The Impact of Group Size on Collaborative Search

Master's Thesis

Kilian Cornelis Grashoff

The Impact of Group Size on Collaborative Search

THESIS

submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE
TRACK INFORMATION ARCHITECTURE

by

Kilian Cornelis Grashoff
born in Utrecht, The Netherlands

Web Information Systems
Department of Software Technology
Faculty EEMCS, Delft University of Technology
Delft, the Netherlands
http://wis.ewi.tudelft.nl

The Impact of Group Size on Collaborative Search

Author: Kilian Cornelis Grashoff
Student id: 4171373
Email: kiliangrashoff@gmail.com

Abstract

Collaborative search, where the activities of multiple users are combined to satisfy their information need, is an effective tool to to handle complex search tasks. People search collaboratively in groups of varying sizes. Various collaborative search systems have been studied in previous work, but they only investigate a fixed group size. Therefore, the impact of group size on retrieval effectiveness in collaborative search is an open research question.

We investigate the effect of group size on retrieval effectiveness in collaborative search in a crowdsourced study with a total of 305 participants, in groups varying in size from one to six. We use a web-based system for collaborative search in this study called SearchX. We extended SearchX with two features for *algorithmic mediation*, which aims to support users in division of labour and sharing knowledge with collaborators. We investigate three variants of our system with and without features for algorithmic mediation to investigate its effect on retrieval effectiveness for groups of varying sizes.

Our results show that the group recall increases linearly with group size. In contrast to a previous simulation study by Joho et al. [20] we do not find diminishing returns in group recall with increasing group size, suggesting that larger groups may increase group recall further. We also find that the investigated algorithmic mediation features do not significantly affect retrieval effectiveness. We conclude that the simulation results do not translate to our study, and that future collaborative search systems should be designed while taking the effects of mediation features on real users into account.

Thesis Committee:

Chair:	Dr. Claudia Hauff, Faculty EEMCS, TUDelft (supervisor)
Committee Member:	Dr. Christoph Lofi, Faculty EEMCS, TUDelft
Committee Member:	Dr. Julián Urbano Merino, Faculty EEMCS, TUDelft
Committee Member:	Felipe Moraes, M.Sc., Faculty EEMCS, TUDelft (supervisor)

Preface

The subject of this master's thesis is related to two topics that have been passions of mine for a long time: the interaction of people with information, and the interaction of people with each other in the digital domain. During this work I have gained a deeper appreciation for the complexities involved in these topics, and I am very happy to have had the privilege to work on them. Both the successes and struggles during the work have helped me to grow academically and as a person. This would not have been possible without the advice and support of the people that I would like to acknowledge here.

First, I would like to express my gratitude to my primary supervisor **Dr. Claudia Hauff**. During the entirety of this work her guidance was instrumental, from formulating the initial topic to the last comments on the final text. Her clear and prompt feedback was invaluable to help me find the right direction, especially when I was lost, and it was a privilege to learn from such an experienced researcher.

I am very grateful as well to my second supervisor, **Felipe Moraes**. His insights during our discussions down to the details of this work were very useful. I learned a lot from him, especially on how to analyze experimental data effectively. During our interactions Felipe was one of the most kind and positive people I have met, which helped me a lot during more difficult parts of the work.

Lastly, I would like to thank the friends and family who helped me get to this point. My parents, **Kees Grashoff** and **Anneclaire van Leest**, who have been a steady pillar of support during my studies. My grandfather, **Piet van Leest**, who is my role model as a technical student. My circle of close friends in Delft, **Wouter Bos**, **Hugo Hagedooren**, and **Luuk de Niet**, whose advice the past year was invaluable.

Kilian Cornelis Grashoff
Delft, the Netherlands
January 8, 2019

Contents

Preface iii

Contents v

List of Figures vii

1 Introduction 1
- 1.1 Collaborative Search . . . 3
- 1.2 Algorithmic Mediation . . . 3
- 1.3 Research Objective . . . 4
- 1.4 Approach . . . 5
- 1.5 Scientific Contribution . . . 5
- 1.6 Main Findings . . . 5
- 1.7 Outline . . . 6
- 1.8 Publications . . . 6

2 Related Work 7
- 2.1 Types of Search . . . 7
- 2.2 Collaborative Search . . . 7
- 2.3 Collaborative Search Systems . . . 10
- 2.4 Algorithmic Mediation . . . 15
- 2.5 Group Size . . . 18
- 2.6 Summary . . . 18

3 Extending `SearchX` 19
- 3.1 `SearchX` . . . 19
- 3.2 Supporting Multiple Search Providers . . . 22
- 3.3 Algorithmic Mediation Features . . . 23
- 3.4 Synchronizing Application State . . . 27
- 3.5 ScentBar . . . 28
- 3.6 Summary . . . 29

4 Research Design 31
- 4.1 Search Task . . . 31

4.2 Study Setup 32
4.3 Dataset and Retrieval Model 34
4.4 Crowd Work Setup 35
4.5 Post-processing 36
4.6 Metrics 38
4.7 Summary 38

5 Results **39**
5.1 Retrieval Effectiveness 39
5.2 User Behaviour 44
5.3 Summary 48

6 Conclusions **49**
6.1 Limitations 49
6.2 Future Work 50

Bibliography **53**

A `SearchX` Experiment Interface Screen Shots **57**

List of Figures

2.1 SearchTogether interface. 12
2.2 CoSense search strategies view. 13
2.3 Coagmento browser plugin interface. 14

3.1 Results page of SearchX version 0.1. 20
3.2 Architecture of SearchX version 0.1. 21
3.3 Search results with soft division of labour. 24
3.4 Architecture of algorithmic mediation components. 25
3.5 SearchX SERP with and without collapsed results. 26
3.6 Collaborative ScentBar. 29

4.1 Task template. 31
4.2 Example of a snippet generated for a search result. 34
4.3 SearchX experiment waiting room. 36

5.1 Mean group recall for each topic and search variant by group size. 39
5.2 Mean group recall for each topic, search variant, and group size computed in two-minute intervals. 41
5.3 Mean group precision for each topic, search variant, and group size computed in two-minute intervals. 42
5.4 User ratings for questions two to six of the post-test. 46
5.5 User ratings for question seven of the post-test. 47

A.1 SearchX experiment landing page. 57
A.2 SearchX experiment user registration. 58
A.3 SearchX experiment pre-test. 59
A.4 SearchX experiment waiting room. 60
A.5 SearchX experiment task description. 60
A.6 SearchX experiment introduction step. 61
A.7 SearchX experiment search engine results page. 61
A.8 SearchX experiment search engine results page with collapsed results. . . 62
A.9 SearchX experiment document viewer with AQUAINT news article. . . . 62
A.10 SearchX post-test questionnaire. 63
A.11 SearchX experiment thank you screen and confirmation token. 64

Chapter 1

Introduction

People have studied how to organize and retrieve information for hundreds of years in the field of library science [46]. With the advent of computing, and in particular the Web, the amount of information that could easily be accessed exploded. Search has become a daily activity for many people [24]. They not only use search engines to look up specific facts, but also for much more complex activities such as learning, planning trips, and scientific research. These search activities have been characterized as *Exploratory Search*: search activities where the user's information need is initially not well defined and constantly evolving [24].

One way to answer complex information needs is to search together with other people in collaboration. By collaborating with others, users can divide work [9]. Collaboration also allows users to benefit from pre-existing knowledge and expertise among members of the group [38]. For other activities, such as planning a trip, the goal of the activity is inherently collaborative. Search activities where the product of the activities of multiple people are combined to satisfy their information need are referred to as *Collaborative Search* [30].

A way to characterize the different types of collaborative search was was proposed by Golovchinsky et al. [12] and extended by Morris [29]. It characterizes collaborative search according to 6 dimensions:

- Intent: explicit vs implicit. During explicit collaborative search users are aware of the fact that they are working together in a group towards a common goal, while implicit collaborative search refers to techniques such as recommending similar items to similar users.

- Mediation: user interface vs algorithms. User interface-based mediation gives users tools to help them collaborate, but mediation features can also be built into the algorithms of the search system. The latter is referred to as algorithmic mediation.

- Concurrency: synchronous vs asynchronous. In a synchronous collaborative search session users are collaborating at the same time, while asynchronous search takes place at different moments in time.

- Location: co-located vs remote. Co-located collaborative search takes place on the same device or a system of connected devices in close physical proximity.

For remote collaborative search, users can be separated by large distances and their devices are connected over the internet.

- Role: symmetric vs asymmetric. During symmetric collaborative search all people in the group have equal roles, while asymmetric collaborative search gives users different roles. This allows the group to benefit from the expertise of individual members, but may require unavailable information about the users.

- Medium: PC vs emerging devices. Emerging devices includes all mobile devices such as smart phones and tablets.

In this work we focus on explicit collaborative search among a group of people who are remotely located and connected over the internet. We consider both user interface as well as algorithmic mediation features. The collaborative search system we use does not employ asymmetric roles, and is intended for use with PC's.

Morris [28] surveyed user behaviour related to collaborative search in 2006, and 2012 [29]. They found that the percentage of people who search collaboratively on a daily basis increased over ten-fold from 0.9% in 2006 [28] to 11% in 2012 [29]. The size of groups in which people search collaboratively varies. In 2006 only 19.3% of the surveyed users searched in groups with a size larger than 2 [28], while in 2012 this grew to 68.8% [29]. This increase shows that people frequently search in groups larger than 2 people, and the frequency of search in larger groups is increasing. However, there are challenges with searching collaboratively in groups. Kußmann et al. [22] found that users spend a significant amount of time on communication and monitoring other group members to orchestrate which activities the various collaborators perform. As groups grow in size, this orchestration becomes more and more difficult, and communication between group members can start taking up a significant amount of time. Because this time cannot be used for search activities, it incurs a communication overhead on the search activity [9]. How well collaborative search activities scale to larger groups has only previously been studied in a simulation study Joho et al. [20]. It is therefore an open research question how well real groups of users scale in terms of retrieval effectiveness.

Even though the effect of group size has not previously been studied extensively, approaches have been developed in collaborative search that can help to alleviate communication overhead. One approach is to give users tools to communicate more effectively. These are referred to as interfaced-based collaboration features. Although interface-based features may help with communication, the users still need to spend time and cognitive effort to use the features. Another way to alleviate communication overhead is to include techniques and algorithms in the search system that help to mediate which activities the group members perform. This is referred to as *Algorithmic Mediation* [11].

The main goal of this work is to study the **impact of group size on collaborative search**. To this end we have performed a crowdsourced experiment with our online system for collaborative search `SearchX`. We investigate the impact of group size on collaborative search with and without features for algorithmic mediation.

1.1 Collaborative Search

People can interact with others in various ways during a search activity. In the broadest form, these interactions are known as *social search* [7]. The subset of social search where users work together to satisfy an information need is known as *collaborative search* [36]. Collaborative search has been shown to be an effective way to help the user satisfy complex information needs, i.e., information needs that are explorative, open-ended and multi-faceted [36]. The domains in which collaborative search has been shown to be an effective solution are varied, such as patent research [15], travel planning [28], and personal health [29].

An important property affecting collaborative search is the size of the collaborating group. Even though users frequently collaborate in groups larger than 2, much existing research is focused on groups of size 2 [3, 21, 30, 33, 19, 38, 39, 40, 42, 16, 2, 17, 14, 37]. Even if larger groups are considered, they are usually of a fixed size [1, 31, 5, 32]. These studies can not provide insight into the relationship between group size and the collaborative search process. Joho et al. [20] considered group size as a factor in a simulation study. They investigated the retrieval effectiveness of explicitly collaborating groups of sizes up to five, by comparing various ways of assigning documents to users with and without algorithmic mediation features. Their results show that larger group sizes lead to higher search effectiveness in a recall-oriented task, albeit with diminishing returns. It is an open question how well these findings translate to the real world. A simulation always has limitations, one example in this case is that all relevance judgments by users were assumed to be correct. Another aspect that was not covered by the simulation is the cognitive load that may be experienced by users. We hypothesize that as groups grow in size, the increase in cognitive load that is likely required to coordinate the users actions with others in the group may affect their behaviour. Previous studies have shown that coordination efforts can take up a significant amount of time during collaborative search [9, 22].

1.2 Algorithmic Mediation

Algorithmic mediation features mediate work through the algorithms that are used to determine what results are shown to the user. This way, work is mediated without requiring the user to spend time communicating with others to coordinate the search activity. Algorithmic mediation can therefore be used to support collaboration by reducing the cognitive load due to coordination efforts [9]. There are two main types of algorithmic mediation: *Division of Labour* and *Sharing of Knowledge* [9].

Division of Labour refers to any process which helps the users in a group share their workload. The aim is to divide the work in such a way that redundant work is minimized. Work can be split in different ways. One way is to split the result space, assigning part of the documents to one user and part to another [9]. Another way is to give users different tasks in the search process, such as letting one user find new documents and having another examine them in-depth [11].

Sharing of Knowledge refers to any process that supports collaborating users to exchange information in the group [9]. This may help the group to evolve their understanding of the information need, by exchanging ideas between users. Search can be

regarded in this sense as a learning process [18]. Various interface level features for sharing of knowledge have been studied. However, these features run the previously mentioned risk of incurring a communication overhead on the users [9]. Algorithmic mediation features that support sharing of knowledge have also been proposed. Both Foley and Smeaton [9] and Joho et al. [20] have explored sharing of knowledge by using the relevance judgments of users to share knowledge in the group implicitly. They do this through *shared relevance feedback*, where the documents that each user marks relevant affect the results of later searches.

1.3 Research Objective

We consider the following research questions in this work:

RQ1 What is the impact of group size on retrieval effectiveness in a collaborative search session?

We study this question by setting up a crowdsourced experiment with real users. The users are assigned to groups of varying sizes. The groups are given a task with a pre-defined information need, and are instructed to collaborate to execute the task. Retrieval effectiveness of the group is investigated. We expect the following hypotheses related to this research question to hold based on a simulation study by Joho et al. [20]:

H1.1 Group recall increases with increasing group size, with diminishing gains.

H1.2 For topics with a higher number of relevant documents, increased group size will have a relatively higher impact on group recall (as it takes more work to find all relevant documents).

H1.3 A large group size is more useful early in the search session, with improvement in recall over lower group sizes decreasing as the search session progresses.

RQ2 How can features for algorithmic mediation be integrated in a collaborative search system and how do those features affect retrieval effectiveness in a collaborative search session?

We implement various features for division of labour and sharing of knowledge that have been shown to perform well in a simulation study [20]. Since the features have only been tested in a simulation, translating them to a real system is a non-trivial task. For example: only showing users unjudged documents may confuse them, because the results that they get will change and become seemingly less relevant over time (because documents judged as relevant are removed from later result lists). We investigate variants of our system with various mediation features and compare retrieval effectiveness between them. We expect the following hypotheses related to this research question to hold:

H2.1 Division of labour across a group of users increases their group recall, the effect is consistent across group sizes.

H2.2 Sharing of knowledge in a group of users increases their group recall, the effect is consistent across group sizes.

RQ3 What is the impact of group size on user behaviour in a collaborative search session?

Because the impact of group size in a collaborative search group has only been studied in simulation studies, its impact on user behaviour is an open research question. We therefore explore various aspects of user behaviour in our experiment in an exploratory fashion.

1.4 Approach

In order to investigate our research questions, we extended our collaborative search framework `SearchX` with algorithmic mediation features. We designed a crowdsourced study with 305 participants. The main dependent variable of our study was group size. We investigated groups of 1, 2, 4 and 6 collaborating searchers.

Participants in our study were asked to perform a recall-oriented search task, with topics selected from the TREC Robust track from 2005 [45]. Each group was given three topics in a random order, and asked to find as many relevant documents as possible in a time span of 10 minutes. Topics were selected to be difficult in order to mimic a complex real-world scenario where groups may benefit from collaboration.

1.5 Scientific Contribution

The main contributions of this work are:

1. An open-sourced implementation of both interface and algorithmic mediation features that facilitate division of labour and sharing of knowledge in a web-based system for collaborative search.

2. An empirical investigation with crowdworkers into the effect of group size on collaboration.

1.6 Main Findings

The main findings of this work are:

1. We find that most prior simulation-based results on the impact of group size on behaviour and search effectiveness do not hold in our experiment.

2. We find linear gains in retrieval effectiveness with increased group size in a task with a difficult topic; i.e. we do not find diminishing returns with increasing group size. This is in contrast to previous simulation studies, and suggests that explicit collaboration may benefit from even larger groups.

3. We find that algorithmic mediation features for division of labour and sharing of knowledge do not lead to a significant increase in retrieval effectiveness, and discuss various causes for this behaviour. This is also in contrast with previous simulation studies.

1.7 Outline

We describe related work in detail in Chapter 2. In Chapter 3 we describe how we extended our system for collaborative search `SearchX` to include algorithmic mediation features and other required modifications for the experiment. In Chapter 4 we describe the research design and in Chapter 5 we describe and discuss the results of the experiment. Lastly, we draw conclusions from our research in Chapter 6.

1.8 Publications

During the work for this thesis the author has contributed to the following papers that have been published or are in the process of being published.

1. Sindunuraga Rikarno Putra, Kilian Grashoff, Felipe Moraes, and Claudia Hauff. On the development of a collaborative search system. In *DESIRES '18*, 2018
   Full conference paper that describes version 0.1 of `SearchX`. The author contributed to the development of the system.

2. Felipe Moraes, Kilian Grashoff, and Claudia Hauff. On the impact of group size on collaborative search effectiveness. *Information Retrieval Journal*, 2019 (accepted for publication)
   Full journal paper that investigates the same research questions as this work. The author contributed to the development of the system, research design, conducting the experiment, and data analysis.

Chapter 2

Related Work

In this chapter we describe related work on collaborative search. We first describe types of search, and then describe collaborative search in more detail. We describe various systems for collaborative search, and compare the system that was used in this work `SearchX` to previous systems. We finish the chapter by describing previous work related to group size and algorithmic mediation.

2.1 Types of Search

Marchionini [24] classified three kinds of search activities: lookup, learn and investigate. In lookup search, the user has a clearly defined information need. For example, the user could be looking for the answer to a specific question, or looking up a specific web page. During a learning activity, the user aims to learn information about a given topic. Here, the information need is less clearly defined: the user does not yet know what specific information they aim to learn. Investigation activities are characterized by iterative searches over a longer period of time. After the user has acquired information, they formulate new ideas and hypotheses, and perform new searches to investigate further. Both learn and investigate tasks involve what Marchionini [24] described as *exploratory search*: search where the information need may not initially be clearly defined and evolves throughout the search process. Because learn and investigate activities are complex and iterative in nature, users can benefit from collaborating in a group [36].

2.2 Collaborative Search

Users frequently engage in collaborative search activities to conduct complex and exploratory search tasks [29, 38, 9]. They often perform these collaborative search activities using tools that are designed for single-user search [43, 31, 29]. Popular commercially available tools for single-user search such as Google or Bing currently do not support exploratory search well [24]. For example, they do not show a history of recent queries on the results page, which could help users evolve their understanding of the information need. Commercially available tools also do not support explicit collaboration [28]. This has a number of disadvantages. Users are not aware of the activities of collaborators, and may therefore duplicate work. If users want to exchange

knowledge with collaborators, they have to use external collaboration tools, which do not integrate with the search engine.

Morris [28] investigated the behaviour of users related to collaborative search in a survey conducted in 2006 [28], and subsequently in 2012 [29]. The surveys differed in approach. The 2006 survey was held among 204 Microsoft employees, while the 2012 survey was performed online with 167 participants. An important finding of these surveys was that the frequency of collaborative search increased significantly. Daily collaborative search increased from 0.9% to 11.0%, and weekly collaborative search increased from 25.7% to 38.5%.

