User Intent Transition for Explicit Collaborative Search Through Groups Recommendation

Juan M. Fernández-Luna
Departamento de Ciencias de la Computación e Inteligencia Artificial, CITIC-UGR.
Universidad de Granada, 18071 Granada, Spain
jmfluna@decsai.ugr.es

Juan F. Huete
Departamento de Ciencias de la Computación e Inteligencia Artificial, CITIC-UGR.
Universidad de Granada, 18071 Granada, Spain
jhg@decsai.ugr.es

Julio C. Rodríguez-Cano
Centro de Desarrollo Territorial Holguín. Universidad de las Ciencias Informáticas, 80100 Holguín, Cuba
jcrcano@uci.cu

ABSTRACT

Collaborative information retrieval is an emerging research field in charge of establishing techniques and methods to satisfy the shared information needs of groups of people that work together as a team, starting from the extension of the information seeking and retrieval process with the knowledge about the queries, the context, and the explicit collaboration habits among them. Unfortunately, in broad online communities that besides grow continuously (e.g., social networks, e-learning systems, or peer-to-peer networks) can be difficult the conformation of these groups, or all benefits can not be obtained, for the lack of transparency among users' seeking tasks in distributed environments. To address this issue, we propose in this work a recommender agent based on latent semantic indexing formalism to assist the users that search alone to find and join to groups with similar information needs. With this mechanism, a user can change easily her solo search intent to explicit collaborative search. We assume as hypothesis that both, the group and the new member will benefit. To validate our hypothesis we have designed an experiment with twelve groups of students in the context of search-driven software development.

Categories and Subject Descriptors

H.5.3 [**Information Interfaces and presentation (e.g., HCI)**]: Group and Organization Interfaces; H.3.3 [**Information Storage and Retrieval**]: Search Process.

General Terms

Design, Human Factors

Keywords

collaborative information retrieval, collaborative search, source code search, search driven software development.

CIR'11, October 28, 2011, Glasgow, Scotland, UK.

1. INTRODUCTION

"Multiple realities inform each other, fertilize, stimulate, and stir the cauldron of creativity."
– David La Chapelle

Collaboration is an important social aspect in modern information seeking and retrieval scenarios, but commonly it has only been seen from an asynchronous and implicit perspective, such as within the recommender systems, in particular Collaborative Filtering. As a complement, many scenarios requiring explicit collaborative search techniques also call for people to work together at the same time to satisfy shared information needs. Such collaborations are difficult to achieve with existing digital tools, often resulting in high overhead such as undesired redundancy of effort [10]. To address this issue, the community has adopted Collaborative Information Retrieval (CIR) as an emerging research field [3, 7].

In CIR scenarios, people who are requesting relevant documents for their common needs could take advantage grouping through themselves and working together, supported by a wide range of tools. But how individual users could know other users or groups with the same interests? In other words, how users could be grouped in homogeneous groups? The simplest solution is when there is previous and global knowledge about the information needs of each user, and these are naturally grouped in working groups. But when this information is not explicit, CIR systems should try to integrate users working alone in the best group.

For the task described above, one alternative could be to show to the isolated users any kind of information about the interests of the groups which are already established and, according to that, the users make a decision about which group they will be joined, probably using an explicit group search as for example Mendeley[1]. A second possibility would be to leave this action to the system, so it automatically would detect common interests analyzing users and groups' search sessions, and suggests possible groups who would be the most appropriate for isolated users. This second approach will be explored in this paper.

Particularly, our research addresses the question: How isolated users in distributed environment can be assisted in order to find groups with shared information needs during the transition from implicit to explicit collaborative search intent?

[1]http://www.mendeley.com/groups/

With this question in mind, in this paper a user intent transition technique based on Latent Semantic Analysis (LSA) [16] is presented, in order to recommend the most interesting groups to users, according to data obtained from the search sessions. This is the main contribution of this paper. As far as we know, in the state-of-the-art there is no proposed mechanism to suggest groups to users in a search session. With the aim of testing the effectiveness of our approach, we have conducted a formative study that evaluates the performance of those groups that increment their size with users presenting similar information needs, under a context of collaborative search.

