of similar interests.

For the definition of inter-type relationships, all users' search requests and their related sets of results ($SingleA$) can be compared to the different groups' $QCloud$, SAQ, and FAQ. Thus, in terms of a continuous evaluation each exchange of association between users is logged. Based on this (constant) analysis possible group memberships can be proposed automatically. This solves the problem of explicit group attendance as it is necessary in most community systems nowadays. This technique can also be used to find experts for a certain topic[b]. For these purposes users can be divided into *information consumers* or *providers*. In correlation

[b] The problem of finding experts is described in [2]. Here, a decision-tree-based approach is used to propose appropriate communication partners.

with Virtual Knowledge Communities a user can be an expert (ranked as information provider) for special topics and at the same time common users for other topics. This simplifies the interpretation of associations and a surplus value is generated for traditional Web ranking.

In traditional Information Filtering static objects are evaluated by users. However, no possibility exists for including the users in the evaluation. Due to heterogenous topics over all communities, special knowledge carriers can be identified. A first approach for a classification of users is the *user relevance* measure, in order to log the number of accepted recommendations of other users. Hence, we can compute an intra-type relationship degree by considering all users, which exchange information with each other. With the help of this procedure, users can evaluate others by collecting information on the local side. Thus users, which satisfied the information need of a user in the past, possess a higher relevance, contrary to these, from which a user could not attain satisfaction. The user relevance of user u for user v depends on the number of accepted results and an overall set of users U

$$UserRel_{u,v} = \frac{arc_v}{\sqrt{\sum_{w \in U} arc_w^2}}. \tag{5}$$

arc_v is the number of accepted results recommended by user v. The range of the user relevance measure is the interval $]0, 1]$. High values represent a high relevance of a user v for a user u. At runtime the user relevance is updated by the value of accepted recommendations. Summarizing, user relevance is a major part of the overall accuracy concept. As a basic principle we presuppose a "Network of Trust", as no abusive use occurs. In cases where the quality of a recommendation does not cope with the expectations of a user, the user relevance affects the ranking of results for future recommendations of a special user as described in the next section.

5.2 Community-based Filtering

Community-based Filtering summarizes all recommendations of other users. Due to the automatic requesting of users of the same Virtual Knowledge Community, a context-specific selection of users is the goal (cf. Section 6.2). Hence, an information flooding can be avoided, and a dynamic set of data is mapped to a static snapshot of fundamental associations for a set of users. This static set is used as a basis for an enhanced Collaborative Filtering approach, where all explicit and implicit relevance feedbacks, as well as all user relevances are considered.

Definition 6 *(Community-based Filtering) For a user $u \in U$ with memberships to the groups $C_i \subset VKC$, a prediction for a recommended document d is computed over all peers c of the group C_i*

$$cof_{u,d} = \overline{r_u} + \frac{\sum_{c \in C_i} (sim(u,c) + UserRel_{u,c})(r_{c,d} - \overline{r_c})}{\sum_{c \in C_i} (sim(u,c) + UserRel_{u,c})}$$

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 r. The similarity of two users can be computed with standard similarity measures [41]. Like Collaborative Filtering approaches, we use the Pearson correlation coefficient [40] by considering the implicit relevance feedbacks of two users u and v.

As described above, community systems that are based on groupware techniques, can be extended by Information Retrieval techniques in a meaningful way. On one hand, automatic support identifies relevant Web links, as well as finding colleagues with similar interests. This facilitates the support of self-organized groups. On the other hand, regarding communities being fuzzy and instable, these groups are missing in most current Information Retrieval approaches. The combination of these two approaches makes the World Wide Web more transparent. Summarizing, the Web search effectiveness can be significantly enhanced by this interdisciplinary approach.

6 Framework for Congenial Web Searching

According to Belkin and Croft [10], information needs can be distinguished between one-time goal, or long-term goals. Information Retrieval systems are presently concerned with a single uses of the system. However, Information Filtering addresses repeated uses of the same system. Congenial Web searching builds the conceptual basis for both types of information acquisition. With the local individualization strategy (Pillar 1), long-term information needs can be identified. For this reason, an integrated information service assisting Information Retrieval and Filtering is proposed based on a peer-to-peer (P2P) architecture [6]. A P2P architecture offers a transparent service, because no personal information of former search requests is stored at the central server. 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, collaboratively; and when information is provided, it becomes the passive part. A pull-push cycle [23] defines a cooperative exchange of information. Furthermore, each peer works with others for a common purpose, as proposed in the second pillar. A Collaborative Information Pull is initiated by a user query (cf. Section 6.1). Personalized User Awareness, proposed in Pillar 3, propagates (push) individually computed recommendations to each peer. Summarized in a Cooperative Information Push (cf. Section 6.2), the system automatically generates queries by exploiting stored ranking lists.