The topic domains that people search for collaboratively was different in the two surveys by Morris [28, 29]. The top-5 topics are shown in table 4.1. This difference may be partly due to the difference in methodology. However, it is interesting to note that there seems to be a shift from using collaborative search for leisure activities such as travel planning, to professional use. In recent years the scientific community has developed various tools for collaborative search. However, Morris [29] found that users tend to "glue" existing other collaboration tools together instead of using dedicated tools for collaborative search. One possible explanation that was given is that these tools were too heavy weight, and that users preferred using existing tools that they already knew.

2006	2011
travel	professional
shopping	health/medicine
literature search	news/current events
technical information	technology
fact finding	travel

Table 2.1: Top-5 most frequently mentioned topic domains in 2 surveys into user behaviour related to collaborative search by Morris [28, 29], conducted in 2006 and 2012.

Various taxonomies to classify collaborative search have been proposed. Six dimensions to classify collaborative search were described in chapter 1. We describe various other taxonomies in order to help understand the properties of different approaches to collaborative search.

Capra et al. [4] described various styles of collaboration related to the role dimension. In *directed collaboration* a single person leads the search and directs what tasks the others should perform. This is a form of asymmetric roles. In *tightly coordinated collaboration* the collaborators coordinate with each other to divide the search task into parts, and each perform their part. In *loose/informal collaboration* people search with little coordination, and share results on an ad-hoc basis.

Kußmann et al. [22] investigated how Ellis' model of information seeking can be applied to collaborative search. These phases are useful to understand what types of activities users engage in during search, and how they can divide the activities between different collaborators in a group. The model consists of the following phases:

- *Starting*: beginning the search process (e.g. saying hello, agreeing on the information need, initial division of work)

- *Chaining*: using hints and links to find related documents from a document currently being viewed
- *Browsing*: searching for an area of interest in a semi-directed way
- *Differentiating*: filtering the material by quality (e.g. by reliability of the source)
- *Monitoring*: observing information sources of time to find new information that is added or information that is changed
- *Extracting*: systematically going through a single document and extracting information of interest
- *Verifying*: examining and comparing information to verify that it is correct
- *Ending*: stopping the search process and linking collected information together

The phases can occur in various orders. Kußmann et al. [22] performed a study with 15 participants in groups of 3 in a synchronous setting. Participants were given a realistic search task, where they had to make a presentation on a given topic. Kußmann et al. found that chaining and verifying did not occur often, and that monitoring did not occur at all. The lack of monitoring is explained by the fact that the search task was limited to a short period of time. They found that participants frequently switched between the browsing, extracting and differentiating activities. All groups followed a tightly coordinated collaboration style, where the task was split up between different collaborators. They found that participants spent a considerable amount of time and effort monitoring the activities of others. Team members often waited for replies to chat messages that they sent before continuing their task, causing a significant amount of idle time. This caused some groups not to complete the task in time. We hypothesize that this communication overhead can reduce the retrieval effectiveness of a group of collaborators significantly. Another result was that participants employed two different strategies for evaluating results. Participants that *scan* results only briefly viewed them, while other participants *read* large parts of documents. The authors found that teams that mainly employed scanning were more effective at the given task (reaching a higher F1-score and recall), while reading teams achieved higher precision.

Based on the behaviour of collaborative searches, previous work has also formulated design principles for collaborative search systems. Morris and Horvitz [30] formulated the following principles:

- Raising *awareness* among collaborators about the activities that others are doing. Examples of this include a shared query history, and information on what other group members did with individual documents (number of visits, ratings, and comments).
- Enabling *division of labour* in order to help group members coordinate their search activity. This can consist of interface features such as a chat that allows group members to divide work manually, or system-level features such as automatically providing different results to different users.

- *Persistence* of the results of the search activity. This helps users to perform complex searches over a longer period of time by recovering the context of previous search activities. In this way, both individual and collaborative search is supported. The noted awareness features help with persistence, other possible persistence features are the exporting and sharing of relevant documents, and generating a summary of the results of the search activity such as comments.

Foley and Smeaton [9] added a design principle:

- Enabling *sharing of knowledge* among collaborators. This refers to exchanging ideas and information between collaborators. Various forms to implement this principle exist, again ranging from interface features such as a chat or shared workspace, to implicit features such as shared relevance feedback, which will be described in the next section.

The features used for collaboration can be implemented using various approaches. Joho et al. [20] classified these approaches in three levels. Interface features that allow users to collaborate, such as a chat widget or a shared query history are classified as the *interface level*. The *technique level* consists of techniques that are implemented behind the scenes that allow the users to collaborate. Examples of this level include approaches that re-rank results, or allow users to split the results between collaborators. Finally, when the information retrieval model is adapted for collaborative use, this is called the the *model level*. Features may be implemented on one level or span multiple levels.

We can see from the different taxonomies that there are many different types of collaborative search. The taxonomies and design principles can be used to inform the design of a collaborative search system by carefully choosing which features support which types of activities. The levels from Joho et al. [20] can be used to inform the architecture of the implementation of collaborative search features.

2.3 Collaborative Search Systems

Several research systems for collaborative search have previously been developed. An overview of these systems and the properties of studies that were performed with them is shown in table 2.2. We can see from the table that most studies only investigate groups of size 2, and the others only investigate a single group size. Because of this, these studies do not provide insight into the effects of group size on collaborative search. Various systems provide interface features that support users in division of labour and sharing of knowledge. However, only `Cerchiamo` has features for algorithmic mediation. Most systems are not actively developed and not open-source. `Coagmento` is the only system that is open-source, but because it requires a browser plugin it is not suitable for crowdsourced studies.

The first system that was developed for explicit collaborative search is `SearchTogether` [30]. The authors first conducted a survey of Microsoft employees to investigate collaborative search behaviour, and developed SearchTogether as a prototype to test several design ideas for collaborative search systems. The system is aimed at remote collaborative search in groups with varying sizes. A screen shot of the interface

Table 2.2: Overview of systems for collaborative search. The following key statistics of studies into the systems are listed: group size (GS), number of groups (#G), number of search tasks per group (#T) and study type: [lab-fixed] refers to a lab user study with one or more fixed work/personal search tasks, [lab-nat.] to a lab user study where users self-selected their search task(s). Collection refers to the data collection used: Aquaint [45] is a commonly used collection of news articles, and TRECVid07 is a collection of videos. AL means the system has features for algorithmic mediation. OS means the system is published Open Source.

Name	**GS**	**#G**	**#T**	**Type**	**Collection**	**AL**	**OS**
SearchTogether [30]	2	7	1	lab-nat.	Web		
CoSearch [1]	3	12	3	lab-nat.	Web		
CoSense [31]	3	10	1	lab-nat.	Web		
Cerchiamo [11, 33]	2	4	24	lab-fixed	TRECVid07	✓	
none [19]	2	12	3	lab-fixed	Aquaint		
CoSense [32]	4	12	1	lab-fixed	Web		
Coagmento [14]	2	30	1	lab-fixed	Web		✓
Coagmento, Diigo [21]	2	8	1-3	lab-nat.	Web		✓
Querium [6]	-	-	-	-	-		
Results Space [5]	4	11	1	lab-fixed	Aquaint		
none [17]	2	10	3	lab-fixed	Aquaint		
`SearchX`	1-6	111	3	lab-fixed	Aquaint	✓	✓

is shown in Figure 2.1, we refer to letters in this figure in the following paragraphs. `SearchTogether` has several features that support awareness of the actions of other users. These features are a query history (b) that shows what queries each collaborator has posed, a chat client (a), and metadata for each document (h). The metadata consists of which collaborators have viewed a document, ratings, and comments.

`SearchTogether` includes two forms of division of labour at the techniques level. Split search (f) assigns the results of a query among collaborators in a round-robin fashion. When a user in the group executes a split search, all other collaborators are shown a new tab with the results that have been assigned to them. The second type of division of labour is multi-engine search (g), where the search query is sent to multiple search engines, and each user is assigned the results of one engine. While these features enable users to divide work, they need to coordinate the usage of the features manually. We hypothesize based on the work by Kußmann et al. [22] that this may cause an increased cognitive load and causes the users to spend time on communication. Users can manually assign which user gets the results of which engine, or use the default pairings. Persistence is supported in two ways: all sessions are persisted permanently, and users can export a summary of all documents that were rated positively.

`CoSearch` was developed as a prototype for co-located collaborative search [32]. The authors conducted interviews among students in order to determine what their challenges with co-located collaborative search were. The interviewed users noted that they had difficulty collaborating in groups larger than two our three users because the management of *off-task behaviour* became difficult. Users believed that larger groups

Figure 2.1: `SearchTogether` interface. SearchTogether was the first system developed for explict collaborative search, the following features are highlighted in the image: (a) chat, (b) query history, (c) search results, (d) recommendation queue, (e)(f)(g) search buttons, (h) page-specific metadata, (i) toolbar, (j) browser [30].

would have value if they could overcome these difficulties. The authors evaluated the system in a lab study with 36 participants in groups of three. They compared three conditions. In the *Parallel* condition collaborators searched on individual computers. In the *CoSearch* condition they used a single computer with the `CoSearch` system and multiple input devices. In the *Shared* condition all users used the same computer. While users preferred the parallel condition they reported low levels of awareness of the actions of others and little communication in this condition.

`CoSense` is a continuation of the work in CoSearch aimed at sensemaking [32]. It provides real-time visualization and contextualization of group search information by presenting various views of the information produced by the group. The *search strategies view* is shown in Figure 2.2. It displays the number of queries each member posed, and interactive tag clouds of the keywords used by the group and individual collaborators. The goal is to support awareness of the roles and skills of group members. The same view can also be used to show the number of URL's each user visited and tag clouds of the domains visited by the group and individual collaborators. The *timeline view* shows all chat messages and queries in chronological order, and is aimed at aiding the groups understanding of query evolution. The *chat-centric view* shows the chat and allows collaborators to open the page that other group members had open when they posted a message. Finally, the workspace shows a summary of commented results, allows users to tag results, and contains a to-do list and collaborative text editor.

The `CoSense` authors performed an observational study, where groups of three users were asked to collaborate in a vacation planning task [32]. The study consisted

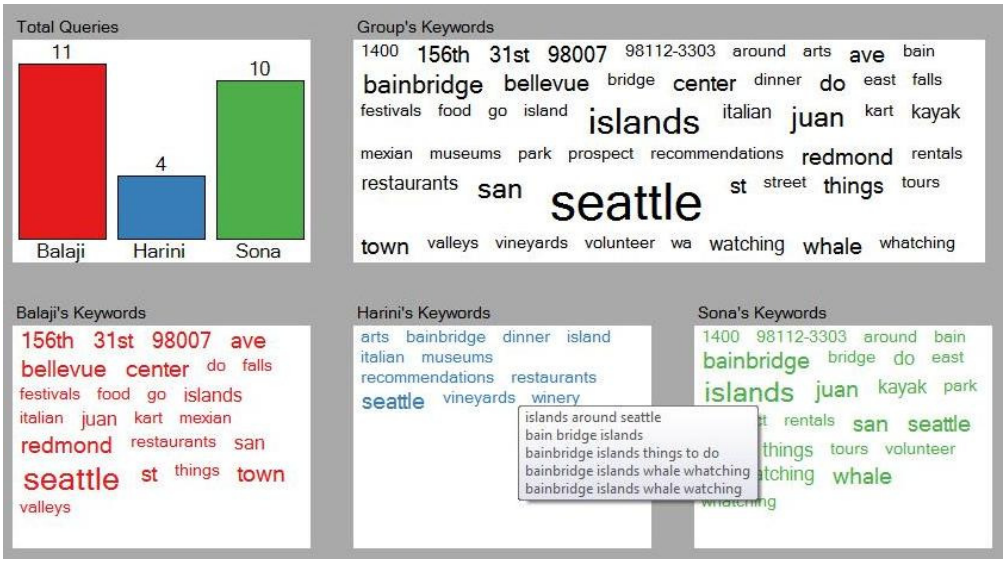


Figure 2.2: CoSense search strategies view [32]. This view shows information that helps users make sense of the collaborators earch activities. Shown are the total number of queries per collaborator, the top keywords from queries by the group, and the top keywords from queries by each individual collaborator.

of two phases: in phase one users searched collaboratively in a group. In phase two an extra group member was added who had to complete the planning task based on the information gathered by the group in phase one. In phase one users mainly used the tag clouds to check the skill of other users. In phase two the workspace and timeline views were mainly used to understand what the contributions, roles and decisions of the users in phase one were. Users reported in a post-study questionnaire and interviews that CoSense helped them in the handoff of sensemaking activities. We view this observational study as a form of distribution of labour by assigning different phases of the model by [22] (described in Section 2.2) to different users. The users in phase one were mainly tasked with starting, chaining, browsing, and extracting; the users in phase two were tasked with differentiating, verifying and ending.

Cerchiamo was the first, and until our work only, system to support algorithmic mediation [33]. The design goal of the system is to support synchronous collaborative search for video document collections in groups of two users. Each user has their own role. One user has the role of *prospector* and focuses on exploring the search space. The other user has the role of *miner* and examines the results for queries in detail. Both users are given a interface that is customized to their task. When the prospector marks documents as relevant other unjudged documents for the same query are added to a queue for the miner to inspect. When the prospector issues new queries and judges more documents, the queue is updated and reordered. The miner goes through the queue and judges which documents in it are relevant. The algorithmic mediation also works in reverse: when the miner judges documents as relevant, they are analyzed to suggest new query terms for to the prospector to use.

Joho et al. [19] built a system for collaborative search with a shared query history and chat. They compared three conditions: independent search, shared query history,

and shared query history + communication. They found that searchers had a more diversified search vocabulary when using the collaborative variants compared to the independent variant, but that the retrieval effectiveness of groups was not increased by collaboration [19].

Coagmento is the only existing system we found that is open source and actively developed [13]. Coagmento started out as a standalone application, but evolved to a hybrid web-based and plugin design based on user feedback. The interface of the browser plugin is shown in Figure 2.3. It provides shared bookmarks, a shared query history, the ability to save snippets from documents, the ability to rate documents, a chat interface, a collaborative notepad, and notifications when other users complete actions. Users can also view current information on a webpage, but need to use the browser plugin to perform new actions related to the page that they are currently viewing. This has the advantage of allowing the system to integrate with existing search engines. However, a disadvantage is that the user has to install software, making it unsuitable for crowdsourced studies.

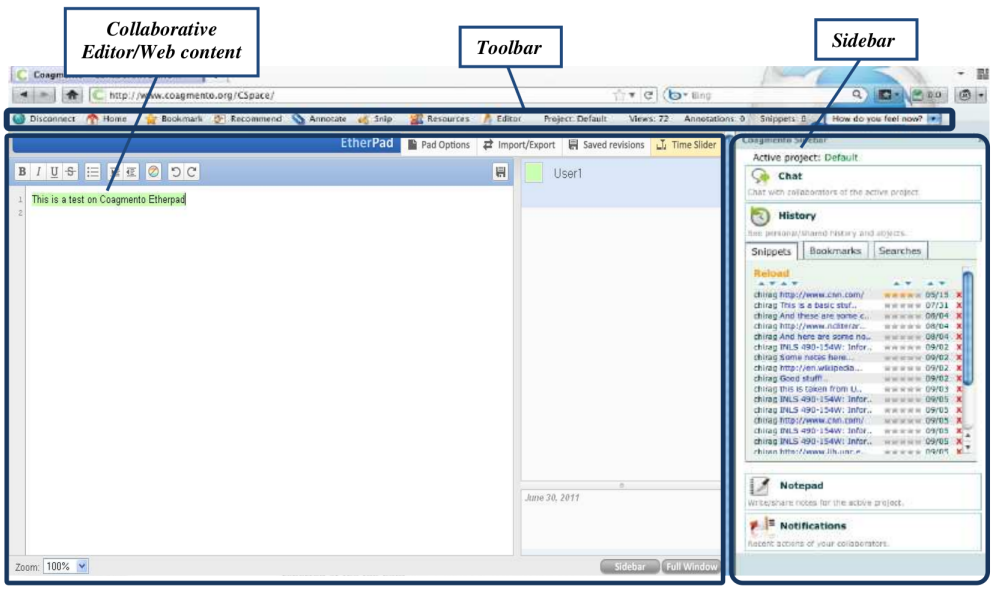


Figure 2.3: Coagmento browser plugin interface [13]. The collaborative editor can be used to share knowledge with other users. The toolbar is to perform actions related to the current page and shows metadata for the current page. The sidebar allows users to view all information that the group has produced so far.

González-Ibáñez et al. [14] investigated Coagmento as a tool for collaborative search. They tested variants with co-located collaborators, remote collaborators with only text communication, and remote collaborators with also audio communication. They found groups with remote collaborators to be more diverse in the information they explored compared to co-located groups. Adding audio support increased the time teams spent on communication, but lowered their cognitive load [14]. Kelly and Payne [21] also conducted a study that investigated how Coagmento's different features were used, and compared it to the system Diigo. They let users use both systems and interviewed them about their experience, and found that users experienced both

systems as useful. They also stress the importance of light-weight tools for collaborative search, because users were dissatisfied with tools that required too much effort to use compared to existing tools they already used [14].

`Querium` is a research system for session-based explicit collaborative search [6]. It has several features that distinguish it from earlier systems. For each result a histogram is shown that shows the history of who viewed a result, and how long ago it was viewed. The user can fuse the results lists of multiple queries, and issue relevance feedback searches which use a document to formulate a new query using reverted indexing [10].

`Results Space` is a prototype system aimed at supporting asynchronous collaborative search in groups of two to six users [5]. Distinctions of this work are that results space is web-based, and that it was evaluated using the task from the TREC 2005 Robust track [45]. This allowed the authors to calculate precision and recall metrics. They performed a lab-based study with 11 participants to evaluate the system. The task was described as a university assignment for which the users had to search for news articles. First, three participants conducted a collaborative search session. After the others had completed the session, a fourth participant was asked to help the group complete the assignment. They were not given instructions on what search strategy they should adopt. The group recall of the group with and without the fourth participant was compared. The increase in group recall was smaller than the authors expected. A possible explanation is that users found it easier to re-rate existing documents than to find new ones. This explanation is supported by the finding that users were more likely to re-rate existing documents than to rate new documents. The authors found that participants adopted various strategies, with some increasing recall more, while others increased precision more.

Htun et al. [17] investigated non-uniform information access scenarios among users in collaborative search. In a non-uniform information access scenario, the collections that the collaborators search in are different. Htun et al. [17] built a system for collaborative search with a query history, and awareness indications such as which documents have been viewed and judged as relevant. They found that the non-uniformity of the user's access to information did not have a significant impact on the retrieval effectiveness of a group [17]. This result is different from what the authors expected based on an earlier simulation study [16].

`SearchX` was developed by Putra et al. [35] in our research group. Its goal is to provide an open-source platform for research into collaborative search. It was initially used to study *search as learning*, where search engines are used to support learning activities [26]. It is implemented with a web-based interface to enable crowdsourced user studies into collaborative search. The main collaborative interface features seen in previous work are included: a shared query history, a list of saved documents, ratings, and comments. `SearchX` is described in detail in Section 3.1.

2.4 Algorithmic Mediation

The use of algorithms that automatically mediate work without needing explicit user input has been termed *algorithmic mediation* [33]. Pickens et al. [33] implemented a prototype for such a system called `Cerchiamo`, which was described in the previous

section. They also formulated a model and design principle for algorithmic mediation. The main design principle of algorithmic mediation is that "influence should be synchronized, but workflow should not" [33]. This means that the activities of one user should not interrupt the activities of other collaborators, allowing all users to work at their own pace. At the same time, the influence of the activities on the mediation algorithm should be immediate, allowing users to benefit maximally from mediation.