This paper is organized as follows: In Section 2 we describe the basics of CIR and LSA. The details of our contribution are given in Section 3 and the empirical study is presented in Section 4. The paper is concluded in Section 5 with implications and futures research directions.

2. BACKGROUND

This section is intended to help the reader to understand the rest of the paper.

2.1 Collaborative Information Retrieval

Traditionally, researchers on Information Retrieval (IR) have been considered the search of relevant information as an individual task. However, with the computer networks participation era and ubiquitous computing arrivals, some of them have identified many contexts where the users (e.g., friends, colleagues, classmates, developers) collaborate actively during the search process, such as exploratory Web search, health-care teams, digital libraries, learning environments, search-driven software development, etc. [12].

Current operative IR systems do not have support for explicit collaboration among users with shared information needs (except a few of them, e.g. SearchTogether [12], Coagmento [15], or COSME [2], but those address only some facets associated to CIR). The community working on this gap has adopted CIR as an emerging research field in charge of establishing techniques and methods to satisfy the shared information needs of groups of people that work together as a team, starting from the extension of the information seeking and retrieval process with the knowledge about the queries, the context, and the explicit collaboration habits among them.

Collaboration during search tasks can be classified along four dimensions [6]: *intent*, *mediation*, *concurrency*, and *location*. In terms of *intent*, collaborative search can be implicit or explicit. Implicit collaboration encompasses recommender systems, such as e-business applications, in which the behaviour of people searching for particular content is used to inform the search results of others searching for similar content. Explicit collaboration, on the other hand, occurs when people search for completing a joint task, such as travel planning. Depth of *mediation* is the level at which collaboration occurs in the user interface versus search engine back-end. Additionally, collaborative search tasks can be synchronous or asynchronous (*concurrency*), and co-located or distributed (*location*).

Another important contribution in CIR is the identification of three fundamental features for CIR systems that can enhance the users' benefits [15, 11, 13, 4]: (i) *awareness*, (ii) *division of labor*, and (iii) *sharing of knowledge* (some authors consider persistence mechanisms as another key feature, but in our point of view, persistence can be included in the features aforementioned). We consider as another important feature (iv) *user intent transition* from implicit to explicit collaboration.

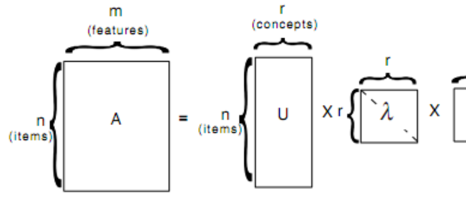

Figure 1: Illustrating the basic Singular Value Decomposition Theorem [14].

User intent transition is an important aspect when exist users searching alone in online communities, such as social networks, e-learning systems, wikis, or peer-to-peer networks, where groups of people interact actively to share common goals, traits, and interests. *User intent transition* help users to find groups with shared information needs across personalization technique like in recommender system, facilitating the transition from implicit to explicit intent and group association.

2.2 Latent Semantic Analysis

Latent Semantic Analysis [16] is an indexing and retrieval method that uses a mathematical technique called singular value decomposition (SVD) [8] to identify patterns in the relationships between the terms and concepts contained in an unstructured collection of text. LSA is based on the principle that words that are used in the same contexts tend to have similar meanings. A key feature of LSA is its ability to extract the conceptual content of a body of text by establishing associations between those terms that occur in similar contexts.

LSA is based on vector space model (VSM) [9], an approach using linear algebra for effective automated information retrieval. LSA is considered a truncated VSM that represent types and documents in a particular high–dimensional type-document vector space. The truncated VSM uncovers the underlying or latent semantic structure in the pattern of type usage to define documents in a collection [16].