6.1 Collaborative Information Pull

For a Collaborative Information Pull, a user initiates a search request for a dynamic set of data. Unlike traditional retrieval processes, our approach facilitates a mutual engagement of all users to answer their specific needs. In a coordinated effort, documents from different sources are merged for a Collaborative Information Pull. For this task, we distinguish three types of search engines: (1) Web search engine, (2) community search engine, and (3) local search engine. With the Web search engine, each user can choose his favorite system. The retrieval in a community of users has an important impact on the collaborative approach. As depicted in Figure 9, the users' information need, formulated as a query, is requested in

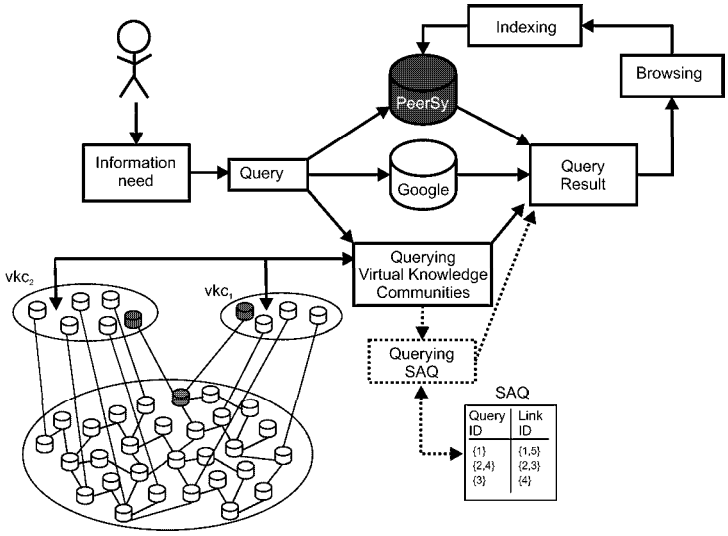

Fig. 9. Collaborative Information Pull

a distributed manner. All three search engines process this request. First, a common Web search engine (e.g. Google) retrieves all documents that match with that query. This system has a high-availability and a large index. We have no influence on this retrieval model, and all results are collected for the final processing by our system. The integration of a common Web search engine serves as an efficient preprocessing of information needs. Second, the community search engine considers two cases: (1) If the topics of a Virtual Knowledge Community match with the query, a group member gets validated results of others. Otherwise, the user has to join the group in order to get sufficient results. (2) If no Virtual Knowledge Community exists, the query is used to request the global list of Seldom Asked Queries (SAQ). A successful match with SAQ enhances the set, and a query trend begins to increase. At this point, the emerging of a query trend can lead to the creation of a new Virtual Knowledge Community. Finally, the local search engine retrieves the former relevant documents, if any exist. The results of all three collections are presented to the user according to his personalized ranking strategy. On this account, the user can browse through the result set, and if he finds relevant documents, his explicit feedback is stored. All assigned results are logged in a local database as a new fundamental association, which can be shared with other users.

6.2 Cooperative Information Push

The main goal of a Cooperative Information Push is an automatic recommendation of new documents. The user is inactive within this search process, and the system generates recommendations. Especially for long-time information needs, the user is informed about new relevant documents. All queries are generated automatically based on a user profile. This profile is computed with all local associations (cf. Pillar 1). For an automatic selection of relevant information, all peers are involved in a cooperative manner, and are responsible for a portion of the work. In order to be aware of new relevance feedback assessments in a com-

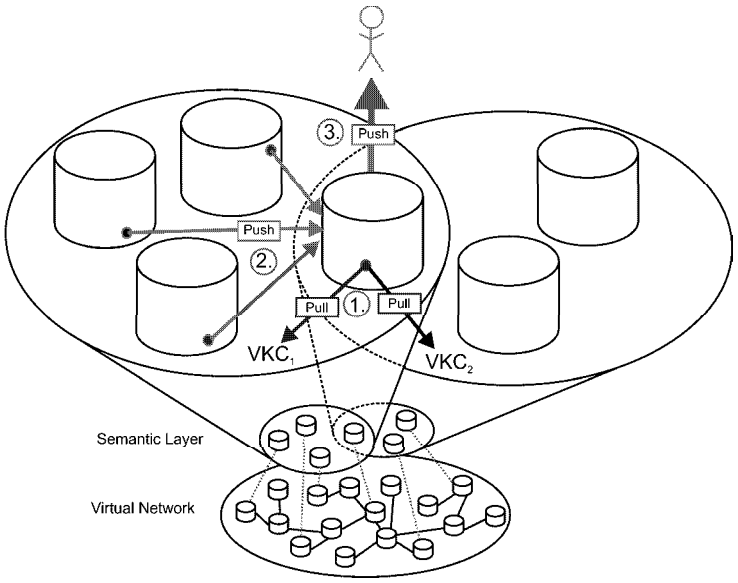

Fig. 10. Cooperative Information Push

munity, each peer is responsible for the collection of information. This cooperation between peers is activated by a trigger on each peer, which the user can individually configure. In terms of a cooperative pull-push cycle [23], first, a peer composes a set of query terms which describe long-term information needs. These requests are sent out in the peer-to-peer network, and other peers assume further processing. To gain sufficient qualitative results an information pull is initiated for the selected queries. As depicted in Figure 10, only peers of Virtual Knowledge Communities are requested in which the actual peer is a member. These communities facilitate a context-driven global prefiltering of peers. This preselection of peers mainly decreases the network load by considering affiliated peers on the semantic layer. The second step in the cooperative pull-push cycle is the processing of requests by all preselected peers. For this task, they send all new relevant information to the requesting peer. Finally, an information push for the user is generated with this information, and the peer undertakes the final weighting of suggested documents taking social relationships into account (cf. Section 5.1). Hence, with the pull-push cycle each community is always aware of new documents within the community.