Several authors have conducted studies that evaluate the effects of different forms of algorithmic mediation on retrieval effectiveness. These studies and their properties are listed in table 2.3. We see that again, most studies only consider a single group size. The exception is the work by Joho et al. [20], they investigated groups of size one to five.

Table 2.3: Overview of key statistics of studies into algorithmic mediation: group size (GS), number of groups (#G), number of search tasks per group (#T) and study type: [sim.] refers to a simulation study with batch evaluation, [lab-fixed] to a lab user study with one or more fixed work/personal search tasks. Collection refers to the data collection used. - indicates unknown. DoL refers to evaluation of features or algorithms for Distribution of Labour, SoK refers to Sharing of Knowledge.

Authors	**GS**	**#G**	**#T**	**Type**	**Collection**	**DoL**	**SoK**
Joho et al. [20]	1-5	500	13	sim.	Aquaint	✓	✓
Shah et al. [38]	2	5	10	sim.	-	✓	
Soulier et al. [40, 41]	2	–	20	sim.	TREC Vol. 4	✓	
Soulier et al. [39]	2	70	1	lab-fixed	Web	✓	
Tamine and Soulier [42]	2	75	1	lab-fixed	Web	✓	
Htun et al. [16]	2	55	13	sim.	Aquaint		✓
Böhm et al. [2]	2	–	314	sim.	OHSUMED, CLEF-IP	✓	
SearchX	1-6	111	3	lab-fixed	Aquaint	✓	✓

Joho et al. [20] simulated 100 groups varying in size of 1 to 5 collaborators. Each simulated group searched for 13 different topics (out of 15 total topics) from TREC Robust. The associated corpus is the AQUAINT document collection containing news articles. The groups search for 20 iterations, judging 20 documents per iteration. The relevance judgments are assumed to be the same as those of the TREC judges. The queries are selected from a pool of queries. The query pool is generated by performing query expansion on queries that were collected from a user study. Which results are returned for a given query depends on the search strategy that is used. The authors tested 8 different search strategies for algorithmic mediation which are shown in Table 2.4.

The baseline strategy SS1 simulates independent search. The full document space is assigned to all users. In variant SS2 only unjudged documents are assigned to a user. This implements a basic form of division of labour, and was considered a more efficient baseline strategy by Joho et al. Strategy SS3 implements independent relevance feedback, using the set of documents that has been judged as relevant by a user for query expansion. Strategy SS4 uses the set of documents judged by the entire group

Abbreviation	Strategy
SS1	Independent search
SS2	Show only unjudged documents
SS3	SS2 + independent relevance feedback
SS4	SS2 + shared relevance feedback
SS5	Division of labour by cluster document space
SS6	Division of labour by round-robin document assignment
SS8	SS4 + SS5
SS10	SS4 + SS6

Table 2.4: Search strategies investigated by Joho et al. [20]. In each of the strategies, a different algorithm is used to mediate what documents are shown to the user for a given query.

instead. This implements an algorithmic form of sharing of knowledge: by judging documents as relevant users share their knowledge about topical relevance with others in the group. Both SS3 and SS4 were shown to increase group recall compared to lower-numbered strategies. More advanced variants of division of labour such as clustering the document space (SS5) and assigning documents in a round-robin fashion among group members (SS6) were also explored. While the simple division strategy SS2 significantly outperformed SS1 in group recall, variants SS5 and SS6 under performed compared to SS2. Combinations of the advanced division of labour strategies and relevance feedback strategies (SS8 and SS10) also under performed.

Foley and Smeaton [8] also performed a simulation study where they investigated algorithmic division of labour and sharing of knowledge. Their simulation differed from Joho et al. [20] in several ways. Groups of size two were simulated by pairing data of users that originally searched independently. Only formulate a single query is formulated for all users per group. After the first relevance judgment, relevance feedback is applied and the result list is immediately updated. This is called *incremental relevance feedback*. Errors in relevance judgments were simulated by detecting documents that could be perceived as relevant by the user incorrectly, and judging those as relevant. Five different variants were investigated to investigate the effects of division of labour. Best Individual considers the best result among individual group members. Independent considers the average result of the group without any collaboration. SCIR considers the result for a collaborating group without any division of labour. SCIR + Docs Seen Removed removes all documents that a collaborator has previously seen from the results list. SCIR + Full Div also removes documents that another collaborator currently sees from the result list. Foley and Smeaton [8] found that both the SCIR and independent variants under performed compared to the Best Individual result. The division of labour approaches were effective at improving the number of relevant documents found. Both approaches outperformed Best Individual, with Full Div outperforming Docs Seen Removed. They note that the division of labour approaches may decrease the user's understanding of the information need while searching in real life, and propose to include an *awareness widget* that shows users that documents have been removed.

Soulier et al. [40] showed that assigning collaborators roles according to domain expertise or based on the user's search behaviour [39] can be an effective form of algorithmic mediation when users have varying levels of expertise in different topics. Böhm et al. [2] developed a cost model for collaborative search. Their model showed that the effectiveness of division of labour approaches depends on the behaviour of users. If all users judge the same documents as relevant, division of labour is the most effective. However, when the judgments of users differ, division of labour may under perform, because users no longer re-judge documents that have already been judged by other collaborators. They developed an integer linear program that optimizes the result distribution to take this factor into account, and showed that it was effective in a simulation.

The various simulation studies and `Cerchiamo` show that approaches to implement division of labour and sharing of knowledge may be effective at increasing the effectiveness of a collaborating group. However, due to the assumptions these simulations make, it is an open question whether these results translate to the real world. `Cerchiamo` is a prototype that was built, but it only supports a specific type of algorithmic mediation, not the more generic variants investigated by simulation studies.

2.5 Group Size

Only Joho et al. [20] have previously studied the effects of group size on retrieval effectiveness. They found that adding collaborators to a group increased group recall, with diminishing returns. They also found that larger group sizes were primarily useful early in the search session, with the advantage diminishing over time [20]. This simulation study has several limitations, most importantly that all relevance judgments by the simulated users are assumed to be correct. Also, cognitive load that may be experienced by users due to interacting with other group members is not simulated. Therefore, we consider the effect of group size on collaborative search to be an open research question.

2.6 Summary

In this chapter various types and models for collaborative search from previous work were described. Existing studies into collaborative search systems and algorithmic mediation were surveyed, and relevant results from existing work was described.

Chapter 3

Extending SearchX

In prior work Putra et al. [35] developed an online system for collaborative search called SearchX. We extended SearchX with two features for algorithmic mediation. One implements distribution of labour by only showing the user unjudged results by default. The other feature implements sharing of knowledge through shared relevance feedback. While integrating these features into SearchX its architecture evolved in various ways. We took particular care in synchronizing the current state of the application in a way that gives the user consistent and up-to-date information without a jarring user experience. In the following sections, we first describe the original version (as released prior to this work) of SearchX, and then the modifications we made to extend SearchX for our experiment.

3.1 SearchX

Version 0.1 of SearchX was released in early 2018. This version includes various interface-based mediation features, but no algorithmic medation features. The aim of SearchX is to be a complete platform for research in collaborative search. The platform consists of an online collaborative search system, with various interface features to support collaboration between users. In order to enable empirical research, SearchX supports crowdsourced studies. In the following subsections we first introduce the original version of SearchX by describing its main design goals, features, and architecture. We then describe the updates that were made as part of this work to extend SearchX for our experiment: supporting multiple search providers, and adding two features for algorithmic mediation.

3.1.1 Features

Figure 3.1 shows the *search engine results page* (SERP) of SearchX version 0.1. On the top there is a searchbox where the user can enter their query (a). On the left hand side the list of search results is shown (b). SearchX supports various types of results, referred to as *verticals* (c). Included verticals are web pages, images, videos and news. SearchX can easily be extended to include new verticals. Each result can be saved using the yellow flag icon (d), the corresponding document is then shown in the list of saved documents saved documents (e). Saved documents can be deleted to remove them from the list, or starred to pin them at the top of the list.

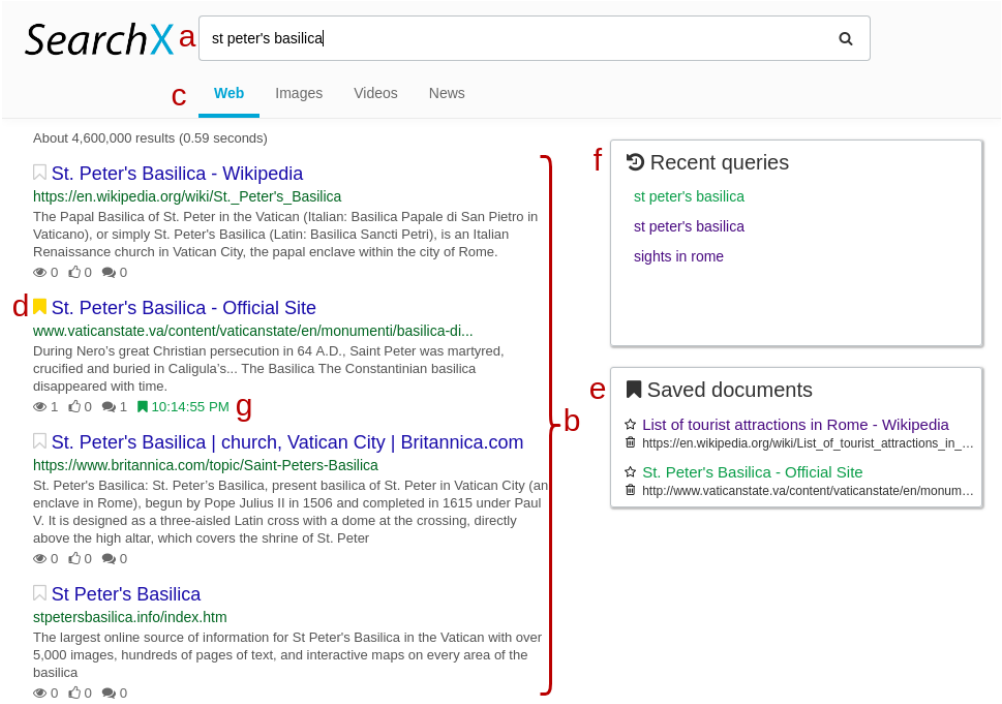


Figure 3.1: Results page of SearchX version 0.1. The following features are highlighted: (a) searchxbox, (b) list of search results, (c) vertical selection, (d) save result button (e) saved documents, (f) recent queries.

Users can collaborate in groups of varying size in SearchX. For experiments users are assigned to a group. The list of saved documents (e) is shared among the group. The second collaboration feature is the list of recent queries (f). This list shows the queries that users in the group posed, sorted with the most recent at the top. This helps users to be aware of queries that other people in the group posed, which can help them to improve their own queries.

Users can also view various metadata related to documents. The amount of views for each document, the current rating, amount of comments, and date and time of the last bookmark are shown for every result in the list (g). When a user views an individual document they can rate it, and leave comments for other users. All information that relates to a specific user is color-coded to indicate which user produced it.

SearchX contains various features to support its use in user studies. User actions are logged for data analysis. It supports easy definition of pre- and post-test questions. It also allows *introduction steps* to be defined, that highlight and explain various interface features. This allows users to quickly get familiar with the search interface, so that they can start a study without taking a long time to learn the interface.

3.1.2 Architecture

The architecture of SearchX version 0.1 is shown in Figure 3.2. SearchX is divided at the project level in a back-end and front-end application. Both are written in Javascript ES6, which allows easy re-use of code and data structures between the two. The back-

end runs on a centralized server using Node.JS. The front-end is a client side web application, which uses the API exposed by the back-end to implement `SearchX`'s interface.

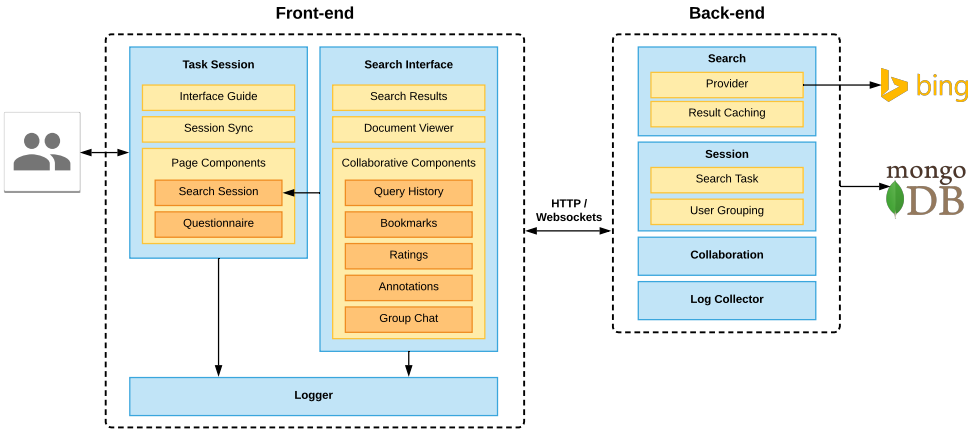


Figure 3.2: Architecture of `SearchX` version 0.1. The front-end runs client-side in the browser, and implements the collaborative `SearchX` interface. The back-end provides an API to the front-end with all functionality that it needs, such as saving user data in a mongo DB database and communicating with the search provider (Bing).

The back-end of `SearchX` version 0.1 consists of four main parts: search, session, collaboration, and logging. Each of these parts expose their own API endpoint that is used by the front-end. The *search* part interfaces with the *search provider*: the system that provides `SearchX` with its search results. `SearchX` 0.1 was hard coded to use the Bing provider. On top of the search provider a layer for result caching is implemented. If the same query is executed multiple times, the results are cached. This serves multiple purposes: each user is always shown the same set of results, and costs for using the Bing API are reduced.

The *session* part of the `SearchX` back-end is used to manage the groups and sessions in which users collaborate. A collaborative search session is comprised of a group of people all searching with a common information need. All metadata related to collaborative features, as well as the search task that users complete in the experiment, are linked to one session. The session API is used to retrieve the task and group information for a given user. The *collaboration* API is used for all collaborative features, such as adding and deleting bookmarks, annotations, and ratings. The *logging* API is used to store logs in the database. The back-end is not concerned with the content or types of logs: any data that is logged by the front-end is stored in the database.

The front-end of `SearchX` version 0.1 consists of three main parts: task session components, search interface components and the logger. The search interface components provide all search and collaboration functionality of `SearchX`. The task session components add functionality for a pre- and post-test, a task description, a timer, and introduction steps to the search interface, in order to enable `SearchX` to be used for user studies. Lastly, the logger is used by the other components to log user interactions

with the interface and send them to the server.

3.2 Supporting Multiple Search Providers

To run our experiment, the search provider needs to 1) support indexing and full-text search on our own dataset, and 2) support relevance feedback. Since Bing does not satisfy these properties, SearchX needed to be updated to support other search providers.

Originally the provider module in SearchX implemented the Bing API directly. It called the API with a separate method for each vertical, with the query and search options as arguments. In order to support multiple search providers, we decoupled the search provider from the rest of the SearchX back-end by introducing the *provider* interface.

The provider interface is defined as shown in listing 1.

```
fetch(
  query, // the search query
  vertical, // type of search results (web, images, etc)
  pageNumber, // result pagination number
  resultsPerPage, // the number of results to use per page
  relevanceFeedbackDocuments // the set of documents to use
  // for relevance feedback (if supported by provider)
)}.
```

Listing 1: Search provider interface specification

If a search provider does not support an option, or the option has an incorrect value, it must throw an error. The search provider must return a Javascript object which is structured as shown in listing 2. Strings represent variables that will be filled in with the value that is described in the string.

```
{
  matches: "number of matches",
  results: [
    "result",
    ...
]}
```

Listing 2: Search provider return object specification

The contents of result are determined by the vertical type that is used. For example, the web vertical has the following result type:

The full list of result types can be found in the SearchX back-end documentation.[1]

[1] https://github.com/felipemoraes/searchx-backend

```
{
  name: "name of the result",
  url: "full url",
  displayUrl: "url formatted for display",
  snippet: "part of text to display on search engine results page"
}
```

Listing 3: Web vertical results object specification

By specifying this interface, search provider modules can be written for any search engine that supports search with a text query, irrespective of the type of results. We have written search provider modules for `Elasticsearch` and `Indri`. The `Indri` module was used for the experiment described in this work. It uses an adapter module called `node-indri` that exposes `Indri` as a native Node.JS module [25].

3.3 Algorithmic Mediation Features

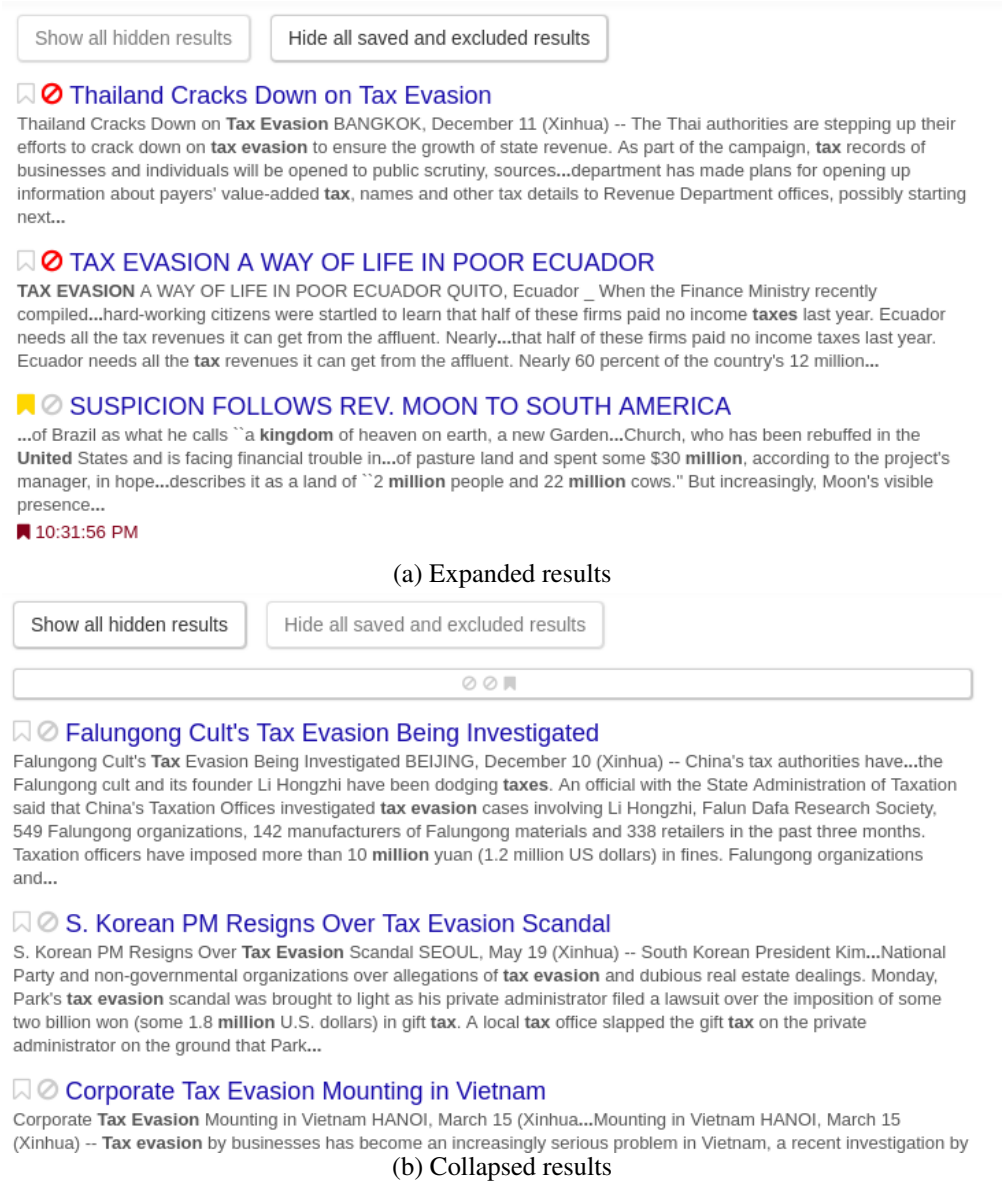
We implemented two features for algorithmic mediation, based on the work by Joho et al. [20]. The first feature is division of labour by showing only results that have not yet been judged by a user, and the second feature is shared relevance feedback.