Many research papers have been presented on LSA by data mining and information retrieval communities [14]. These works prove that exploiting latent semantics analysis can greatly improve efficiency and performance of applications, specifically for information indexing and searching.

More specifically, LSA uses common linear algebra techniques to learn the conceptual correlations in a collection of text. In general, the process involves constructing a weighted term-document matrix, performing a SVD on the matrix, and using it to identify the concepts contained in the text. LSA begins by constructing a term-document matrix, A, to identify the occurrences of the m unique terms within a collection of n documents. In a term-document matrix, each term is represented by a row, and each document is represented by a column, with each matrix element, a_{ij}, initially representing the number of times the associated term appears in the indicated document, tf_{ij}. This matrix is usually very large and sparse.

Once a term-document matrix is built, local and global weighting functions can be applied to it to condition the data. The weighting functions transform each cell, a_{ij} of A, to be the product of a local term weight, l_{ij}, which describes the relative frequency of a term in a document, and a global weight, g_i, which describes the relative frequency of the term within the entire collection of documents.

In our case, we use log-entropy weighting functions (some authors found in [16] that log-entropy gave the best retrieval results). In other words, each entry a_{ij} of A is computed as:

$$a_{ij} = g_i \ln(tf_{ij} + 1), \tag{1}$$

$$\text{with } g_i = 1 + \sum_j \frac{p_{ij} \log_2(p_{ij})}{\log_2 n}, \text{ where } p_{ij} = \frac{tf_{ij}}{gf_i}. \tag{2}$$

3. GROUPS RECOMMENDATION

In CIR, the association of individuals to groups may be done in an explicit manner. One way is when various users can agree to make a search a priori, for example, because their interests are similar (a research team at a school). Or an alternative, when the responsible of a group of workers could create working teams, each of them in charge of a different task. But, this explicit group creation is not always effective, because it presupposes that users know themselves in advance.

Once various users are explicitly or implicitly associated they can create a collaborative search session (CoSS) and work together to solve shared search interests. Meanwhile, users working alone maintain individual search sessions (InSS). In this paper we will explore a different alternative in which several users can be joined with a group to make searches. In our approach the search agent must be able to suggest the most appropriate group to each user[2]. In order to perform these recommendations, we will use the information in those documents assessed as relevant by a user, or a group of them, in previous searches, i.e. the search profiles.

With respect to the method, LSA is used to describe the groups' interests using abstract concepts (semantic domains), which are more descriptive than a raw term/document representation. As consequence, given a user, we transform her information needs (represented by a query q) into a concept-based representation and then, the closest groups in these domains will be the ones suggested to the users.

Following the notation presented in section 2.2, matrix A is formed considering the occurrences of terms in the set of documents selected in the CoSS of each group. Placing terms in rows and groups in columns, if a term T_i occurs in any of the relevant documents selected by group G_j, then $a_{ij} = 1$ and 0, otherwise. Finally, new values a_{ij} are computed applying the log-entropy weighting function from equation 1.

Applying the SVD algorithm, the matrices obtained are the following: U, associating terms and latent concepts; λ, the singular values matrix, containing the latent concepts; and finally, V, the matrix relating concepts and groups. Then, if T is the set of terms in the collection: G, the set of teams, which members are working collaboratively, and C, the set of latent semantic concepts, the sizes of matrices are $|T| \times |G|$ for A, $|T| \times |C|$ for U, $|C| \times |C|$ for λ, and $|C| \times |G|$ for V.