7 ISKODOR - **Prototype for Congenial Web Searching**

ISKODOR implements a personalized access to information from the Web. Through suitable structuring and linkage of information, it facilitates Information Filtering, as well as Information Retrieval. Hence, the approaches of individualization, collaboration, and socialization are integrated in this platform. A first prototype of ISKODOR exists, and it is implemented with JXTA, the de-facto standard for peer-to-peer architectures. Due to a distributed data storage of all local associations, peer-to-peer systems are advantageous to avoid the single-point-of-failure problem. Furthermore, user objections regarding the central storage of personalized

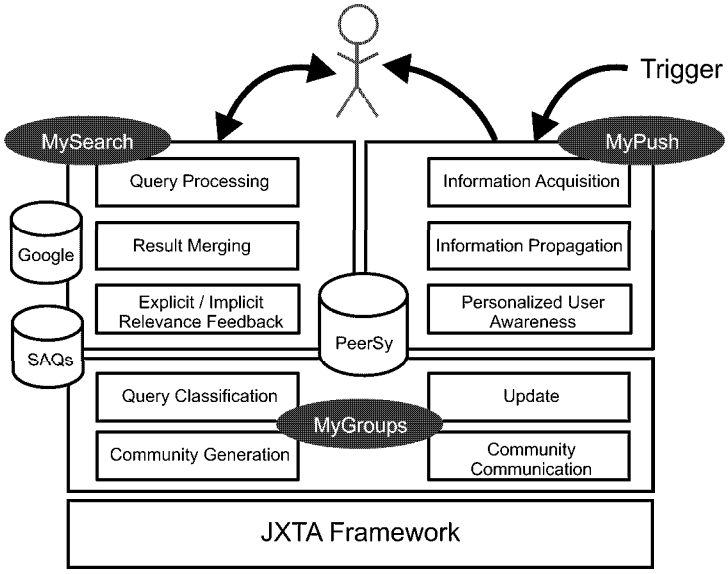

Fig. 11. ISKODOR system design

search information can be diminished.

In order to address a large number of users, the system is designed to be platform and browser independent. A graphical overview of all integral parts is depicted in Figure 11. The system is built of three central components: MySearch, MyGroups, MyPush (cf. Figure 11). The main part of the entire system is accomplished by the Peer Search Memory (PeerSy). In PeerSy all user queries are stored together with validated results. The storage of these query-result pairs is done in a local database. This facilitates the recovery, and the exchange of associations with other users. Besides these personalization issues, all functions for the collaborative exchange among peers are summarized with the component MyGroups. Based on JXTA, MyGroups handles the peer grouping in respect to all Frequently Asked Queries (FAQ), the peer communication, and all group updates. Further information about the technical details of our approach can be found in [23].

Figure 12 depicts the appropriated user interface for PeerSy. There are three tabs that representing the three mentioned components. First, the tab MySearch assists a Collaborative Information Pull. The search interface is arranged into four sections. The user can formulate his information need as a query in the left corner. Below, additional statistic information from previous searches and favorites are presented. The result is presented on the right side of the window. For the visualization of an intersection between known and unknown results, we choose a set view as depicted in Figure 12. Only the summary of all collections facilitates an optimal personalization. New results are aligned with former relevant documents. Below this presentation, detailed information for each result is presented. Second, the tab MyGroups gives an overview about all group memberships of a user, and all existing groups. Furthermore, this tab assists the administration of selected memberships.

Finally, the "MyPush" tab (cf. Figure 13) organizes recommendations in order to present