3.3.1 Division of Labour

In their simulation study, Joho et al. [20] excluded results that have been judged by a member of the group from the result pages of all group members in the future. However, this approach has several drawbacks that we argue may affect retrieval performance negatively. Because results that used to be there disappear, users can get confused and start to view the system as unreliable. Users are also less likely to re-consider judgments by other users, which may lead to a decrease in precision of the judgments by the group. Finally, excluding results may disrupt the users evolved understanding of the information need: since they are missing relevant results, they may not get ideas for formulating new queries that could lead to more relevant results [9].

In order to combat these issues, we have implemented what we call *soft division of labour*. Instead of excluding results we collapse them in the result list. For each contiguous group of collapsed results we show an indicator that results have been hidden. The interface of our implementation is shown in Figure 3.3. Any document that a collaborator has saved or excluded is *collapsible*, and will be collapsed by default on new page loads.

The user can save a result with the yellow flag icon, but can now also exclude irrelevant results with the red exclude icon. This helps to prevent the precision of the viewed results from decreasing over time When results have been saved or excluded, they are hidden from future searches for the entire group. Hidden results are shown at the top of the results list in figure 3.3a. The user can click a contiguous set of hidden results to expand them. The user can also click an expanded result to collapse it again.



(a) Expanded results

(b) Collapsed results

Figure 3.3: Search results with soft division of labour. When results are collapsed, a bar is shown with icons indicating how many and which type of results are collapsed (saved or excluded).

At the top and bottom of the list, the user can use buttons to expand or collapse all collapsible results.

3.3.2 Sharing of Knowledge

The second algorithmic mediation feature we implemented is sharing of knowledge through shared relevance feedback. Relevance feedback means that after the user is shown a set of documents as results for their query, some form of feedback is gathered as to which documents the user thinks are relevant. We call this set of documents R.

This feedback is then used to expand the user's future queries with additional information, in order to improve the quality of the search results that are returned.

The algorithm that is used for relevance feedback is provided by `Indri` and is called *Relevance Model method 2* (RM2) [23]. RM2 is an extension of the retrieval model technique called *Language Modeling* (LM). In LM a language model M is constructed based in a given query Q. For each document D the probability of observing the query given the document $P(Q|M_D)$ is calculated, and the documents are ranked according to this probability. RM2 extends LM by calculating the probability that a word w in the set of documents R co-occurs with the query $P(w|Q)$. The query is then expanded by adding the k words that are most probable to co-occur with the query, k is a hyperparameter that is set in advance.

In the usual implementation of explicit relevance feedback, the set of documents that was judged by the user as relevant is used as set R. In our case we implemented shared relevance feedback, where the set of documents that has been saved by the entire group is used. Through shared relevance feedback, users are implicitly sharing knowledge about what information is relevant with other users in the group. Suppose for example, that one user starts searching after another user in the group has already saved a number of documents. Their results will be affected by the relevance feedback, thereby using the knowledge that the other user has created by saving documents. This mechanism is completely invisible to the user.

3.3.3 Algorithmic Mediation Architecture

The functionality needed for algorithmic mediation is part of the front-end, back-end, and search provider. An overview of this functionality is given in figure 3.4. The diagram shows the different modules in `SearchX` related to algorithmic mediation.

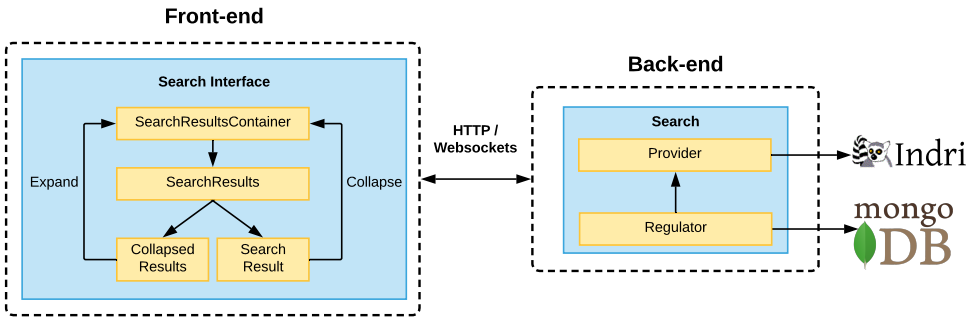
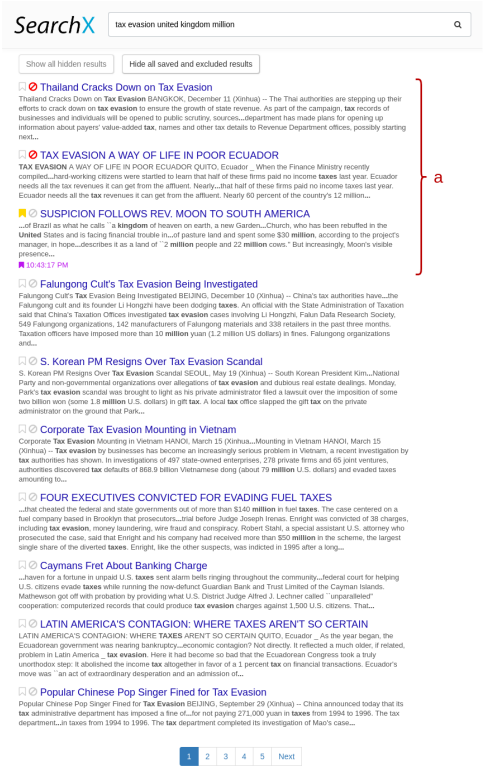


Figure 3.4: Architecture of algorithmic mediation components. All mediation functionality in the back-end is contained in the regulator, which calls the provider with the required arguments for the type of mediation that is activated. In the front-end, various components implement functionality to collapse results in the interface. Arrows indicate navigable references of one module to another in the direction of the arrow.

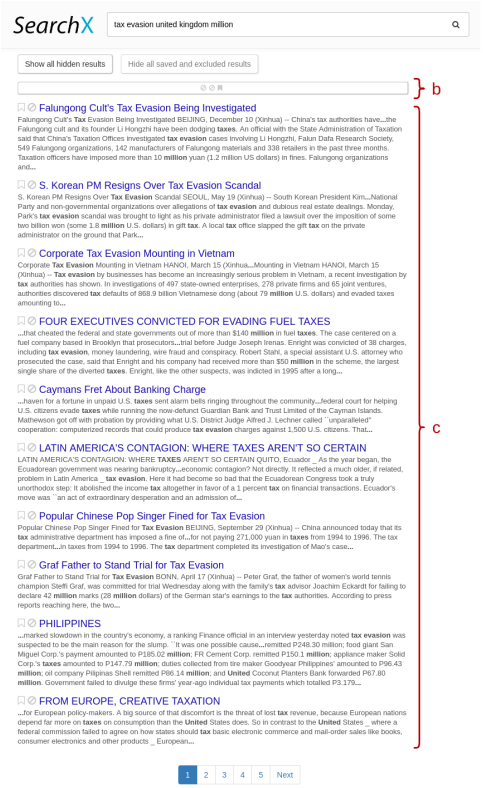
All functionality in the back-end is contained in the regulator layer. When the back-end API is called, it calls the search module. This module calls the regulator (instead of directly calling the provider as it did in the original version of `SearchX`). The

regulator gathers all needed information for algorithmic mediation, such as the identifiers of all bookmarked and excluded documents. The regulator calls the provider, with the correct arguments for the type of mediation that is enabled, and applies the necessary filtering steps to return the set of results for the page that the user requested.

An important complication due to the enabling of algorithmic mediation is that the total number of results per page can vary due to the hiding of collapsible results. Suppose the user views a SERP with results. If a user saves or excludes three results on the page as displayed in Figure 3.5a, there are now three *collapsible* results on that page (a). If the user reloads the page, it now contains 13 results in total as shown in Figure 3.5b. This is the case since a page always contains 10 *uncollapsible* results (c), and the page now also contains 3 collapsible results (b). In order to take this into account, the regulator requests extra documents from the search provider, so in this case 13 instead of 10. It then filters the result list such that there are the required number of uncollapsible results, and returns the resulting list.



(a) `SearchX` SERP with no collapsed results. Note that there are 10 results in total, of which three are collapsible (saved or excluded). (a) collapsible results.



(b) `SearchX` SERP with three collapsed results. Note that there are 13 results in total including the ones that are collapsed. (b) collapsed results, (c) uncollapsible results (results that are not saved and not excluded).

Figure 3.5: `SearchX` SERP with and without collapsed results.

In the front-end, the changes for algorithmic mediation are part of the `SearchResultsContainer`, `SearchResults`, and `CollapsedResults` components.

The `SearchResultsContainer` contains the list of results that are currently being

displayed on the page. It also manages the state of which results are currently collapsed or not by keeping a map of the id's of all collapsed results. The module has functions to update the collapsed state of either individual results, or all collapsible results on a page. These methods are passed down to the `SearchResults` component in order to allow the lower-level components to trigger the expanding and collapsing of results.

The `SearchResults` component is responsible for deciding whether to show a normal result, or a `CollapsedResults` component. It iterates over the list of results, and for each contiguous list of currently collapsed results it constructs a `CollapsedResults` component. This component shows the indicator we saw before. Because all state update logic is contained in the `SearchResultsContainer`, the `SearchResults` component is not concerned with how updates change the collapsed state: it is automatically updated by react when its properties change.

Finally, the `CollapsedResults` and `SearchResult` components contain the buttons that can change collapsed state. They call the update functions that are passed down to them as properties when clicked. When a collapsible `SearchResult` is clicked, it becomes (part of) a `CollapsedResults` component, and when a `CollapsedResults` component is clicked all results that were collapsed under it become `SearchResult` components.

3.4 Synchronizing Application State

There are two possible ways to implement division of labour and shared relevance feedback: immediate or delayed. In the immediate version, the SERP of all collaborators is updated as soon as a collaborator judges a document. However, because this causes results to hide unexpectedly, it may lead to a jarring user experience. For example, if a user is reading a the snippet of a result on the SERP while a collaborator bookmarks that result, it will suddenly disappear while the user is reading it. For relevance feedback, the same issue issue occurs, but with result reordering instead of disappearing. Recall the design principle for mediation that was posed by Pickens et al. [33]: "influence should be synchronized, but workflow should not". Because the immediate implementation requires users to interrupt their workflow when other users perform an action, we argue that it violates this design principle.

In order to combat this issue we implemented a delayed version of the algorithmic mediation features. Once a user has loaded a SERP, the results are fixed. The effects of new actions by other users on the results list are only seen when they load a new page. However, the state of the save and exclude buttons and saved documents list, do update immediately in response to actions by collaborators. This allows users to benefit from real-time awareness, without their workflow being interrupted. In the previous example where a user is reading a document while another bookmarks it, they can now keep reading it without being disturbed, because the result will not be collapsed immediately. Once they load a new page or click the *Hide all saved and excluded results* button, the result will be hidden.

Implementing delayed algorithmic mediation poses a technical challenge. The front-end of the of `SearchX` version 0.1 was implemented in such a way that the entire result list was refreshed in response to metadata update events. This means that the architecture of how updates are handled by the front-end had to be adopted. Instead

of re-fetching the results, we only re-fetch the relevant metadata. The existing result object is then updated in response to the updated metadata. Another option which could be implemented in the future is to move to a push-based model, where required metadata is included in the update event. Since this requires more extensive changes we chose not to implement a push-based model in the current version.

The delayed variant still poses challenges to the user's workflow. These challenges are related to pagination. Assume that a user is on a page greater than one. If a result on page one is saved by a collaborator, the total amount of items on that page increases by one. The reason for this is that one of the results will now be collapsed by default, so the page contains an extra result. This effect is shown in Figure 3.5. If the user now navigates to a later page, they will miss one item, since it has been shifted one page forward. A similar problem can occur with relevance feedback: when relevance feedback promotes a result that the user has not yet seen to page 1, they may never see it, even though it is considered likely to be relevant.

We combat these issues in two different ways. We define the next page that a user navigates to as always starting with the next result that they have not yet viewed. This ensures that the user will not miss results due to division of labour. If they reload the page it will start at a different place, but we consider this to be an acceptable trade-off in order to ensure that a user does not miss results. Secondly, we keep track of all documents that a user has seen. The regulator layer checks whether there are any unseen results on pages with a number lower than the page the user requests. If there are any results on lower-numbered pages, they are promoted to the current page. This ensures that the user does not miss results that are promoted by relevance feedback.

3.5 ScentBar

The last feature we implemented is a collaborative *ScentBar* based on the work by Umemoto et al. [44]. This feature was implemented with the intention of using it in future work, so it was not evaluated as part of the experiment in this work. The ScentBar shows a list of query suggestions. We use the Microsoft Bing API to provide us with query suggestions. For each of the suggestions, the information *gain* of the results for that suggestion to the current topic is calculated. Gain is a score that indicates how much information that is likely to be relevant for the current topic is contained in the unexplored results for that query. For each topic t, a set of aspects A_t is mined. This mining can be performed in various ways, in our case we use query suggestions provided by the Bing API based on the topic title. The algorithm that is used to calculate gain is designed according to three criteria: *importance* (documents relevant to aspects central to the topic produce higher gain), *relevance* (documents with higher relevance to an aspect produce more gain), and *novelty* (documents relevant to an unexplored aspect produce more gain compared to explored aspects) [44].

The ScentBar shows shows both how much information gain the user has already explored, and how much gain is still unexplored. We extend the work by Umemoto et al. [44] by making the ScentBar collaborative: instead of only showing how much the individual user has explored, we show an extra bar that shows how much information the group has explored. This allows the user to select suggested queries that contain the most potential information gain for the group. The interface of our Scent-

Bar implementation is shown in Figure 3.6.

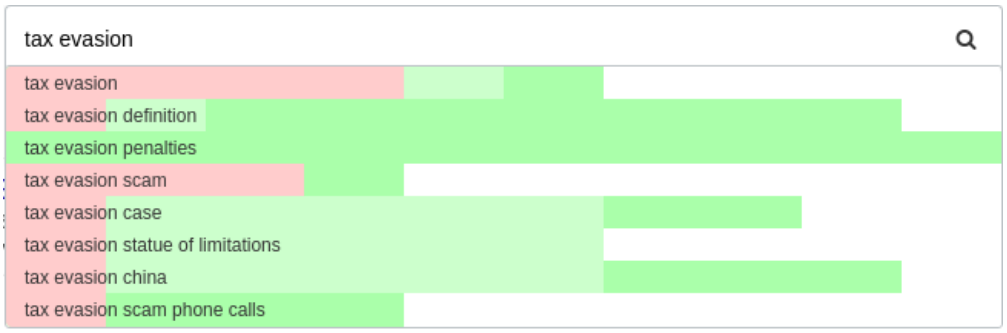


Figure 3.6: Collaborative implementation of ScentBar based on [44]. The dark green bar shows the information gain the user has already explored, and the light green bar the information gain the group has collaboratively explored. The red bar shows the unexplored information gain.

3.6 Summary

In this chapter, we first described the features of the version 0.1 of SearchX. We then described how we extended SearchX to support multiple search providers, which allowed us to use the Indri search engine for our experiment. We also described the two main features for algorithmic mediation we added: division of labour by hiding judged results, and sharing of knowledge through shared relevance feedback. We described the technical changes need to implement these features, and challenges related to the synchronization of application state.

Chapter 4

Research Design

To investigate our research questions, we ran a crowdsourced study with 305 participants. Recall that we are trying to answer the following research questions:

- **RQ1** What is the impact of group size on retrieval effectiveness in a collaborative search session?
- **RQ2** How can features for algorithmic mediation be integrated in a collaborative search system and how do those features affect retrieval effectiveness in a collaborative search session?
- **RQ3** What is the impact of group size on user behaviour in a collaborative search session?

4.1 Search Task

The performance of information retrieval systems can be measured in many ways. Two of the most commonly used metrics are recall and precision [48]. Recall indicates the proportion of the total amount of relevant documents in a corpus that the user has judged correctly to be relevant, and precision indicates the proportion of relevant documents in the set of documents that the user has judged to be relevant. Participants in our study were asked to perform a recall-oriented task in groups varying in size from 1 to 6 people. The task was described to participants as follows:

> Imagine you are a reporter for a newspaper. Your editor has just told you to write a story about [**ROBUST05 topic title**]. There's a meeting in an hour, so your editor asks you and your colleagues to spend 10 minutes together and search for as many useful documents (news articles) as possible and save them. Collect documents according to the following criteria: [**ROBUST05 topic description**].

Figure 4.1: Task template. This template is used to describe the topic to the user. The topic title and description are filled in according to the fields shown in Table 4.1.

As corpus we used the Aquaint document collection, which contains 1,033,461 news articles. We took topics for Aquaint from the TREC 2005 robust track [45],

which we will refer to as ROBUST05. We selected the ten most difficult topics. This was done by computing the average precision for the three best performing runs from ROBUST05, and selecting the topics with the lowest average precision. We selected difficult topics since collaborative search has the most potential to be effective in this case. From those 10, we manually selected three different topics we deemed interesting with the additional constraint of at least 30 documents. If a topic contains too few documents it will be difficult to evaluate the results. The topics selected are shown in Table 4.1.

The topics we selected are:

Table 4.1: Topics used in `SearchX` experiment. The id indicates the topic id in the ROBUST05 track [45] and the description is the text that is shown to the user to describe what documents are relevant to the topic.

topic title	id	description
piracy	367	What modern instances have there been of old fashioned piracy, the boarding or taking control of boats?
tax evasion indicted	650	Identify individuals or corporations that have been indicted on charges of tax evasion of more than two million dollars in the U.S. or U.K.
airport security	341	A relevant document would discuss how effective government orders to better scrutinize passengers and luggage on international flights and to step up screening of all carry-on baggage has been.

4.2 Study Setup

We explored three variants of `SearchX` in our study. An overview of the variants is given in table 4.2. We chose these variants since they correspond to the independent baseline and the two other most successful variants with algorithmic mediation in the work by Joho et al. [20]. More complex types of division of labour and sharing of knowledge did not yield additional increases in performance in their work, as described in detail in Section 2.4. We did not investigate a variant with interface based collaboration features and no division of labour to limit the number of variants, and since this variant of `SearchX` has already been explored in previous work [26]. We also did not evaluate the ScentBar feature in our experiment since it was implemented with the intention of using it in future work.

Variant S-Single is our baseline where users search independently: they are not aware of any of the actions of other users in a group. This variant replicates the strategy used by users when they search collaboratively in a group using search systems designed for single user search and no external communication.

S-UI-Coll is the basic version of our collaborative search system with division of labour. It contains both interface-based features (the shared query history and shared saved documents), as well as division of labour by only showing unjudged results by default. This variants allows us to investigate the effect of group size with a search

Table 4.2: Overview of our collaborative search conditions and their correspondence to the variants explored in [20].