In order to suggest a group to a given user, we will require that the system might know the topics involved in the search. We consider that this requirement holds when the user had found at least two relevant documents according to her information need, expressed by a query q. Then the query is transformed from the term dimension to the concept dimension in the following way. The query vector is mapped into matrix U, obtaining a new vector q^C with size $|C|$. This new vector contains all the concepts in which each query term, q_i, is mapped in the new latent semantic space. The weight of component ${q^C}_j$ is the sum of the weights of all the terms to which the component is related to, i.e. ${q^C}_j = \sum_{i|q_i \in T} u_{ij}$

Once the query is represented in the new semantic space, the similarity of the query with respect to each group, represented by a vector extracted of its corresponding column in matrix V, is computed, also considering the singular values from matrix λ. Finally, the user is offered a ranking of groups sorted decreasingly by similarity. Then the user makes the decision about what team to join and from that moment work collaboratively with her new teammates.

In our case, we have employed the cosine similarity, so the function used to measure this value is the following:

$$Sim(q^C, gr_j) = \frac{\sum_{k=1}^{|C|} (\lambda_{kk} \times q_k^C) \times (\lambda_{kk} \times v_{kj})}{\sqrt{\sum_{k=1}^{|C|} (\lambda_{kk} \times q_k^C)^2} \times \sqrt{\sum_{k=1}^{|C|} (\lambda_{kk} \times v_{kk})^2}} \tag{3}$$

Once the user makes the decision of joining a group, the relevant material that he/she has selected is added to that one from the group, updating the corresponding A matrix and proceeding to the corresponding information updating in order to incorporate information from the new group member.

4. EXPERIMENTATION

In this section a description of the design of the experiment in the context of search-driven software development to evaluate our proposal is presented as well as it results and conclusions drawn from it.

4.1 Search-Driven Development

Software development is a collaborative process of both information creation and gathering. This is the main reason by which the software developers are constantly searching for the pertinent information to solve their problems at hand. The information needs of software developers range from those related to code, to process and teammates.

In the last few years, software developers habits of searching good source-code to reuse or rewrite, has increased the community's interest to improve it and some researchers are starting to refer to as Search-Driven Software Development or Search-Driven Development for short (SDD). So source code search engines have been implemented to facilitate this task. Some examples include *Google Code Search*, *Krugle*, *CodeFetch* and *Koders*.

Therefore, we have identified SDD context as a very interesting field where CIR features could be greatly exploited. For this reason we use the phrase collaborative SDD to refer to the application of different CIR techniques in the SDD process.

[2]We have to remark that it could be also applied to the case in which there are no predefined groups (all the users are searching by their own), so groups can be created from the scratch.

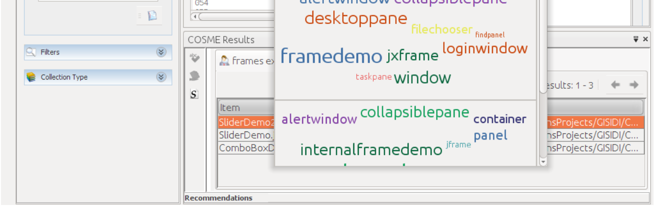

Figure 2: CoSS suggestion.

4.2 The Experimental Platform

Current SDD systems do not have support for explicit collaboration among developers with shared technical information needs, which frequently look for additional documentation on the APIs, read posts for people having the same problem, search the company's site for help with the APIs, or looking for source code examples where other people successfully used the APIs.

As an approach to these situations, we propose in this work, as experimental platform, the COSME plug-in [2], standing for COde Search MEeting. It makes a contribution in current SDD providing explicit support for teams of developers, enabling developers to collaborate on both the process and results of a search. COSME also provides collaborative search functions for exploring and managing source-code repositories and documents about technical information in the software development context.

In addition, we incorporate within COSME this time a recommender agent in the back-end side to suggest groups with similar information needs according to the queries of users that search alone, sending through front-end GUI a list of tag cloud. Each tag cloud represent the most relevant terms for each suggested group (Fig. 2).

4.3 The Test Collection

In the context of SDD, we have created our own test collection, composed of a total of 2420 files from different Java APIs for managing Graphic User Interfaces (GUI). More specifically, these libraries are (with their corresponding number of files considered): *Jidesoft* (634), *OpenSwing* (434)), *SwingX* (732)) and *Swing* (620). We have focussed on these APIs because they are directly related to the context of the experiment. Although we have to mention that they are not complete: we have only considered their most relevant API packages, removing those less important.