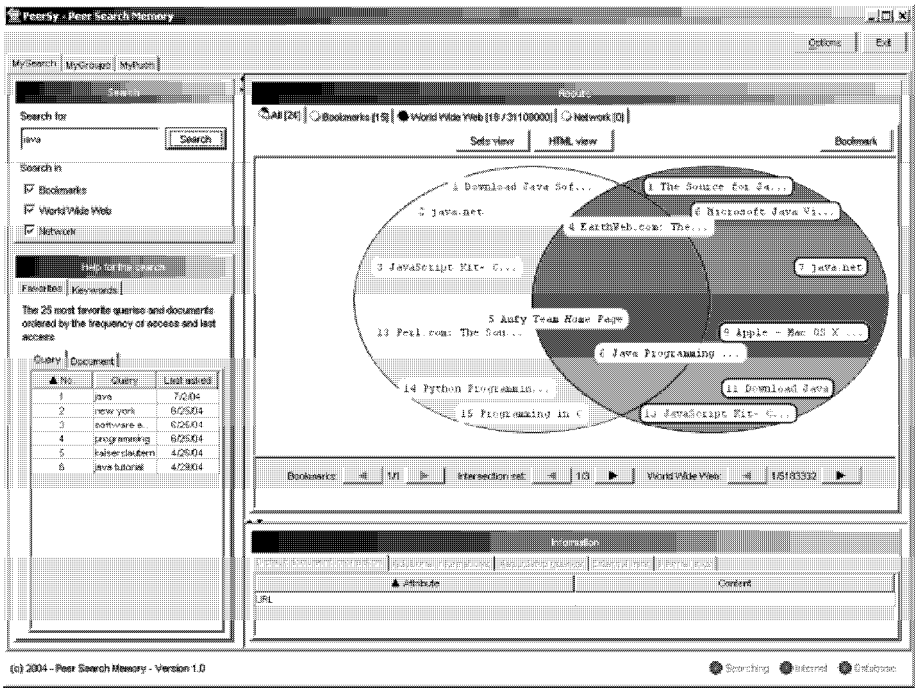

Fig. 12. User interface for MySearch

all consumed and provided information. To achieve a maximum of system transparency each user is aware of all relayed information to other peers. Furthermore, administration settings like connection type, peer name, and peer trustability can be set with this interface. All recommendations are updated automatically by a peer trigger, or manually by the user pressing the button "Manual Activation".

8 Evaluation

The effectiveness of our system is affected by several parameters. In comparison to traditional retrieval systems, no existing test collection like TREC can be used in our context, because they lack the user's feedback for relevant results of their queries. In order to get this information, a large user study would be necessary. In this paper, we present a first evaluation setting based on clickthrough data. For this task, we used one-year user logs from a German campus Web proxy. From these logs, we extracted 86,969 user query sessions to a Web search engine. In our evaluation setting, we use Google, because it was the favorite search engine in 75% of all query sessions. In one year, 278 users requested 29,869 distinct queries and viewed 73,549 distinct documents. These documents represent our explicit feedback, because no real user ratings are available. This is a limitation of our evaluation, although, several irrelevant results are already discarded by the users.

Averagely, 2.91 distinct documents have been viewed for a query. The maximum number of different documents viewed for a query was 238. We conjectured that a large number of queries are very specific, and the users expect one or two high-quality results on the first page.

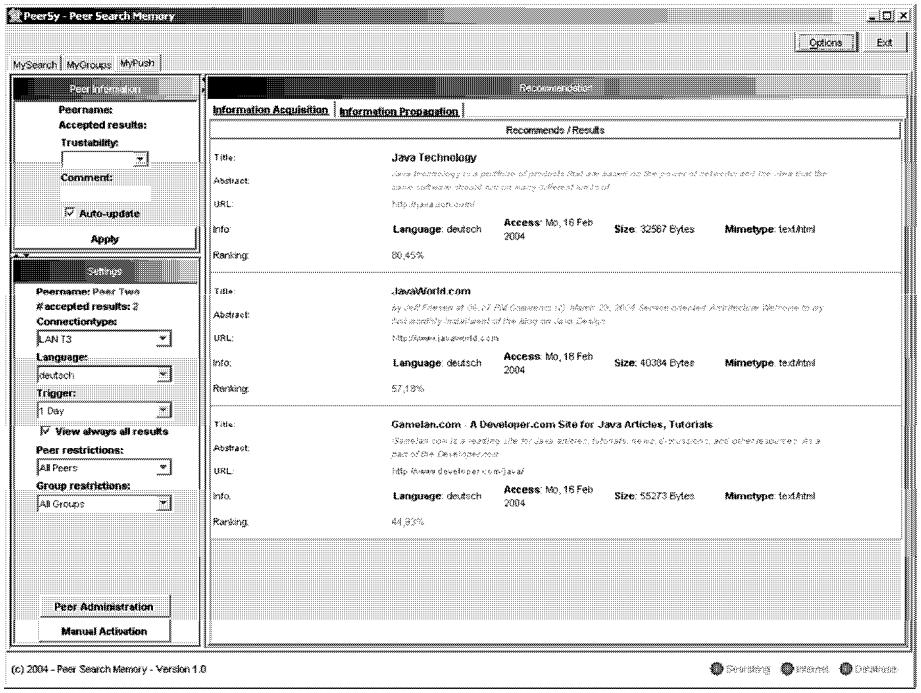

Fig. 13. User interface for MyPush

Indeed, our data shows that for 69.4% of all queries only one or two documents have been viewed. Furthermore, this observation can be seen in correlation to the query length.

Figure 14 illustrates the distribution of query lengths in terms of the number of words. We notice that 25% of the queries contain only one keyword, and 38% of the queries contain two keywords. The average length of all queries is 2.34. The distribution of query length is similar to those reported by others: an analysis of major web search engines revealed that on the average any query contained 2.21 terms according to [27], and 2.35 according to [45].

Due to the user-centered design of our system, we have to analyze the search behavior of each single user in detail, in order to measure the improvement of search effectiveness. Figure 15 depicts the distribution of queries and links for each user. On one hand, a small number of users requested many queries. On the other, many users submitted only a small number of queries. The main strength of our system is the community support, and we want to measure the effectiveness for queries of this kind.