S-Single	Independent search with individual bookmarks and individual query history (no awareness, no division of labour)
	Similar to variant SS1 of Joho et al. [20]
S-UI-Coll	S-Single + Shared saved documents, shared query history and collapsing of saved and excluded documents in the SERP (awareness, division of labour)
	Similar to variant SS2 of Joho et al. [20]. In contrast to Joho et al. [20], we collapse saved and hidden documents instead of excluding them.
S-UIAlg-Coll	S-UI-Coll + Shared relevance feedback (awareness, division of labour, and sharing of knowledge)
	Similar to variant SS4 of Joho et al. [20]

interface that is familiar to the user, while incorporating a basic form of division of labour. We chose not to include non-essential interface features such as chat, annotations, or ratings. This prevents users from spending extra time on communication overhead with increasing group size.

An important difference between S-UI-Coll and SS2 by Joho et al. [20], is that we do not exclude judged documents, but only hide them. We hypothesize that excluding results fully could negatively affect the search process in various ways. Users may be confused by missing results, or may miss information that helps them to evolve their understanding of the information need.

S-UIAlg-Coll is the variant of our system with all mediation features, interface-based division of labour, and techniques- and model-level sharing of knowledge. Sharing of knowledge is implemented by shared relevance feedback: documents saved by the group are used to expand the search query. In this way, knowledge about topical relevance is accumulated by all team members [20]. We explored sharing of knowledge in conjunction with division of labour since they serve complementary goals: sharing of knowledge improves the quality of results but increases their similarity, because the terms added by query expansion do not change when the user poses different queries. Division of labour helps users to avoid revisiting documents that have already been judged for relevance.

Joho et al. [20] showed shared relevance feedback to be effective at increasing recall over only division of labour, however we hypothesize that this effect may be different in the real world. Joho et al. [20] assumed perfect relevance judgments in their simulation, while in reality the quality of the query expansion is decreased when users incorrectly judge documents to be relevant. In addition, we hypothesize that users may experience an increased cognitive load because documents are sometimes re-ranked after a new page load, which may decrease the benefit of relevance feedback. We hypothesize that this problem gets worse as group size increases, because the frequency of new judgments increases, thereby increasing the frequency of the re-ranking

of results.

4.3 Dataset and Retrieval Model

Aquaint was indexed using the search engine Indri version 5.11. Prior to indexing we removed near-duplicate documents. This was necessary for interactive retrieval because the Aquaint collection contains many near-duplicates, causing the results to common queries to be very similar. For near-duplicate detection we used SimHash with the parameters $blocks = 4$ and $distance = 3$. We also removed documents with no title. This ensures that the user has a clear list of documents on the results page, instead of seeing many documents with untitled. After the cleaning steps our index contained 854,130 documents, 82.6% of the original size of the corpus.

We indexed the Aquaint collection with stopword removal and Krovetz stemming, using the default list of stopwords provided by Indri. We also used the functionality provided by Indri to generate query-dependent snippets with highlighted query terms. These snippets are much more informative than naive approaches such as taking the start of the document, because they show parts of the document text around occurrences of the query terms. An example of a snippet is shown in Figure 4.2.

Figure 4.2: Example of a snippet generated for a search result for the query “united states tax evasion million”. The snippet is generated such that parts of the text that occur around query terms are shown, and query terms are highlighted with bold text.

The retrieval model we used was language modeling (LM) with Dirichlet smoothing for S-Single and S-UI-Coll. [47] We used the hyper-parameter setting $\mu = 2500$. S-UIAlg-Coll uses RM2 relevance-based language modeling. [23] We used 10 feedback terms, and the set of documents used for language feedback consisted of all documents saved by a group of collaborators.

In order to verify the relevance feedback works as intended, we ran an offline simulation. For each of our three topics, we sampled five documents from the official TREC judgments for relevance feedback 20 times. We submitted a query to Indri with the topic title as query text with both LM and RM2. For RM2, the 5 sampled documents were used for relevance feedback. We removed the 5 sample documents from the results, and compute the recall of the retrieved ranked list. This simulation is similar to the approach used by Joho et al. [20] We found that RM2 to outperform LM on average by 82.65% across the three topics used in our study. This confirms that as long as our users save mostly relevance documents, RF improves the retrieval effectiveness.

4.4 Crowd Work Setup

Because we evaluate four different group sizes (1, 2, 4, and 6) and three variants for three topics, we have 36 different experimental conditions. Larger groups for variant S-Single are simulated by combining individual users, for financial savings. This is possible, since the users have no interaction with each other in this condition. We aim at having 10 groups of each non simulated size, based on the number of groups in previous work seen in table 2.2. This means we need a total of at least $(1+2+4+6)*2*10+10 = 270$ users to achieve this goal. In order to achieve this, we used a crowdsourcing setup. We experimented with both Amazon M-Turk and Prolific Academic as platform for recruiting crowd workers.

Due to the real-time nature of users joining groups, we got more participants for some group sizes than others. 335 workers participated in our study, of which 30 were excluded because they did not perform any actions. This means there were a total of $n = 305$ participants across 111 non-simulated groups of which data was used. The task took an average of 42 minutes, for which we paid £3.75.

On both platforms we ran into issues with forming groups. In order to form a group that can conduct the experiment in a synchronized way, we need multiple crowd workers to join at the same time. The platforms do not provide facilities for this, so we implemented our own *waiting room*. After a user joins, they wait in the room until there are enough collaborators, or they reach a pre-set timeout. If they reach the timeout, the worker can not complete the task, but we still need to pay them for their time. If the timeout is too long, workers leave because they get bored. The waiting room is shown in Figure 4.3.

We implemented several features to help us form large enough groups. The waiting room shows a counter of how much time the worker has spent and clearly explains the maximum waiting time (10 minutes). This gives the workers clarity about how long they will have to wait at most. We also included a game of snake that helps workers to pass the time, and play a sound when the group is formed to allow workers to open other tabs while waiting. In order to prevent losing valuable workers when they reach the timeout, we fall back to a smaller group size in that case. In order to ensure we have groups of the required sizes, groups with an odd number of users are split into a group of size 1, and a group with an even number of users. By starting the experiment with larger groups this allows us to decrease the number of smaller groups needed later.

Each group completes all three topics in a random order. When the group is formed, they are shown a description of the first topic for a fixed amount of time. This ensures that the workers stay synchronized. The workers are then shown a series of introduction steps that explain the functionality of the system. Figure A.6 shows an introduction step. This helps the workers to get started quickly, instead of having to spend time figuring out the features during the first task. The task completes after ten minutes. We call one execution of a task by a group a *search session*. This process (without the introduction steps) repeats until all three topics have been completed. If workers try to close the tab during the experiment they are shown a warning that this will cause them to quit. However, due to the crowdsourced nature there is still a significant number of workers that quits during the experiment. Users are stimulated to complete the experiment because they only get paid if they complete the experiment correctly. We can verify this because users enter their Prolific Academic platform to-

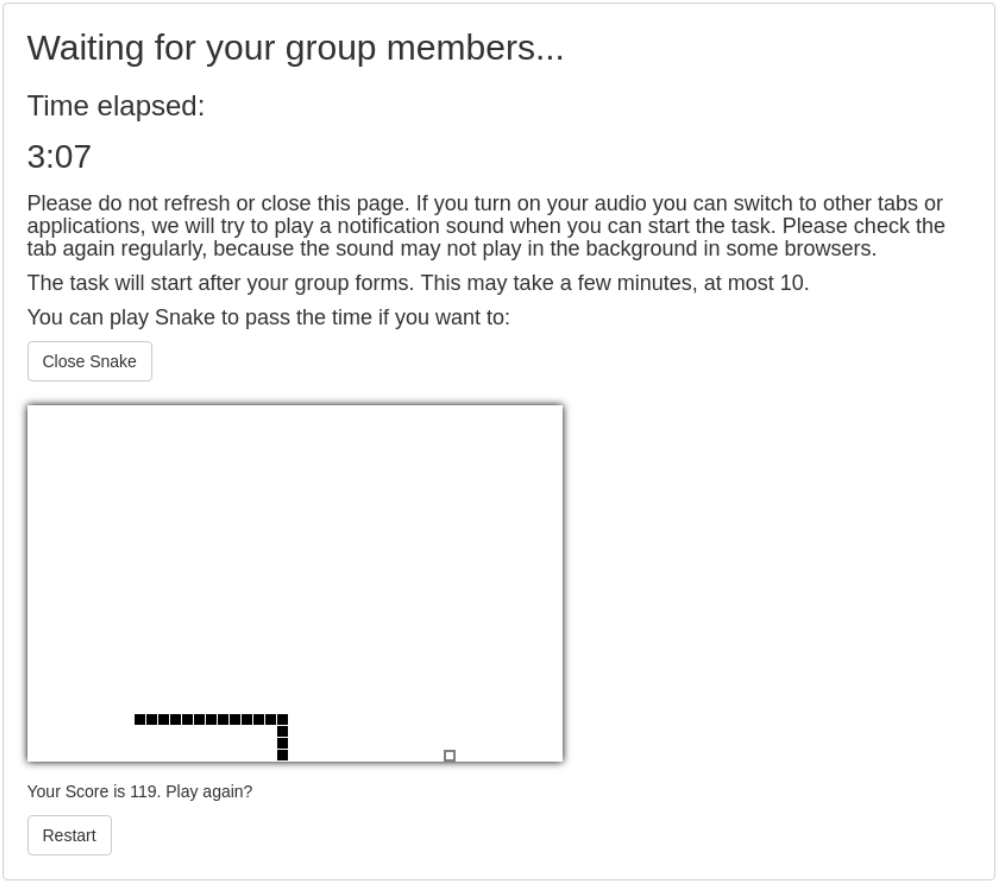


Figure 4.3: `SearchX` experiment waiting room. At the top a description is shown that explains to the user how the waiting room works. If the user wants to, they can play a game of snake in order to pass the time.

ken when they start the experiment, and we produce a token for them to enter on Prolific Academic when they finish the experiment. At the end of the experiment workers are shown a post-test in order to gather data on their subjective experience.

The post-task questionnaire contains eight questions on search satisfaction, shown in listing 4.

4.5 Post-processing

We deal with the issue of workers quitting by calculating the actual size of groups for each topic. Workers are only considered part of a group when they perform at least one query, either by typing or using the shared query history. Since there are now also groups of sizes 3 and 5, we binned the groups in the following sizes: 1, 2, 3-4, 5-6. We calculated the average size of groups in each bin in order to verify that there were no significant differences in average group size for the bins in each condition, the sizes are reported in chapter 5. We also used this average size in the simulation, making the results as comparable between variants as possible. Due to real-time nature in which participants join the experiment, some bins contain more groups than others. Table 4.3

1. How many people did you just now collaborate with (not including yourself)? *[Number]*
2. The color coding of the query history and bookmarks made sense to me. *5-level Likert scale [Disagree, Agree]*
3. It was easy to understand why documents were retrieved in response to my queries. *5-level Likert scale [Disagree, Agree]*
4. I didn't notice any inconsistencies when I used the system. *5-level Likert scale [Disagree, Agree]*
5. It was easy to determine if a document was relevant to a task. *5-level Likert scale [Disagree, Agree]*
6. How difficult was this task? *5-level Likert scale [Very easy, Very difficult]*
7. Did you find the collaborative features useful? (One row for each feature: Recent queries, saved documents, and hiding saved and excluded results) *5-level Likert scale [Disagree, Agree]*
8. Do you have any additional comments regarding `SearchX`? *[Open Question]*

Listing 4: Post-test questions asked to users. *Italicized* text indicates type of answer required for question.

shows the final number of groups for each bin. We reached the goal of 10 groups for most conditions. We have two conditions with only 8, and two with only 9 groups due to group size reductions after users were excluded for not performing any actions.

Table 4.3: Number of collaborating groups across search variants, topics and group sizes. For S-Single we simulate the collaborative search behaviour of larger group sizes with the data collected from the single-user search data.

	Topic ID	**{1}**	**{2}**	**{3,4}**	**{5,6}**
S-Single	650	12	–	–	–
	367	12	–	–	–
	341	12	–	–	–
S-UI-Coll	650	11	12	10	9
	367	12	11	10	9
	341	13	10	11	8
S-UIAlg-Coll	650	17	8	16	12
	367	17	11	13	13
	341	19	10	14	13

4.6 Metrics

Our main measure of retrieval effectiveness is *group recall*. This is the same metric used by Joho et al. [20]. It is an appropriate metric for our main evaluation, since the task is recall oriented. The group recall $GR(g,t)$ for group g and topic t is calculated by calculating the recall of the set of saved documents for the session in which group g searched for that topic. To calculate the mean group recall $MGR(s,t)$ for all groups for size bin s and topic t, group recall is averaged across all groups in a size bin:

$$MGR(s,t) = \frac{1}{|G_{s,t}|} \sum_{g \in G_{s,t}} GR(g,t). \tag{4.1}$$

In addition to group recall, we also investigate the *group precision* $GP(g,t)$. It is defined in the same way, but calculating the precision [48] for the set of saved documents instead. Although not the main measure we use to evaluate our hypotheses, calculating the group precision gives us extra insight into the quality of the relevance judgments of the group.

Temporal analyses: After {2, 4, 6, 8, 10} minutes we calculate the group recall for each group, using only the set of documents until that point in time. The start time used for each member is the point at which they issue their first query. Due to variance in the time users take to complete the introduction steps, there can be a slight delay between these start times. By setting the time window for each user separately, we make sure that all their actions are included in the analysis. Mean group recall and precision for different group sizes is computed in the same manner as discussed above.

4.7 Summary

In this chapter, we described the search task that crowd workers were asked to complete in three different variants. We described the three variants, and the details of the retrieval model used for each variant. We also described how we set up the crowdsourced study, challenges we faced with group formation and the necessary post-processing to address issues with group size. Lastly, we described the metrics that are used to evaluate our results for retrieval effectiveness.

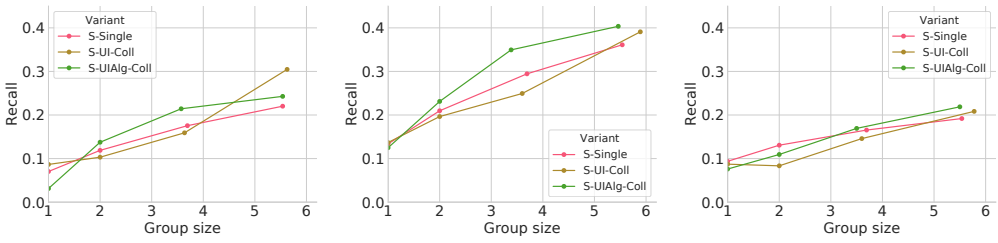
Chapter 5

Results

In this section we present the results of our user study. We first present the main results of our study: the effect of group size and the different collaborative variants we investigated on retrieval effectiveness. We then discuss these results in relation to the hypotheses we posed in the introduction. Finally, we investigate various aspects of how users behaved during the experiment and discuss the implications of these results.

5.1 Retrieval Effectiveness

We now describe results related to **RQ1** (What is the impact of group size on retrieval effectiveness in a collaborative search session?), and **RQ2** (How can features for algorithmic mediation be integrated in a collaborative search system and how do those features affect retrieval effectiveness in a collaborative search session?). Figure 5.1 gives an overview of the main result with respect to these research question and hypotheses **H1.1**, **H1.2**, **H2.1**, and **H2.2**: the effect of group size on group recall for the three variants and three topics that were investigated. We observe that group recall increases with increasing group size, across all topics and variants. We do not observe diminishing returns for the investigated group sizes, in contrast to Joho et al. [20].



(a) Topic ID 341 (b) Topic ID 367 (c) Topic ID 650

Figure 5.1: Mean group recall for each topic and search variant by group size. Group recall is calculated for each search session that a group performed, and averaged by (topic, variant, group size). The position of the points on the x-axis indicates the mean group size for each size bin, which varies slightly due to users dropping out.

Table 5.1 shows the mean group recall values for all investigated group sizes, topics, and variants. Statistical differences in recall are highlighted. These statistics sup-

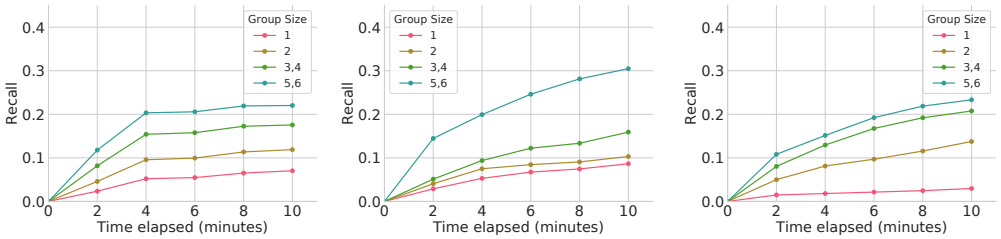
port the significance of our results for the previously mentioned hypotheses. Note that the entries for S-Single can not be compared to the other variants using statistical tests, because group sizes over 1 are simulated. We find that group sizes of 3-4 and 5-6 have significantly higher recall compared to smaller groups across all variants. We do not find statistically significant differences between variants of the same size. Additionally, we find that group recall increases linearly with group size; we do not observe diminishing returns in group recall as group sizes increases. This suggests that even larger groups could be beneficial in the type of collaborative search task that we investigated.

Table 5.1: Mean group recall (across all groups in a single topic/search-variant) for each topic, condition and group size. Statistical significance was determined via Tukey's HSD test independently for each topic; in each topic column, significant improvements at $p < 0.01$ are marked with superscript ^{XY} where X is the variant ('U' in the case of S-UI-Coll and 'A' in the case of S-UIAlg-Coll) and Y is the respective group size. For the S-Single simulated groups we determined significant values among group sizes only within S-Single via Kruskal-Wallis test independently for each topic (we omitted superscript symbols as all group sizes shows significant different results at $p < 0.01$).

	Group size bin	**Mean group recall per topic**		
		650	**367**	**341**
S-Single	1	0.094	0.134	0.070
	2	0.131	0.210	0.119
	3-4	0.165	0.295	0.175
	5-6	0.192	0.361	0.220
S-UI-Coll	1	0.087	0.137	0.087
	2	0.083	0.196	0.103
	3-4	0.146	0.249	0.159^{A1}
	5-6	$0.208^{U1,U2,A1}$	$0.391^{U1,U2,A1}$	$0.305^{U1,U2,U34,A1,A2}$
S-UIAlg-Coll	1	0.076	0.125	0.031
	2	0.109	0.231	0.138
	3-4	$0.169^{U2,A1}$	$0.349^{U1,U2,A1}$	$0.214^{U1,A1}$
	5-6	$0.219^{U1,U2,A1,A2}$	$0.404^{U1,U2,U34,A1,A2}$	$0.243^{U1,U2,A1}$

To investigate the evolution of the search process in more detail and investigate hypothesis **H1.3**, we investigated how group recall developed over time. Figure 5.2 shows the group recall for each topic and search variant computed in two-minute intervals. We observe that group recall increases over time, with diminishing returns, across all topics. We also observe that larger groups almost always outperform smaller groups, across all time intervals. Topic ID 650 Condition S-UI-Coll is an exception, where group size 1 performs the same as size 2.