For evaluation purposes, we created our own relevance assessments: a group of 10 experts proposed a set of 100 topics strongly related to the objective of the experimentation, then their corresponding queries were submitted to each of the following search engines: Lucene, Minion, Indri and Terrier. A document pool was obtained by ranking fusion and later the experts, grouped in pairs, determined the relevant documents for each topic.

In the process of collection text transformation, in order to reduce the size of the index considerably, we use a stop-word set enriched by additional English common words as well as Java keywords. In addition, to improve the source code retrieval, we have used, in the underlying search engine used by COSME, a weighting-field approach to reward those terms that occurring, in descending order, in a class name, class method, class attribute, and source code comment.

4.4 The Evaluation Measures

Golovchinsky et al. remark in [5] that the evaluation is another challenge in collaborative IR, because traditional information retrieval systems are usually evaluated based on recall and precision measures, and these metrics assume a single logical searcher (even if more than one person contributed to the final search results). In [13] Pickens et al. propose the metrics Viewed Precision/Recall and Selected Precision/Recall, i.e. P_v, the fraction of documents - seen by the user - that were relevant and P_s, the fraction of documents - judged relevant by the user - that were marked relevant in the ground truth), and Rv/Rs as their dependent measures (Equations 4 and 5, respectively). To summarize effectiveness in a single number we use F_1 measure.

$$P_s = \frac{\mid D_{rel} \bigcap D_{ret} \bigcap D_{sel} \mid}{\mid D_{ret} \mid} = \frac{\mid D_{rel} \bigcap D_{sel} \mid}{\mid D_{ret} \mid}, D_{sel} \subseteq D_{ret}, \tag{4}$$

$$R_s = \frac{\mid D_{rel} \bigcap D_{ret} \bigcap D_{sel} \mid}{\mid D_{rel} \mid} = \frac{\mid D_{rel} \bigcap D_{sel} \mid}{\mid D_{rel} \mid}, D_{sel} \subseteq D_{ret}. \tag{5}$$

We extended these metrics with a new one [1], the effort of interaction, EoI (Equation 6): The fraction of explicit interaction of each user (E_i , such as explicit recommendations count and instant messages count) and CoSS size (S) multiplied by the CoSS longevity (T):

$$EoI = \frac{\sum_{i=1}^{S} E_i}{S * T}. \tag{6}$$

4.5 Method

The objective of the experiment is to determine whether it is useful the proposed methodology for recommending groups to users which are searching individually. We are not only interested in having good recommendations but also we are looking for objective evidence that the group performance improves when a new member is joined.

The following seeking scenario was proposed to a set of software developers without Java background: To select the most relevant classes to manage GUI components from several Java APIs. Both groups and users were asked to find the most relevant source codes for the following six GUI topics: Containers, Controls, Menus, Windows, Layouts and Events. Particularly, we have 18 groups with two members in such a way that for each topic we will have three different groups doing search tasks. Also, a total of 24 individual users were asked to find the same information, four users per topic.

Groups' member started the searching tasks before individual users, in order to have enough information to perform LSA and transform the content of the relevant documents into the latent dimensions. Regarding the concept space, C, that the LSA technique is going to obtain, we have considered a size of 11 concepts, assuming not only the six topics previously described but also additional concepts that could be considered around these topics, as for example, exceptions, two-dimensions graphics, concurrency, drag and drop, internationalization and themes of the interfaces. This number is used as a parameter for SVD.