In a first step, we compare our average precision at different *user support levels*. A user support level is defined by the number of users, who viewed the same documents for an aggregated set of queries. Hence, a support of 0.5 describes that at least 50% of all users agreed in a result of the Web search engine. For example, the most frequently asked query in our data set is `ksk ahrweiler`.[c] A total of 64 users formulated this request. For all queries viewed by more than one user, we compute the user support value. As a result we

[c] `ksk` is the abbreviation of a German credit institute (Kreissparkasse), and `ahrweiler` is the name of a city

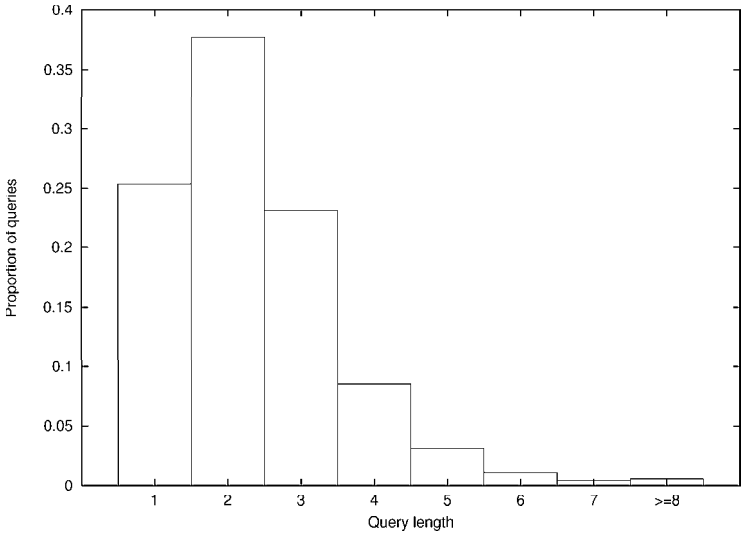

Fig. 14. Distribution of query length

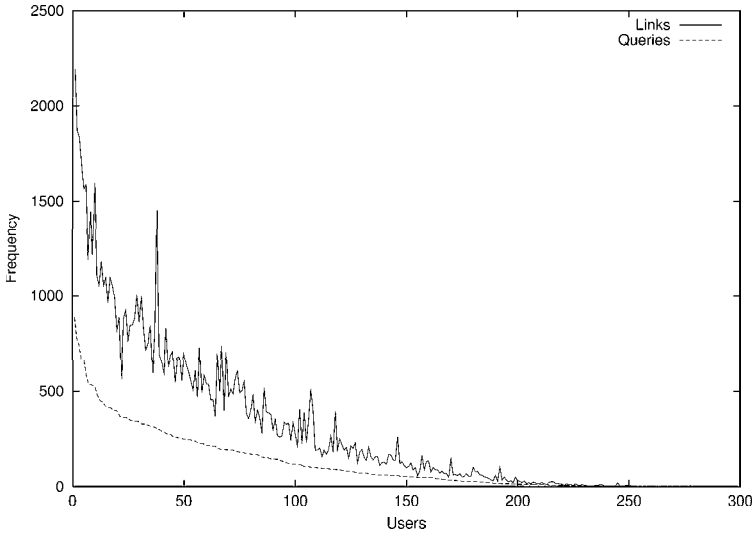

Fig. 15. Distribution of queries and links for each user

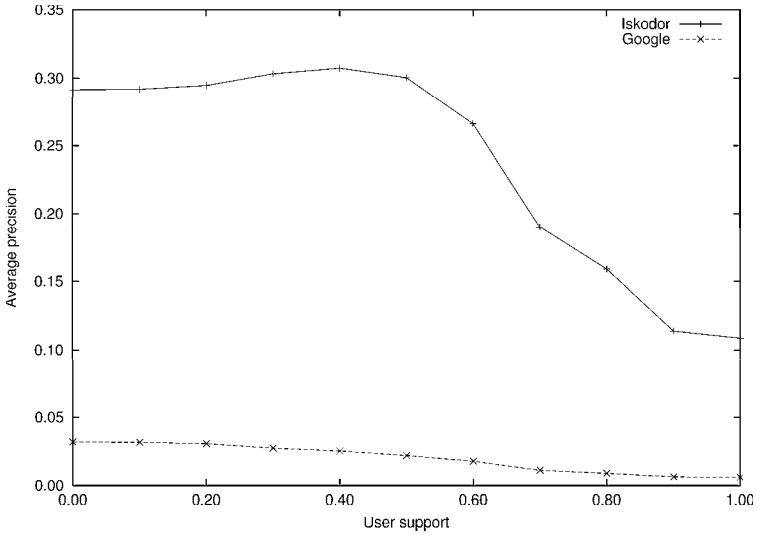

Fig. 16. Average precision for different user support levels

retrieve only two documents with our system: (1) http://www.ksk-ahrweiler.de (with user support 1.0), and (2) http://www.kreissparkasse-ahrweiler.de (with user support 0.5). In comparison, Google retrieves 1,290 results in total. In order to get a better system precision, our system decreases the number of totally retrieved documents to get a smaller denominator. We compute the precision of our system with the traditional formula [41]. All documents, which have been commonly found by Google and our system are selected as a pool of relevant documents for a query. Because user studies showed that most users only view the first result page [26], we compare our result set only with the first 20 Google results – despite the fact that Google retrieves much more than 20 results, i.e. more than one result page. Figure 16 illustrates our results for 867 queries, and shows a significant improvement of the average precision at all support levels. This result is obvious for specific queries like `ksk ahrweiler`, where each user has a special expectation of the result.