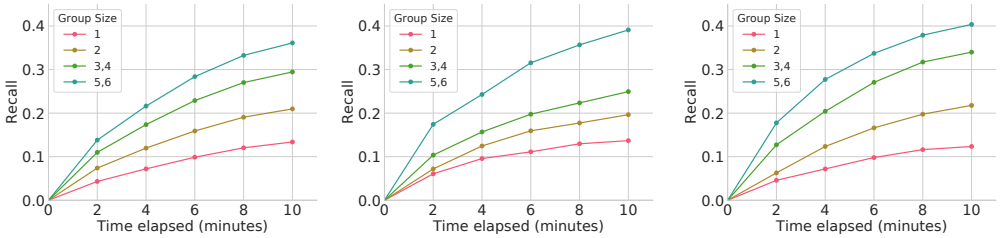
The temporal group precision was also analyzed, the results are shown in Figure 5.3. We observe that group precision generally stays constant from $t = 2min$ onward. Larger groups tend to show a slightly lower precision compared to smaller groups, but



(a) Topic ID 341, Condition S-Single with simulated groups

(b) Topic ID 341, Condition S-UI-Coll

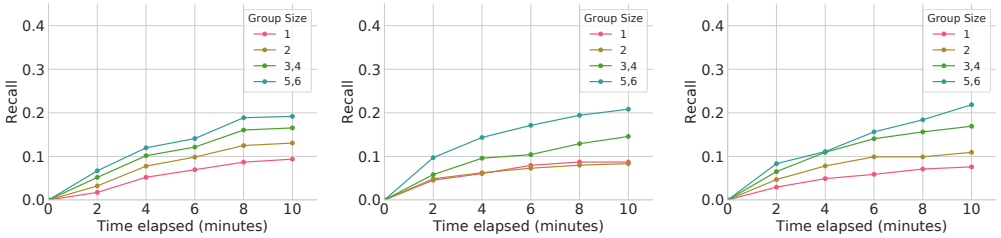
(c) Topic ID 341, Condition S-UIAlg-Colll



(d) Topic ID 367, Condition S-Single with simulated groups

(e) Topic ID 367, Condition S-UI-Coll

(f) Topic ID 367, Condition S-UIAlg-Coll



(g) Topic ID 650, Condition S-Single with simulated groups

(h) Topic ID 650, Condition S-UI-Coll

(i) Topic ID 650, Condition S-UIAlg-Coll

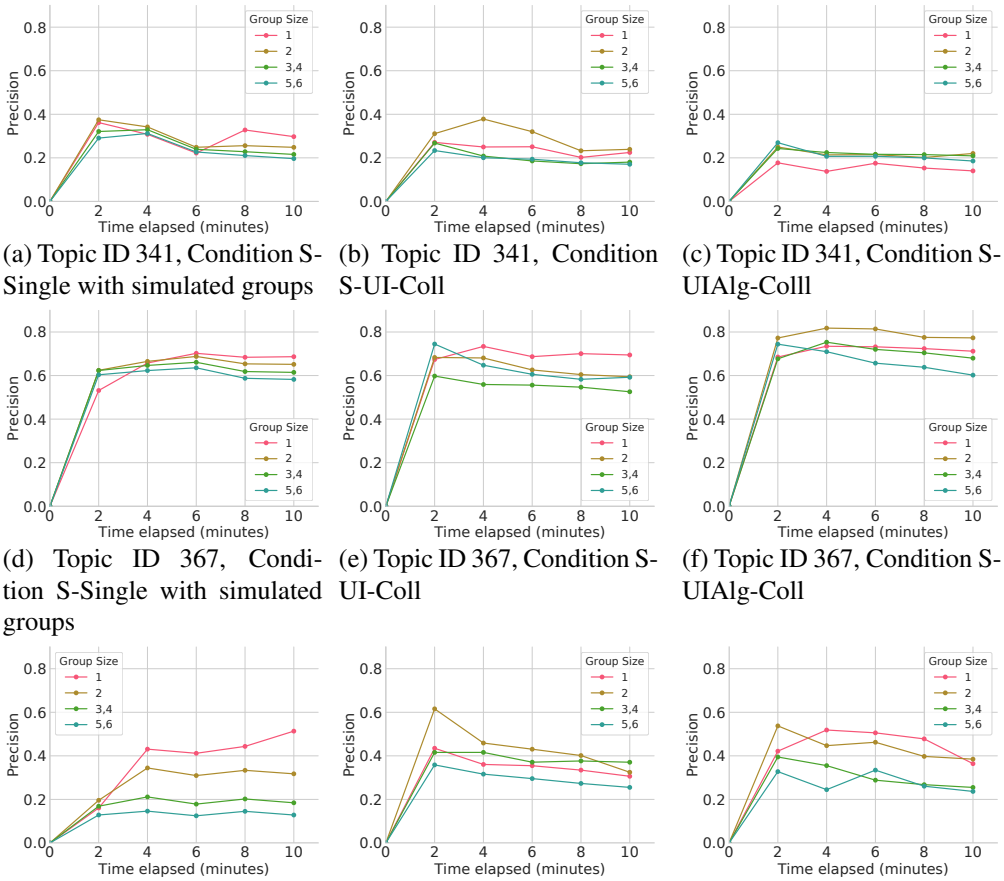
Figure 5.2: Mean group recall for each topic, search variant, and group size computed in two-minute intervals. For each time interval, the data point shows the group recall for the documents saved from $t = 0$ until the given time.

this effect is not very strong for most topics.

We now discuss the implications of the retrieval effectiveness results with respect to the hypotheses outlined in section 1.3. We also discuss related observations.

H1.1 Group recall increases with increasing group size, with diminishing gains. Based on the results shown in Figure 5.1, we find support for the first part of the hypothesis *group recall increases with increasing group size*. For all variants and all topics, larger groups sizes lead to higher group recall, with the exception of group size 2 for topic id 650 S-UI-Coll.

In contrast to Joho et al. [20] we do not find support for diminishing returns with larger group sizes for variants S-Single and S-UI-Coll. The trend for recall in variant S-Single is close to linear. S-UI-Coll only shows a small increase when moving from groups of size 1 to 2, and stronger increases when moving to groups of larger sizes. As shown in table 5.1 the differences between sizes 1 and 2 are not significant, while



(a) Topic ID 341, Condition S-Single with simulated groups

(b) Topic ID 341, Condition S-UI-Coll

(c) Topic ID 341, Condition S-UIAlg-Colll

(d) Topic ID 367, Condition S-Single with simulated groups

(e) Topic ID 367, Condition S-UI-Coll

(f) Topic ID 367, Condition S-UIAlg-Coll

(g) Topic ID 650, Condition S-Single with simulated groups

(h) Topic ID 650, Condition S-UI-Coll

(i) Topic ID 650, Condition S-UIAlg-Coll

Figure 5.3: Mean group precision for each topic, search variant, and group size computed in two-minute intervals. For each time interval, the data point shows the group precision for the documents saved from $t = 0$ until the given time.

the groups in size bins 3,4 and 5,6 show significant increases over the smaller groups. S-UIAlg-Coll shows yet another trend, with stronger increases in recall for groups of size 2 and 3,4 compared to the smaller groups, and slightly diminishing returns for larger group sizes.

The lack of diminishing returns for variants S-Single and S-UI-Coll suggests that even larger group sizes may yield additional increases in recall. A possible explanation for this different result compared to Joho et al. is that we focused on especially difficult topics. For none of the topics, search variants and group sizes the reported recall was greater than 0.4, showing that there were many relevant documents left to be found.

H1.2 For topics with a higher number of relevant documents, increased group size will have a relatively higher impact on group recall (as it takes more work to find all relevant documents). Topic 367 has 95 relevant documents, while topics 341 and 650 have respectively 37 and 32 relevant documents. If this hypothesis holds,

we would expect the impact of group size to be greater for topic 367. However, as we can see from Figure 5.1 the impact of group size for all three topics is very similar. Therefore, we find that H2 does not hold.

The observed lack of diminishing returns provides a possible explanation for this finding: a greater number of relevant documents only leads to a higher impact of group size if it allows the users in the group to find more relevant documents. If there are diminishing returns for topics with a low number of relevant documents due to the group already having found all documents, this is the case. However, since we observed no diminishing returns, there is no benefit to a topic having a larger number of relevant documents, unless the users can actually find more relevant documents (i.e. the precision of the viewed documents is higher).

H1.3 A large group size is more useful early in the search session, with improvement in recall over lower group sizes decreasing as the search session progresses. In order to answer this hypothesis we consider the temporal results in Figure 5.2. We can see that the relative benefit of group size is consistent over time, with smaller groups never catching up to the recall of larger groups. Therefore, we find no support for H1.3.

Again, the lack of diminishing with increasing group size provides a possible explanation. Since more users lead to a constant increase in recall, there is no reason for the benefit to disappear over time. It is interesting to note that for a given group size, recall does consistently show diminishing returns *over time*. This suggests that adding more users to a group is more effective compared to letting a smaller group perform the same amount of work by searching longer.

H2.1 Division of labour across a group of users increases their group recall, the effect is consistent across group sizes. In contrast to our expectations, we find S-UI-Coll to perform the same as S-Single. Therefore we find no support for H2.1. A possible explanation for the lack of effectiveness of division of labour is an increased cognitive load that users may experience. Because users spend time using the interface features, or get confused when results change, their effectiveness in completing the search task may suffer. However, we do not find clear indications for such an effect in the user behaviour results discussed later in this chapter. Another possible explanation is that real users do not benefit as much from division of labour as simulated users, since they may remember what results they have previously seen themselves and quickly skip over them. However, we would still expect to see some effect in this case, since users do not know what results have been judged by other group members in S-Single.

H2.2 Sharing of knowledge in a group of users increases their group recall, the effect is consistent across group sizes. In contrast to our expectations, we find S-UIAlg-Coll to perform the same as S-UI-Coll. Therefore we find no support for H2.2. A possible cause for this result is the cognitive load that was discussed in the previous paragraph. Additionally, the effectiveness of relevance feedback may suffer due to users making mistakes in their relevance judgments and saving documents that are not relevant for the given topic. This explanation is supported by the fact that the peak

precision is only 0.4 for 2 of the topics, indicating that users save many non-relevant documents.

5.2 User Behaviour

In order to answer **RQ3** (What is the impact of group size on user behaviour in a collaborative search session?) we analyzed user behaviour in several ways. We analyzed the behaviour of individual users for the different group sizes and variants in order to determine whether these variables affect the behaviour of individual users. We also analyzed the usage of collaborative features. Lastly, we analyzed the users subjective experience of the search process by a post-test in the form of a questionnaire.

The main characteristics of individual user behaviours are listed in table 5.2. In order to summarize these results, we computed the value per search session for each participant, and report the median value across all topics. We find that there is a decreasing trend in the amount of time spent on each document viewed with increasing group size, and that the document viewing time is relatively short, with users only spending about 10 seconds per document.

Table 5.2: Overview of individual search behaviours across the search conditions per session. The reported value is the median for sessions across all topics.

	Group size	**#Queries**	**Query length (#words)**	**#Viewed docs.**	**Viewing doc. time (#sec.)**	**#Unique saved docs.**
S-Single	1	6.00	3.50	12.00	12.17	10.50
S-UI-Coll	1	7.00	3.72	7.00	10.84	9.50
	2	8.00	3.69	5.00	10.22	6.00
	3-4	8.00	3.63	7.00	9.65	7.00
	5-6	9.00	4.10	6.50	8.71	6.50
S-UIAlg-Coll	1	6.00	3.67	12.00	11.63	7.00
	2	6.00	3.09	12.00	8.14	10.00
	3-4	7.00	4.25	7.00	9.11	7.00
	5-6	8.00	4.00	8.00	8.41	7.00

We analyzed whether users engaged with the provided interface features for collaboration: the query history and list of saved documents. The results are shown in table 5.3. Again, we report the median value across all topics. We find that the median groups of size 2 did not use the query history at all. Larger groups used the interface features more frequently, with groups of size 5-6 clicking about 4 queries and viewing 1 saved document in a session.

The first question in our post-test asked users how they perceive the size of their collaboration group in order to gauge how aware users are of their collaborators. To analyze this perception accurately we only consider groups where no users dropped out in this analysis. We find that users in larger groups generally underestimate the size of the group that they are in. Possible explanations for this are that users do not pay attention to how many different colours they see, or that they could not see the difference between similar colours. The user colours were randomly generated, so

Table 5.3: Usage of collaborative search interface features by groups of collaborators per session. Included are only clicks on queries and saved documents by collaborators that did not issue (save) the original query (document). The reported value is the median for sessions across all topics.

	Group size	**#Clicked queries**	**#Viewed saved docs.**
S-UI-Coll	2	0.00	0.00
	3-4	1.00	1.00
	5-6	4.50	1.00
S-UIAlg-Coll	2	0.00	0.00
	3-4	0.00	0.00
	5-6	4.00	1.50

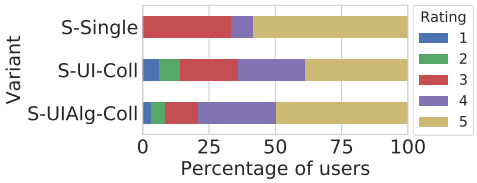
some user colours may have been hard to discern. We also find that a large number of users in single user groups perceive to be in a collaboration. A possible explanation for this is a priming effect due to users having been asked about collaboration in the pre-test questionnaire.

Table 5.4: Group size vs. perceived group size in % across search variants. Results reported based on the post-questionnaire (question 1, cf. Listing 4 in Section 4.4). Shown in grey is the cell value where *actual* = *perceived* group size.

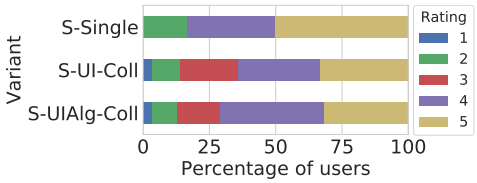
Condition	**Group Size**	**Perceived Group Size in %**						
		1	2	3	4	5	6	7+
S-Single	1	50	42	0	8	0	0	0
S-UI-Coll	1	90	0	0	10	0	0	0
	2	6	81	13	0	0	0	0
	3-4	4	23	31	15	19	4	4
	5-6	3	11	20	43	9	6	8
S-UIAlg-Coll	1	69	19	0	0	6	6	0
	2	0	44	19	25	6	0	6
	3-4	6	26	35	12	15	6	0
	5-6	4	19	10	45	12	10	0

Questions two to six of the post-test concerned the user's experience while using the system. For these questions we used the responses by all users. These questions used a 5-level Likert scale. The questions we asked are listed in listing 4. The full results are shown in Figure 5.4 and the median values for each variant are shown in Table 5.5. Most users agreed with questions two to five, with the median rating varying from four to five across questions and variants. The differences in median values between variants are small. The variance of ratings for S-Single is lower compared to the other variants, which can be explained by the fact that the number of users for this variant is lower (see table 4.3). Users rated the difficulty of the task neutrally with a median rating of three across variants for question six. The fact that this rating is different from the ratings for the other questions can be explained by the fact that the

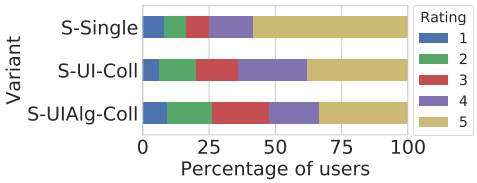
scale of this question concerns difficulty instead of agreement. We view the fact that users rate the task difficulty as neutral as an unexpected result, because the topics that were used in the task were selected to be difficult.



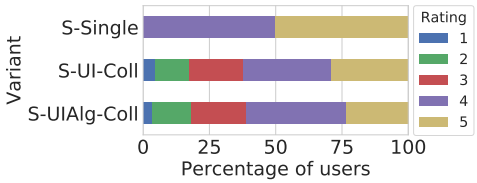
(a) Question 2: The color coding of the query history and bookmarks made sense to me. *[Disagree, Agree]*



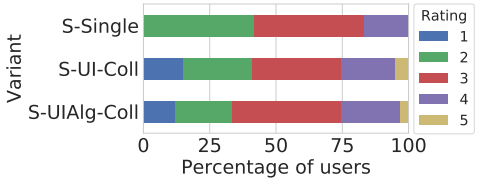
(b) Question 3: It was easy to understand why documents were retrieved in response to my queries. *[Disagree, Agree]*



(c) Question 4: I didn't notice any inconsistencies when I used the system. *[Disagree, Agree]*



(d) Question 5: It was easy to determine if a document was relevant to a task. *[Disagree, Agree]*



(e) Question 6: How difficult was this task? *[Very easy, Very difficult]*

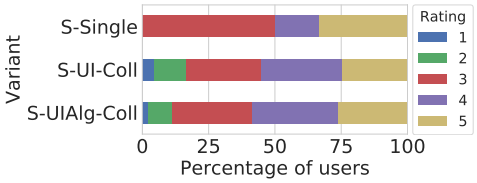
Figure 5.4: User ratings for questions two to six of the post-test on 5-level Likert scale.

Table 5.5: Median user ratings for questions two to six of the post-test on 5-level Likert scale.

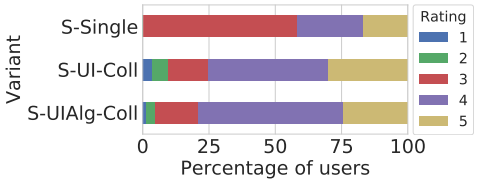
Question	S-Single	S-UI-Coll	S-UIAlg-Coll
2	5.00	4.00	4.00
3	4.50	4.00	4.00
4	5.00	4.00	4.00
5	4.50	4.00	4.00
6	3.00	3.00	3.00

Question seven of the post-test asked whether the user found the collaborative features used in the experiment useful. The question used a 5-level Likert scale from strongly disagree to strongly agree. The full results are shown in Figure 5.5 and the median values for each variant are shown in Table 5.6. Most users rated the features as useful across variants. The lowest median rating was 3 (neutral) for the usefulness of the recent queries list for S-Single. A possible explanation for this lower rating compared to the other variants is that the query history is more useful when used in

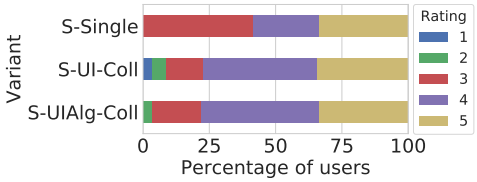
a collaborative setting, because it can then be used to see what queries others posed. Users may be able to remember which queries they have posed themselves over the relatively short time span of the task in our experiment, limiting the usefulness of the recent queries list for S-Single.



(a) Hidden saved and excluded results



(b) Recent Queries



(c) Saved Documents

Figure 5.5: User ratings for question seven (Did you find the collaborative features useful?) of the post-test on 5-level Likert scale.

Table 5.6: Median user ratings for question seven (Did you find the collaborative features useful?) of the post-test on 5-level Likert scale.

Feature	S-Single	S-UI-Coll	S-UIAlg-Coll
Hidden saved and excluded results	3.50	4.00	4.00
Recent Queries	3.00	4.00	4.00
Saved Documents	4.00	4.00	4.00

Question eight asked whether the user had any additional comments. We used an open card-sort approach to sort similar responses into categories. The top-10 categorized responses are shown in Table 5.7. A lot of users left positive comments, stating that they found the system useful, thanking us for the experiment, and stating that the system was easy to use. Users encountered various issues in completing the given task, mainly noting that they found it difficult to find relevant results for the topic. Users requested more search features in order to find more specific results and judge their relevance, such as the ability to exclude results containing given keywords, and showing the source and date for documents. A specific aspect of the documents that the users encountered issues with was the the location of the news articles: users often found that articles were from the wrong country, and requested the ability to filter articles by location. This issue can be explained by the fact that for the tax evasion topic users were instructed to select results from two countries. Users found it frustrating that they encountered similar results for different queries. A possible explanation for this is that relevance feedback caused results for different queries to be similar, and that these users were not able to combat this issue effectively using the division of labour functionality. However, some users noted specifically that they encountered

different results that were similar, indicating that this may have also been an issue with the dataset. Users also indicated frustration that many documents in the dataset were old, and some noted that they wanted to know the age of an article to judge its relevance. We hypothesize that users may intuitively consider older news articles to be less relevant, even though the task description did not mention this as a factor to consider.