Then, the users begin their individual search by submitting queries to the system. Thus when for a given query q, the user selects more than one relevant document (saves a source code file returned by the system, copies a fragment or several of them, or aggregates them to a project), the recommender agent start its work, searching for similar groups in the semantic/concept space. As output, those 5 groups

G_9	LAYOUT	before	0,0239	**0,049**	0,0149	0,464
		after$^{(+1)}$	0,0331		0,0156	
G_{10}		before	0,0243	***0,008***	0,0220	**0,042**
		after$^{(+3)}$	0,0401		0,0311	
G_{11}	EVENT	before	0,0243	**0,021**	0,0175	**0,024**
		after$^{(+3)}$	0,0393		0,0286	
G_{12}		before	0,0331	0,184	0,0162	0,259
		after$^{(+1)}$	0,0415		0,0218	

Figure 3: Means differences between groups evolution using Wilcoxon Signed Rank Test. Bold values correspond to statistical significant difference at $p \leq 0.05$ and italic bold values at $p \leq 0.01$.

with the highest semantic similarity according to Equation 3, ranked decreasingly, are recommended to the user.

The user is able to make the corresponding decision of joining a group or not. But, once a user made the decision of joining a group, he was integrated in the group CoSS as another teammate, working in the same topic as assigned to the corresponding group.

4.6 Discussion

Once the search sessions ended for all the users, COSME shows the decisions of individual users when joining a group. We can observe that all users selected a group to be part of it, and also that this selection was correct, as it was made coherently considering the common topic between user and group. Interestingly, only 12 (of the 18) groups have increased their number of members, as presented in Figure 3. The reasons could be two: first, because the system never recommended it to a user, or because, even being recommended, no user selected it to join to. In this last case, the reasons that motivate a user to join to a given group will be studied in the future, since they are not relevant to the purpose of this paper.

But as we indicate, we are interested in obtain good recommendations, but also to show that better results are obtained when joining forces. Thus, Figure 3 shows that in we obtain better results with 10 groups (83.337%), for both F_{1v} and F_{1s}, although there exists some differences between these metrics. Thus, with respect to F_{1v} 8 groups (66.67%) show significant differences ($G_1, G_2, G_4, G_5, G_8 - G_{11}$) and two of them show highly significant differences (G_5 and G_{10}). On the other hand, with respect to F_{1s} there are only 6 groups (50%) with significant differences before and after gained members ($G_2, G_4, G_5, G_8, G_{10}$, and G_{11}).

With respect to the usefulness of the proposed groups recommendation technique in terms of the EoI measure, comparing a group G_i *b*efore and *a*fter gaining members, $G_i - b$ and $G_i - a$, respectively, our hypothesis is that the EoI is smaller once a group has gained a member with shared information needs: $G_i - a > G_i - b$. In Figure 4 we can see that $G_1, G_3, G_5, G_6 - G_9$, and G_{11} (66.67%) are in line with the hypothesis assumed. Moreover, when $G_i - a < G_i - b$ the required effort does not increase considerably.

Therefore, as a general conclusion of our experimentation, recommending groups in collaborative frameworks is beneficial in two senses, improves the efficacy of the searches and reducing the interaction efforts of the users.

5. CONCLUSION AND FUTURE WORK

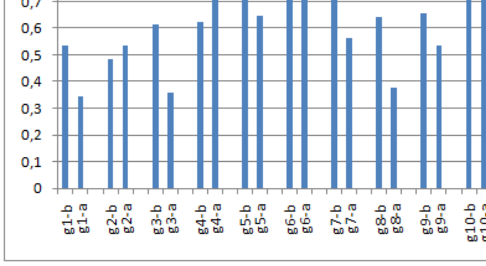

Figure 4: EoI distance for each group before and after gaining members.