With increasing support level the average precision decreases with both systems. This can be explained with two observations. First, we find fewer common documents with both search engines at larger support values. This is a sign for the missing individualization of Web search engines, which can not deal with local search constraints of users. With our system, all users are assisted, who repeat the query `ksk`. Due to previous queries with the context of a special city, our system has a better precision for such queries with a regional focus. Second, we detected a special kind of queries, which have been requested by different user groups, but none of the viewed documents can be found on the first 20 Google results. For this kind of queries, we get a poor precision only because they can not be found in the pool of relevant documents. In reality, these queries do not have a poor precision. The information need of these queries is very broad, and it can be matched with several distinct contexts. Depending on the user support level, we can classify these queries in our data set. For example, `value based management` is such a query, which was requested by 10 users. In total, 72 distinct

documents have been viewed by the users, and only 4 documents could be found on the first result page of Google (rank 5, 6, 7, and 10). All these documents have a user support of 0.2, and we can compute a maximum user support of 0.4 for our first ranking position. On one hand, all queries with a poor precision show the limitation of our evaluation setting, because of the missing explicit user feedback. On the other, it is a proof that the usage of clickthrough data is not enough to increase the effectiveness for all queries.

Figure 17 and 18 show the dependency of the support value and the precision of a query. For example, we tested our system with a user support value of 0.4 and 0.8. By increasing support value, the number of viewed documents is restricted, and a smaller set of documents is in the pool of relevant documents. We find only a small set of high precision queries (18%) with a support value of 0.8. Instead, a lower support value increases the set of relevant documents, but the average number of high-precision queries also decreases. We reason that our system can enhance the effectiveness for 42.7% of all queries with a support value of 0.4, significantly. Furthermore, we conjecture that for these queries users do not expect a high recall. We plan to further evaluate whether the results can be proven with a larger number of users.

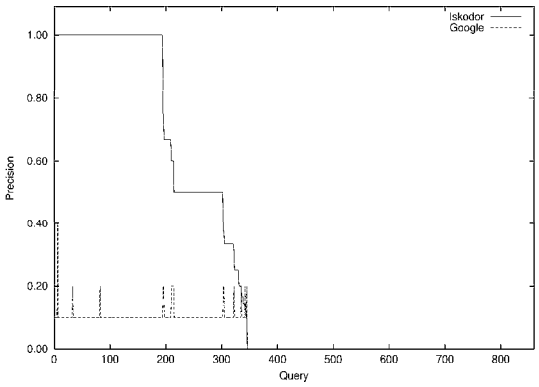

Fig. 17. Precision with user support 0.4

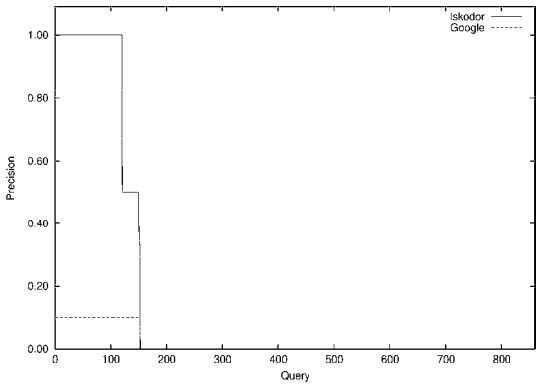

Fig. 18. Precision with user support 0.8

Due to the correlation between a growing support value and a decreasing precision, we detect the need of user-centered effectiveness measures. Such measures should reflect the individual user satisfaction for single queries as well as the user reputation within the network. Both measures can be maximized by the exchange of ratings during retrieval and filtering. Our system fulfils the demand to assist both kinds of queries: (1) specific queries requested by nearly 60% of all users, and (2) broad queries with a nonspecific description of the user need. Also, with our local individualization strategy a user can be assisted even if no community of interest can be found for his search context. For these queries, our system assists the user with a personalized ranking by the Peer Search Memory. If the attention in such topics by other users grows, the SAQ list groups these concepts, and a Virtual Knowledge Community might be established. As future evaluation, we also plan to evaluate our community-based filtering approach based on bibliographic data, which is used as explicit feedback.

9 Related Peer-to-Peer research

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, retrieval methods for peer-to-peer systems are still at their infancy. Intelligent routing strategies are necessary to avoid a high network load. Many of the most effective 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 [34, 35], 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 [48, 47] include systems that can be characterized by unique identification keys. These 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). DHTs can be distributed over several nodes within a network. A concrete implementation of such a DHT is realized in the Content-Addressable Network (CAN) model by Ratnasamy et al. [39]. 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.