Table 5.7: Top-10 categorized user responses for question eight of the post-test.

Comment description	**# responses**
`SearchX` was useful.	29
It was difficult to find relevant results.	25
Thank you.	19
I was unaware of number of other collaborators.	14
Feature request: advanced search functions (boolean search / ability to exclude terms, specific phrases with "", wildcards)	13
The system was easy to use.	11
Feature request: show metadata for results (e.g. source, date)	10
The location of the news articles I found were often not relevant to the query, and it was hard to filter results by location.	10
There was a lack of diversity of results, I often encountered the same results for different queries.	9
The results were too old.	9

5.3 Summary

In this chapter, we showed and discussed our results related to retrieval effectiveness and user behaviour of collaborative groups. For **RQ1** we observed that group recall increases with increased group size, without diminishing returns. For **RQ2** we did not observe a significant increase in recall by using algorithmic mediation. For **RQ3** we reported various statistics on user behaviour, and reported the subjective user experiences based on our post-test questionnaire.

Chapter 6

Conclusions

Collaborative search is a quickly evolving field. While collaboration in larger groups was rare a few years ago, nowadays it is common for people to collaborate in groups of increasing size. Despite this evolution, research into the effect of group size on collaborative search is very limited, with only Joho et al. [20] previously investigating it in a simulation study. Because simulations are a simplified model of the real world, we investigated how well these findings translate to the real world. We extended `SearchX` to enable a crowdsourced experiment into the effect of group size, and included features for algorithmic mediation intended to increase the effectiveness of larger groups.

The results of our study show that many of the findings of the previous simulations do not hold in our case. Of the hypotheses that we investigated, only **H1.1** holds partially. We found that group recall indeed increases with increasing group size, but did not find diminishing gains. This suggests that even larger groups may be useful for the type of difficult search tasks that we investigated. We did not find significant evidence for the other hypotheses.

A possible explanation for the lack of effectiveness of algorithmic mediation approaches in our case is the increased cognitive load that users experience due to them. Because users spend time using the interface features, or get confused when results change, their effectiveness in completing the search task may suffer. It is an open question whether these results translate to tasks that take place over a longer period of time, which we plan to investigate further.

We view this work as a confirmation that `SearchX` is an effective platform for conducting large scale crowdsourced experiments into collaborative search. We were able to implement novel features, and extended the platform to include new sources of search results. `SearchX` continues to evolve as a powerful open-source research tool into collaborative search.

6.1 Limitations

The crowdsourced setup of our study allowed us to empirically investigate the effects of group size and algorithmic mediation in the real world. Despite the fact that our setup is much closer to real-world usage of a collaborative search system than a simulation, it still suffers from limitations.

Most importantly, crowd workers are not real users. The reliability of our results relies on the assumption that crowd workers behave similarly to real users, which may not always be the case. Real users usually have a strong motivation to complete their search task as well as possible, while crowd workers may be less motivated. Besides motivation the demographics of our participants may be different from the group of users that uses a real world system.

A second difference of our task setup compared to real world usage is that the tasks were limited in time to only 10 minutes, and the total experiment was limited to about 40 minutes. This means that users only had limited time to judge relevance of documents, and limited time to get acquainted with how our system works. The synchronized nature of the study also means that our results may not translate to an asynchronous scenario.

Due to financial constraints we used a simulation approach to create the larger groups for S-Single. This provides us with an efficient way to create realistic larger groups, since there are no interactions between the users in S-Single. However, it does mean the significance of these results cannot be compared to real groups. Another constraint due to having to get enough crowd-workers in a short span of time is that we can not form groups much larger than 6 people using our current approach.

6.2 Future Work

The first possibility for future work stems from our result for **RQ1** (What is the impact of group size on retrieval effectiveness in a collaborative search session?) that group recall increases with increasing group size, without diminishing returns. This result suggests that for the type of task we studied, increasing group size further may increase retrieval effectiveness further. Therefore, it may be useful to explore collaboration in groups larger than size 6.

In order to provide a more realistic evaluation setting, `SearchX` could be used in a large scale online learning context: a Massive Online Open Course (MOOC). This will allow us to evaluate the system with users with a real information need, and provide the possibility to form larger groups. Using the `SearchX` system for a real world task will also allow us to evaluate it over a longer period of time, and to include asynchronous collaboration aspects.

With real-world users an aspect that becomes more relevant are the roles that different users take in the search process. Our current implementation is completely role agnostic. To support the coordination of larger groups effectively, research into features that support common roles may be useful. One example of this would be to give users a moderation role, where they are privileged to resolve conflicts between other group members.

Another interesting area for future work is the dynamic relationship between a group's information need and sessions. Currently in `SearchX`, a session relates to a fixed topic, and if a group wants to start investigating a new topic they need to start a new session. In the real world, a group may wish to investigate multiple related topics. As the information need evolves, the group may wish to split a session into multiple sessions for subtopics. Conversely, a group may wish to merge multiple sessions together. Groups may even want to share the results of their session with other groups,

who then use it as a basis for their own sessions. Including features that support the dynamic evolution of sessions could help to support these use cases.

Lastly, previous work suggests that it is useful for collaborative search systems to include features that support collaborative sensemaking and features that make it easier to share results with others [32, 29]. Therefore, we hypothesize that users could benefit from having an integrated workspace that people can use for sensemaking activities and to transform search results into information usable for other purposes. Examples of such activities could include combining snippets of results into summaries of information related to the topic, integrated tools for assessing the reliability of information, and a feature to share results with other collaboration tools.

Bibliography

[1] Saleema Amershi and Meredith Ringel Morris. Cosearch: a system for co-located collaborative web search. In *CHI' 08*, pages 1647–1656, 2008.

[2] Thilo Böhm, Claus-Peter Klas, and Matthias Hemmje. Towards a probabilistic model for supporting collaborative information access. *Information Retrieval Journal*, 19(5):487–509, 2016.

[3] Susan E Brennan, Xin Chen, Christopher A Dickinson, Mark B Neider, and Gregory J Zelinsky. Coordinating cognition: The costs and benefits of shared gaze during collaborative search. *Cognition*, 106(3):1465–1477, 2008.

[4] Robert Capra, Gary Marchionini, Javier Velasco-Martin, and Katrina Muller. Tools-at-hand and learning in multi-session, collaborative search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 951–960. ACM, 2010.

[5] Robert Capra, Annie T Chen, Katie Hawthorne, Jaime Arguello, Lee Shaw, and Gary Marchionini. Design and evaluation of a system to support collaborative search. *Proceedings of the Association for Information Science and Technology*, 49(1):1–10, 2012.

[6] Abdigani Diriye and Gene Golovchinsky. Querium: a session-based collaborative search system. In *ECIR '12*, pages 583–584, 2012.

[7] Brynn M Evans and Ed H Chi. Towards a model of understanding social search. In *Proceedings of the 2008 ACM conference on Computer supported cooperative work*, pages 485–494. ACM, 2008.

[8] Colum Foley and Alan F Smeaton. Synchronous collaborative information retrieval: Techniques and evaluation. In *European Conference on Information Retrieval*, pages 42–53. Springer, 2009.

[9] Colum Foley and Alan F Smeaton. Division of labour and sharing of knowledge for synchronous collaborative information retrieval. *IPM*, 46(6):762–772, 2010.

[10] Gene Golovchinsky and Abdigani Diriye. Session-based search with querium. In *Proceedings of the HCIR 2011 Workshop, Mountain View, CA, USA*, 2011.

[11] Gene Golovchinsky, John Adcock, Jeremy Pickens, Pernilla Qvarfordt, and Maribeth Back. Cerchiamo: a collaborative exploratory search tool. *Proceedings of Computer Supported Cooperative Work (CSCW)*, pages 8–12, 2008.

[12] Gene Golovchinsky, Jeremy Pickens, and Maribeth Back. A taxonomy of collaboration in online information seeking. *arXiv preprint arXiv:0908.0704*, 2009.

[13] Roberto González-Ibáñez and Chirag Shah. Coagmento: A system for supporting collaborative information seeking. *ASIST*, 48(1):1–4, 2011.

[14] Roberto González-Ibáñez, Muge Haseki, and Chirag Shah. Let's search together, but not too close! an analysis of communication and performance in collaborative information seeking. *IPM*, 49(5):1165–1179, 2013.

[15] Preben Hansen and Kalervo Järvelin. Collaborative information retrieval in an information-intensive domain. *IPM*, 41(5):1101–1119, 2005.

[16] Nyi Nyi Htun, Martin Halvey, and Lynne Baillie. Towards quantifying the impact of non-uniform information access in collaborative information retrieval. In *SIGIR '15*, pages 843–846, 2015.

[17] Nyi Nyi Htun, Martin Halvey, and Lynne Baillie. How can we better support users with non-uniform information access in collaborative information retrieval? In *CHIIR '17*, pages 235–244, 2017.

[18] Peter Ingwersen. *Information retrieval interaction*, volume 246. Taylor Graham London, 1992.

[19] Hideo Joho, David Hannah, and Joemon M Jose. Comparing collaborative and independent search in a recall-oriented task. In *IIiX '08*, pages 89–96, 2008.

[20] Hideo Joho, David Hannah, and Joemon M Jose. Revisiting ir techniques for collaborative search strategies. In *ECIR '09*, pages 66–77, 2009.

[21] Ryan Kelly and Stephen J Payne. Collaborative web search in context: a study of tool use in everyday tasks. In *CSCW '14*, pages 807–819, 2014.

[22] Tanja Kußmann, Stefanie Elbeshausen, T Mandl, and C Womser-Hacker. Discovering ellis' phases of information seeking behavior in collaborative search processes. In *CIS Workshop at ACM CSCW*, 2013.

[23] Victor Lavrenko and W. Bruce Croft. Relevance based language models. In *SIGIR '01*, pages 120–127, 2001.

[24] Gary Marchionini. Exploratory search: from finding to understanding. *Communications of the ACM*, 49(4):41–46, 2006.

[25] Felipe Moraes and Claudia Hauff. node-indri: moving the indri toolkit to the modern web stack. In *ECIR*, 2019.

[26] Felipe Moraes, Sindunuraga Rikarno Putra, and Claudia Hauff. Contrasting search as a learning activity with instructor-designed learning. In *ACM CIKM*, 2018.

[27] Felipe Moraes, Kilian Grashoff, and Claudia Hauff. On the impact of group size on collaborative search effectiveness. *Information Retrieval Journal*, 2019.

[28] Meredith Ringel Morris. A survey of collaborative web search practices. In *CHI '08*, pages 1657–1660, 2008.

[29] Meredith Ringel Morris. Collaborative search revisited. In *CSCW '13*, pages 1181–1192, 2013.

[30] Meredith Ringel Morris and Eric Horvitz. Searchtogether: an interface for collaborative web search. In *UIST '07*, pages 3–12, 2007.

[31] Meredith Ringel Morris, Jaime Teevan, and Steve Bush. Enhancing collaborative web search with personalization: groupization, smart splitting, and group hit-highlighting. In *CSCW '08*, pages 481–484, 2008.

[32] Sharoda A Paul and Meredith Ringel Morris. Cosense: enhancing sensemaking for collaborative web search. In *CHI '09*, pages 1771–1780, 2009.

[33] Jeremy Pickens, Gene Golovchinsky, Chirag Shah, Pernilla Qvarfordt, and Maribeth Back. Algorithmic mediation for collaborative exploratory search. In *SIGIR '08*, pages 315–322, 2008.

[34] Sindunuraga Rikarno Putra, Kilian Grashoff, Felipe Moraes, and Claudia Hauff. On the development of a collaborative search system. In *DESIRES '18*, 2018.

[35] Sindunuraga Rikarno Putra, Felipe Moraes, and Claudia Hauff. Searchx: Empowering collaborative search research. In *ACM SIGIR*, 2018.

[36] Chirag Shah. Collaborative information seeking: A literature review. In *Advances in librarianship*, pages 3–33. Emerald Group Publishing Limited, 2010.

[37] Chirag Shah and Roberto González-Ibáñez. Evaluating the synergic effect of collaboration in information seeking. In *SIGIR '11*, pages 913–922. ACM, 2011.

[38] Chirag Shah, Jeremy Pickens, and Gene Golovchinsky. Role-based results redistribution for collaborative information retrieval. *IPM*, 46(6):773–781, 2010.

[39] Laure Soulier, Chirag Shah, and Lynda Tamine. User-driven system-mediated collaborative information retrieval. In *SIGIR '14*, pages 485–494, 2014.

[40] Laure Soulier, Lynda Tamine, and Wahiba Bahsoun. On domain expertise-based roles in collaborative information retrieval. *IPM*, 50(5):752–774, 2014.

[41] Laure Soulier, Lynda Tamine, and Chirag Shah. Minerank: Leveraging users' latent roles for unsupervised collaborative information retrieval. *Information Processing & Management*, 52(6):1122–1141, 2016.

[42] Lynda Tamine and Laure Soulier. Understanding the impact of the role factor in collaborative information retrieval. In *CIKM '15*, pages 43–52, 2015.

[43] Michael B Twidale, David M Nichols, and Chris D Paice. Browsing is a collaborative process. *IPM*, 33(6):761–783, 1997.

[44] Kazutoshi Umemoto, Takehiro Yamamoto, and Katsumi Tanaka. Scentbar: A query suggestion interface visualizing the amount of missed relevant information for intrinsically diverse search. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 405–414. ACM, 2016.

[45] Ellen M Voorhees. The trec 2005 robust track. In *ACM SIGIR Forum*, volume 40, pages 41–48. ACM, 2006.

[46] Howard W Winger. Aspects of librarianship: a trace work of history. *The Library Quarterly*, 31(4):321–335, 1961.

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

[48] Mu Zhu. Recall, precision and average precision. *Department of Statistics and Actuarial Science, University of Waterloo, Waterloo*, 2:30, 2004.

Appendix A

SearchX Experiment Interface Screen Shots

The following screen shots show the user interface screens that a user encounters during the experiment that we ran.

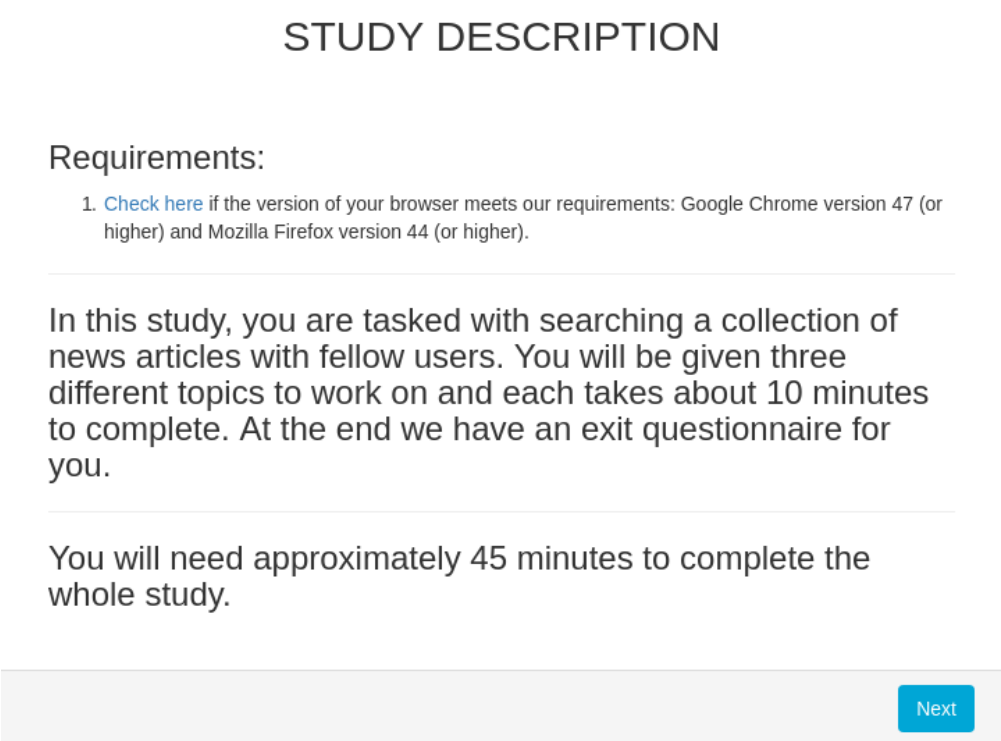


Figure A.1: SearchX experiment landing page.

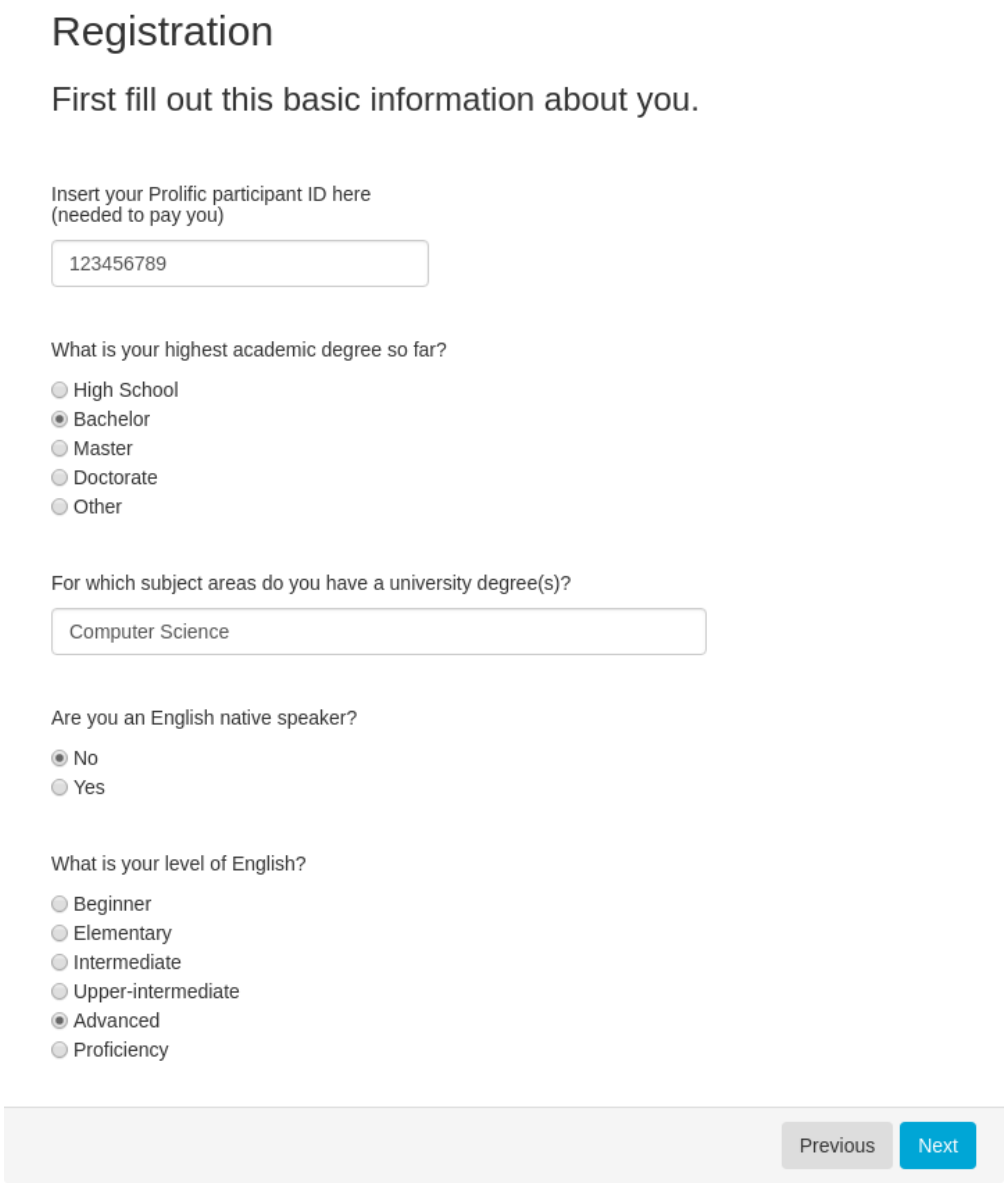


Figure A.2: SearchX experiment user registration.