User intent transition is just being recognized as an important research area within the CIR. While in some cases users' aggregation can be handled by conventional methods, we need to understand how the collaborative nature of online communities affects the requirements on CIR systems in terms of automatic recommendation of users or groups with similar information needs. Research in this direction needs to adopt the theories and methodologies of recommender systems, and supplement them with new approach constructs as appropriate. In this work we present *COSME* as a collaborative SDD tool that helps team developers to find teammates with similar information needs, as well as an experimental approach. Our ongoing work focuses on the *COSME* back-end which poses fundamental research challenges as well as provides new opportunities to let group members collaborate in new ways. In this direction we will integrate COSME with a P2P/hybrid-network Retrieval. Due to scalability and privacy issues we favor a distributed environment by means of a P2P retrieval feature based on hybrid architecture to use the user-generated data and collections of each user peer (*CASPER* – CollAborative Search in PEer-to-peer netwoRks).

6. ACKNOWLEDGEMENTS

This work has been supported by the Spanish research programme Consolider Ingenio 2010: MIPRCV (CSD2007-00018), the Spanish Ministerio de Ciencia e Innovacion under project TIN2008-06566-C04-01, and the Consejeria de Innovacion, Ciencia y Empresa de la Junta de Andalucia under project P09-TIC-4526.

7. REFERENCES

[1] J. M. Fernández-Luna, J. F. Huete, R. Pérez-Vázquez, and J. C. Rodríguez-Cano. Cirlab: A groupware framework for collaborative information retrieval research. *Information Processing and Management.*

[2] J. M. Fernández-Luna, J. F. Huete, and J. C. Rodríguez-Cano. Supplying collaborative source-code retrieval tools to software developers. In *British Computer Society - HCI 2011: 1st European Workshop on Human-Computer Interaction and Information Retrieval*, Newcastle, UK, 2011. CEUR WS.

[3] R. Fidel, H. Bruce, A. Pejtersen, S. Dumais, J. Grudin, and S. Poltrock. Collaborative information retrieval (cir). In *The New Review of Information Behaviour Research: Studies of Information Seeking in Context*, pages 235–247, USA, 2001.

[4] C. Foley and A. Smeaton. Evaluation of coordination techniques in synchronous collaborative information retrieval. In *JCDL Workshop on Collaborative Information Retrieval*, 2008.

[5] G. Golovchinsky and J. Pickens. Collaborative information seeking in electronic environments. In *Information Seeking Support Systems Workshop. An Invitational Workshop Sponsored by the National Science Foundation*, 2008.

[6] G. Golovchinsky, J. Pickens, and M. Back. A taxonomy of collaboration in online information seeking. In *1st International Workshop on Collaborative Information Retrieval held at JCDL '08*, Pittsburgh, PA, USA, 2008.

[7] P. Hansen and K. Järvelin. Collaborative information retrieval in an information-intensive domain. *Information Processing and Management*, 41(5):1101–1119, 2005.

[8] P. Husbands, H. Simon, and C. Ding. On the use of singular value decomposition for text retrieval, 2000.

[9] C. D. Manning, P. Raghavan, and H. Schtze. *An Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

[10] M. R. Morris. A survey of collaborative web search practices. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 1657–1660, New York, NY, USA, 2008. ACM.

[11] M. R. Morris and E. Horvitz. Searchtogether: an interface for collaborative web search. In *UIST '07: Proceedings of the 20th annual ACM symposium on User interface software and technology*, pages 3–12, New York, NY, USA, 2007. ACM.

[12] M. R. Morris and J. Teevan. *Collaborative Search: Who, What, Where, When, Why, and How*. Morgan&Claypool Publishers, 2010.

[13] J. Pickens, G. Golovchinsky, C. Shah, P. Qvarfordt, and M. Back. Algorithmic mediation for collaborative exploratory search. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 315–322, New York, NY, USA, 2008. ACM.

[14] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors. *Recommender Systems Handbook*. Springer, 2011.

[15] C. Shah and G. Marchionini. Awareness in collaborative information seeking. *Journal of the American Society for Information Science and Technology*, 61(10):1970–1986, 2010.

[16] M. Steyvers and T. Griffiths. *Handbook of Latent Semantic Analysis*. Lawrence Erlbaum Associates, 2007.