YouSearch [8] is a distributed (peer-to-peer) search application for personal Web servers operating within a shared context. 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. In comparison to ISKODOR, YouSearch does not provide bookmark-sharing, but more or less file-sharing which is supported by search mechanisms. Another difference is that groups are only formed via manual user actions, and the system does not conduct any proposals to support and enhance this process, i.e. it does not directly approach the relevant peers to recommend a group creation [9].

Tryfonopoulus and Koubarakis [49] 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 the clients. A generic architecture for a P2P-IR system is proposed by Arberer et al. [5]. 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.

Another strategy assisting an efficient query routing is proposed by Bender et al. [11]. They propose a bookmark-driven approach for Web search. 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.

In recent P2P search research, different approaches for efficient routing strategies are already available. Due to the usage of JXTA, efficient routing and retrieval algorithms are supported by our system. In our work, we focus our research on the enhancement of effectiveness in such networks. The design of our retrieval models is logically independent of the infrastructure. We chose a peer-to-peer architecture to guarantee user anonymity and system transparency. Another advantage of peer-to-peer networks is the facility to build so-called peer groups, which allows groups of peers to restrict the access to distinct data to authorized peers only.

10 Conclusion and Outlook

In this paper, we presented our approach for Congenial Web Searching. With regard to the need for improving effectiveness of Web search, a new Web system is proposed, based on local high-quality associations between information needs, and results. The concept is motivated by three conceptual pillars, which are: individualization, collaboration, and socialization. Personalizing search engines seems to be a hopeful way to optimize retrieval within huge data sets, and taking into account individual interests and preferences. The first step, and the basis for our system, is the storing of semantic associations between queries and results, in terms of a personalized ranking matrix. To profit from the very large number of Web users, transaction matrices can be merged. So, new search queries are proposed by ranked results from other users. Furthermore, queries and assigned results can be grouped. Thus, our last step is bringing together people with shared interests, which can be directed by their queries. In our intra-disciplinary approach, we combine the idea of Collaborative Information Retrieval on the one hand, and the concept of knowledge communities on the other. First of all, the grouping of queries, and the automatic classification of common interests may help to automatically classify experts. Secondly, traditional approaches in Information Retrieval, which deal with the grouping of topics and users may be too strict. Regarding the fuzziness of communities according to their membership, and their shift of interests over time, the merging process is enhanced with our system. This last part of the Congenial Web Search approach is part of our ongoing work. Nevertheless, our approach was implemented in the ISKODOR prototype, which is based on a highly flexible peer-to-peer architecture. In the future, we plan to evaluate the whole prototype, and the interaction between the single concepts.

As we have shown, supporting social interaction and community building has a crucial impact on the effectiveness of the retrieval techniques presented here. One of the main concepts we pointed out in this paper was the need for an integration of community software tools and retrieval systems. This integration can be seen in two ways. First, a search request can result in documents from the Web as well as experts in the social network. Second, action trails in a community may lead to a description of "context" of use. Those ideas are part of our current work. In another project, we are developing a community support system, which is based on concepts of context sensitive intelligence. The underlying infrastructure will have a

highly adaptive behavior according to changing user needs. A future goal is to integrate this community platform and our framework for Congenial Web Search.

References

1. D. Abrams, R. Baecker, and M. H. Chignell. Information archiving with bookmarks: Personal web space construction and organization. In *Proc. of Human Factors in Computing Systems (CHI)*, 1998.
2. M. Ackerman and D. McDonald. Answer garden 2: Merging organizational memory with collaborative help. In *Proc. of the ACM Conference on Computer Supported Cooperative Work (CSCW'96)*, pages 97–105. ACM Press, 1996.
3. R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. In *Proc. of the ACM SIGMOD International Conference on Management of Data*, pages 207–216, 1993.
4. R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proc. of the 20th Int. Conference of Very Large Data Bases, VLDB*, 1994.
5. K. Arberer, F. Klemm, M. Rajman, and J. Wu. An architecture for peer-to-peer information retrieval. In DPIR04 [20].
6. D. Barkai, editor. *Peer-to-Peer Computing. Technologies for sharing and collaboration on the Net.* Intel Press, 2002.
7. M. Bates. After the dot-bomb: Getting web information retrieval right this time. *First Monday*, 7(7), 2002.
8. M. Bawa, 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*, pages 577–586. ACM Press, 2003.
9. M. Bawa, G. Manku, and P. Raghavan. SETS: Search enhanced by topic segmentation. In *Proceedings of the 26th annual international conference on Research and development in information retrieval*, 2003.
10. N. J. Belkin and W. B. Croft. Information filtering and information retrieval: two sides of the same coin? *Communications of the ACM*, 35(12):29–38, 1992.
11. M. Bender, S. Michel, G. Weikum, and C. Zimmer. Bookmark-driven query routing in peer-to-peer web search. In DPIR04 [20].
12. M. Berry, S. Dumais, and G. O'Brien. Using linear algebra for intelligent information retrieval. *SIAM Review*, 37(4):573–595, 1992.
13. K. Bharat. Searchpad: Explicit capture of search context to support web search. In *Proceedings of the 9th international conference on World Wide Web*. ACM Press, 2000.
14. B. Brookes. Theory of the bradford law. *Journal of Documentation*, 33(3):180–209, 1977.
15. V. Bush. As we may think. *The Atlantic Monthly*, 176(1):101–108, July 1945.
16. S. Chakrabarti, 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.
17. A. Cockburn, S. Greenberg, B. McKenzie, M. Smith, and S. Kaasten. Webview: A graphical aid for revisiting web pages. In *Proceedings of the OZCHI'99 Australian Conference on Human Computer Interaction*, 1999.
18. R. Cooley, B. Mobasher, and J. Srivastava. Web mining: Information and pattern discovery on the world wide web. In *Proc. of the IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97)*, 1997.
19. P. Dourish and V. Bellotti. Awareness and coordination in shared workspaces. In *Proc. of the ACM Conference on Computer Supported Cooperative Work (CSCW'92)*, pages 107–114. ACM Press, 1992.
20. 2004.