Collaborative search is when users work together to complete a search task.
Collaborating with other people can take many forms, a few examples are shown here: two people searching together on a single machine, several people searching towards a common goal on separate machines either in the same location or in different locations.

Have you ever collaborated with other people to search the Web?

- ○ No
- ◉ Yes

How often do you engage in collaborative Web search?

- ○ Daily
- ◉ Weekly
- ○ Monthly
- ○ Less often

Think about the most recent time you collaborated with others to search the web.

Describe what were you looking for. (e.g. husband and wife planning a trip for the family, a group of students working on a writing assignment and sharing search results/findings, a couple shopping for a new sofa, etc.)

travel planning

With how many others did you collaborate (not including yourself)?

7

Previous Complete

Figure A.3: SearchX experiment pre-test.

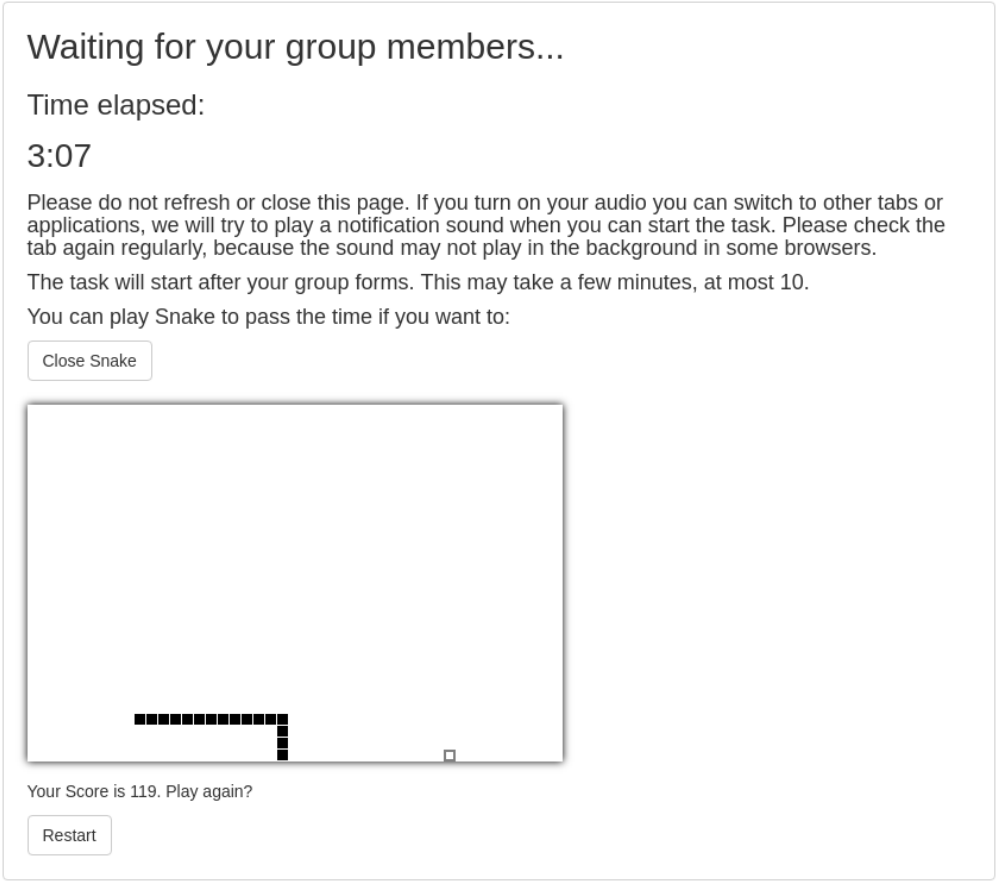


Figure A.4: SearchX experiment waiting room.

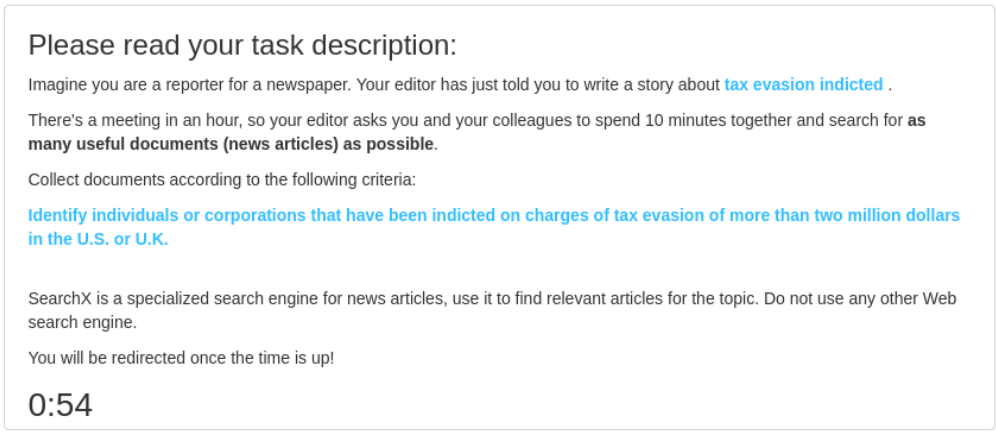


Figure A.5: SearchX experiment task description.

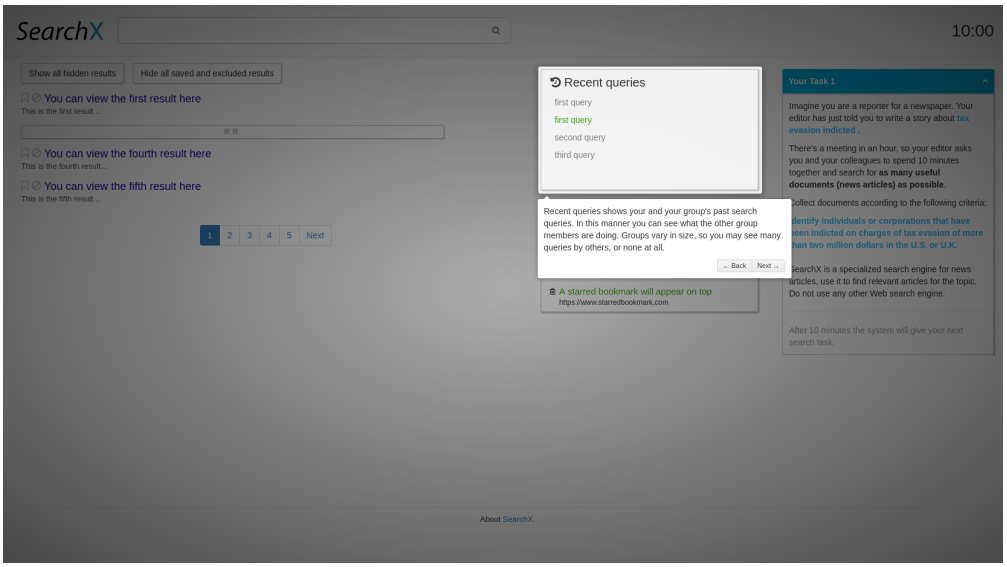


Figure A.6: SearchX experiment introduction step.

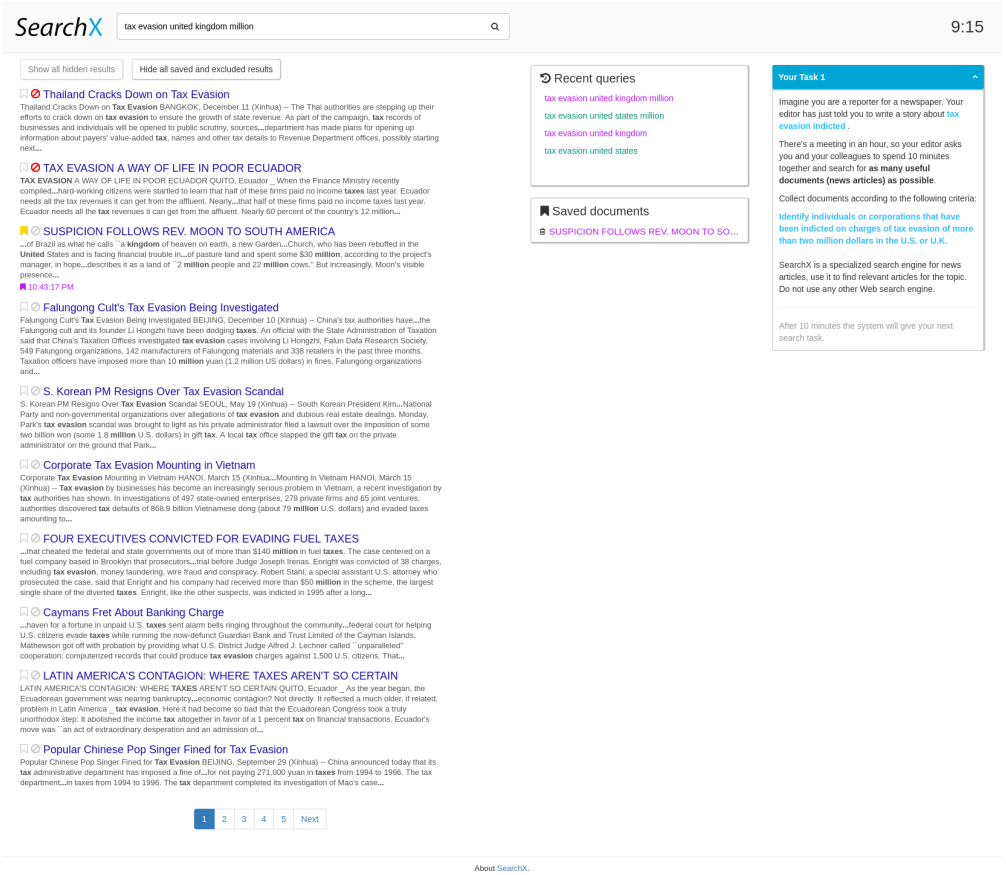


Figure A.7: SearchX experiment search engine results page.

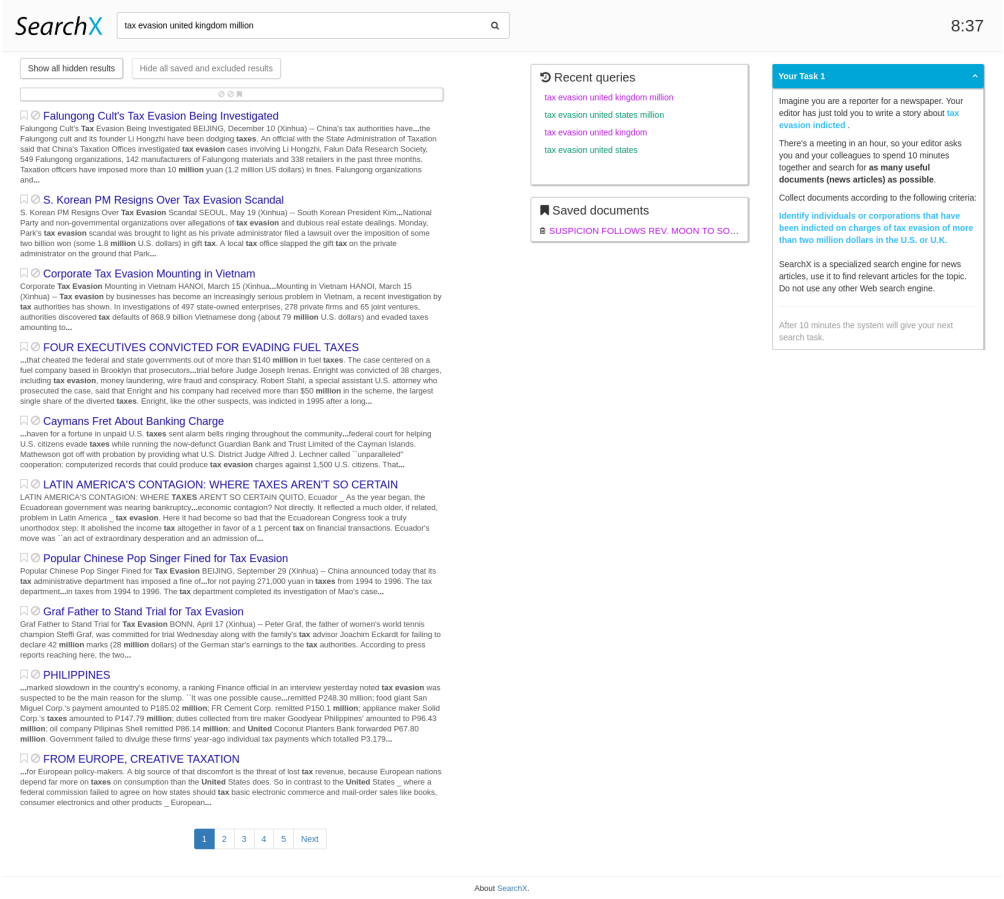


Figure A.8: SearchX experiment search engine results page with collapsed results.

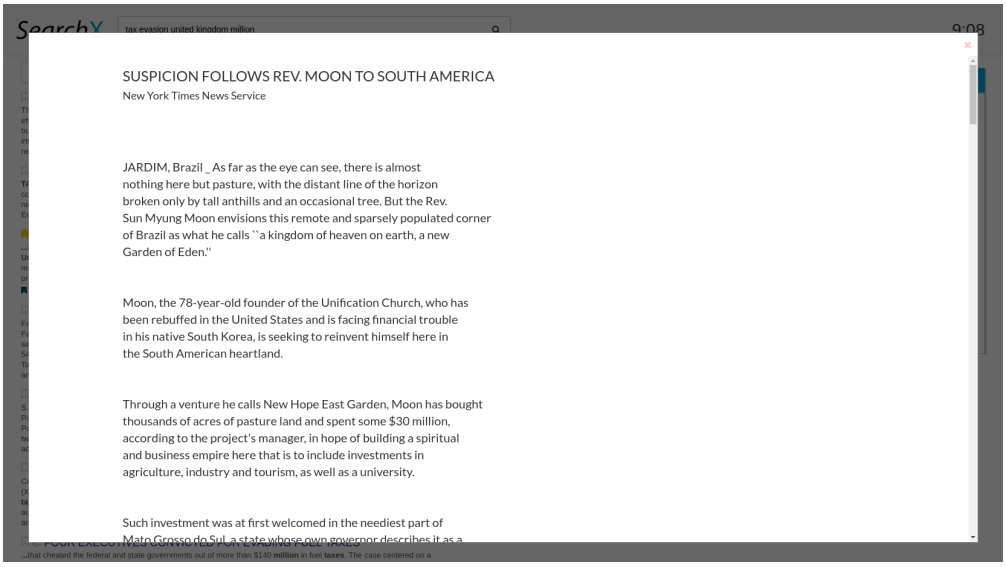


Figure A.9: SearchX experiment document viewer with AQUAINT news article.

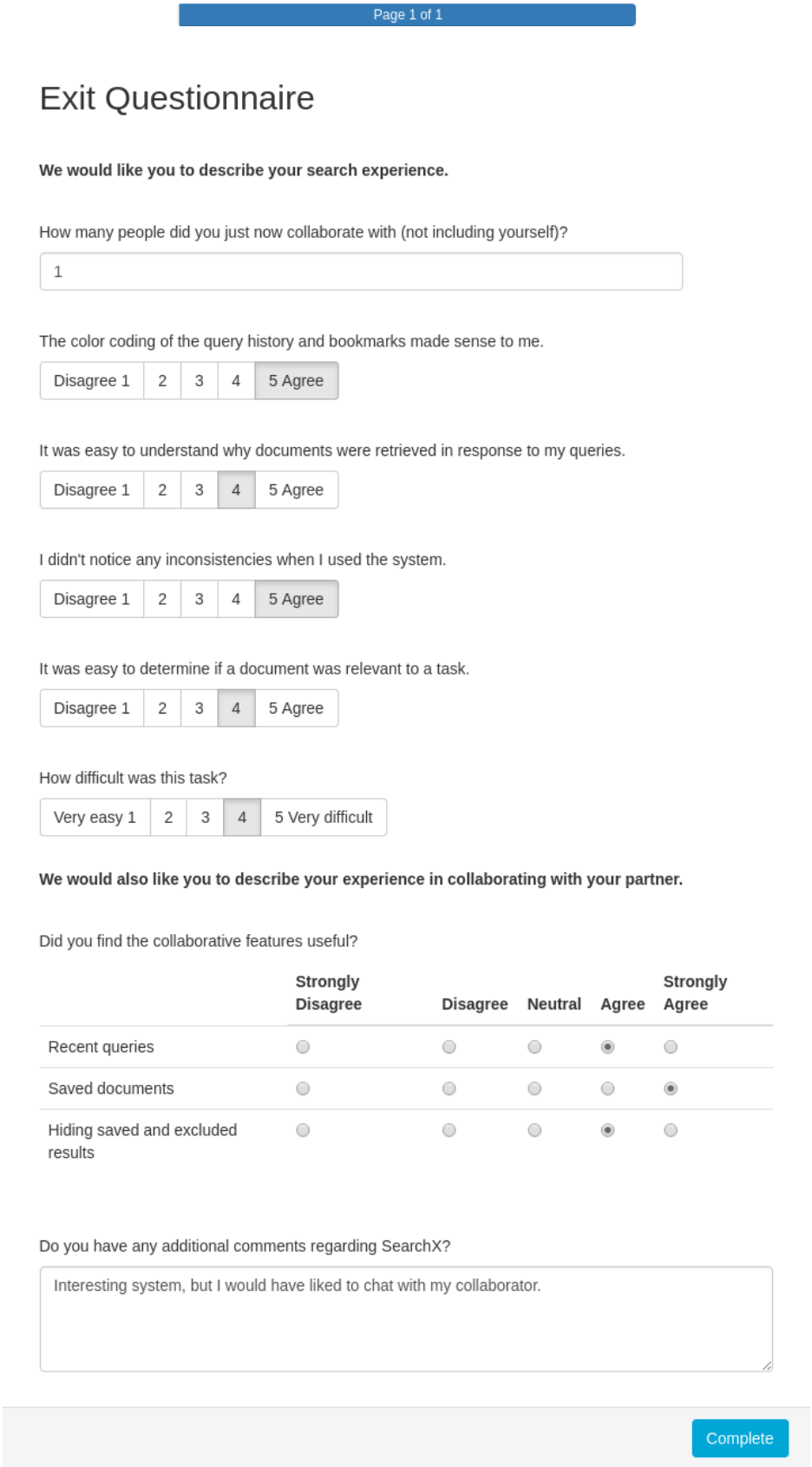


Figure A.10: SearchX post-test questionnaire.

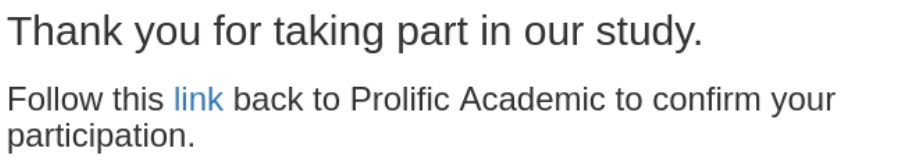


Figure A.11: SearchX experiment thank you screen and confirmation token.