21. M. Eirinaki and M. Vazirgiannis. Web mining for web personalization. In *Proceedings of the ACM Transactions on Internet Technology*, volume 3, pages 1–27, February 2003.
22. M. Gnasa, 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.
23. M. Gnasa, N. Gül, J. Grigull, S. Alda, 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*, 2004.
24. W. Gräther, K. Klöckner, and S. Kolvenbach. Community support and awareness enhancements for cooperative knowledge generation. In *Proc. of the 29th Euromicro Conference (EUROMICRO'03).* ACM Press, 2003.
25. T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual ACM Conference on Research and Development in Information Retrieval*, 1999.
26. B. J. Jansen and A. Spink. An analysis of web documents retrieved and viewed. In *Proc. of the 2003 Internet Computing Conference*, Las Vegas, USA, 2003.
27. B. J. Jansen, 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.
28. T. Joachims. Optimizing search engines using clickthrough data. In *Proc. of the ACM Conference on Knowledge Discovery and Data Mining (KDD).* ACM, 2002.
29. R. Johansen, editor. *Groupware: Computer Support for Business Teams.* Freepress, New York, 1988.
30. W. P. Jones, H. Bruce, and S. T. Dumais. Keeping found things found on the web. In *Proc. of the ACM Conference on Information and Knowledge Management (CIKM)*, 2001.
31. Y. Khopkar, A. Spink, C. L. Giles, P. Shah, and S. Debnath. Search engine personalization: An exploratory study. *First Monday*, 8(7), 2003.
32. M. Koch. Community-unterstützungssysteme - architektur und interoperabilität. Technical report, Institut fr Informatik, Technische Universität München, 2003.
33. S. Lawrence. Context in web search. *IEEE Data Engineering Bulletin*, 23(3):25–32, 2000.
34. A. Loser, 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.
35. J. Lu and J. Callan. Content-based retrieval in hybrid peer-to-peer networks. In *Proceedings of 12th International Conference on Information and Knowledge Management*, 2003.
36. M. Mulvenna, S. Anand, and A. Buchner. Personalization on the net using web mining. *Communications of the ACM*, 43(8):122–125, 2000.
37. D. Nichols. Implicit rating and filtering. In *Proceedings of 5th DELOS Workshop on Filtering and Collaborative Filtering*, 1998.
38. D. Oard and J. Kim. Implicit feedback for recommender systems. In *Proceedings of the AAAI Workshop on Recommender Systems, July 1998.*, 1998.
39. S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A Scalable Content Addressable Network. In *Proceedings of ACM SIGCOMM*, 2001.
40. P. Resnick, 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.
41. C. J. V. Rijsberg. *Information Retrieval.* Butterworths, 1979.
42. J. Rocchio. Relevance feedback in information retrieval. In G. Salton, editor, *The SMART Retrieval System: Experiments in Automatic Document Processing*, pages 313–323. Prentice-Hall, Englewood Cliffs, NJ, United States, 1971.
43. G. Salton and C. Buckley. Improving retrieval performance by relevance feedback. *Journal of American Society for Information Sciences*, 41(4):288–297, 1990.
44. G. Salton and M. McGill. *An Introduction to Modern Information Retrieval.* McGraw-Hill, 1983.

45. C. Silverstein, M. Henzinger, H. Marias, and M. Moricz. Analysis of a very large web search engine query log. *SIGIR Forum*, 33(1):6–12, 1999.
46. J. Srivastava, 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.
47. C. Tang 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.
48. C. Tang, 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.
49. C. Tryfonopoulos, 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.
50. J.-R. Wen, J.-Y. Nie, and H.-J. Zhang. Query clustering using user logs. *ACM Transactions on Information Systems*, 20(1):59–81, 2002.
51. T. Wilson. On user studies and information needs. *Journal of Documentation*, 37(1):3–15, 1981.
52. M. Won and V. Pipek. Sharing knowledge on knowledge - the exact peripheral expertise awareness system. *J.UCS - Journal of Universal Computer Science*, 9(12):1388–1397, 2004.
53. O. Zamir and O. Etzioni. Web document clustering: a feasibility demonstration. In *Proc. of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, 1998.