Algorithmic Mediation for Collaborative Exploratory Search

Jeremy Pickens[†], Gene Golovchinsky[†], Chirag Shah[‡], Pernilla Qvarfordt[†], Maribeth Back[†]

FX Palo Alto Laboratory, Inc.[†]
3400 Hillview Ave, Building 4
Palo Alto, California 94304 USA
{jeremy,gene,pernilla,back}@fxpal.com

School of Information and Library Science[‡]
University of North Carolina
Chapel Hill, NC 27599 USA
chirag@unc.edu

ABSTRACT

We describe a new approach to information retrieval: algorithmic mediation for intentional, synchronous collaborative exploratory search. Using our system, two or more users with a common information need search together, simultaneously. The collaborative system provides tools, user interfaces and, most importantly, algorithmically-mediated retrieval to focus, enhance and augment the team's search and communication activities. Collaborative search outperformed *post hoc* merging of similarly instrumented single user runs. Algorithmic mediation improved both collaborative search (allowing a team of searchers to find relevant information more efficiently and effectively), and exploratory search (allowing the searchers to find relevant information that cannot be found while working individually).

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Search Process

General Terms

Retrieval Models, Interactive Retrieval

Keywords

Collaborative Search, Algorithmic Mediation, Evaluation

1. INTRODUCTION

Information seeking can be more effective as a collaboration than as a solitary activity: different people bring different perspectives, experiences, expertise, and vocabulary to the search process. A retrieval system that takes advantage of this breadth of experience should improve the quality of results obtained by its users [4].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SIGIR'08, July 20–24, 2008, Singapore.
Copyright 2008 ACM 978-1-60558-164-4/08/07 ...$5.00.

There are many forms of collaboration in search, such as community-based recommendation systems [14] or user interfaces that allow multiple people to compose queries [9] or examine search results [13]. In this work we explore the possibilities of synchronous, explicit, *algorithmically-mediated* collaboration for search tasks [10]. We describe a retrieval system wherein searchers, rather than collaborating implicitly with anonymous crowds, collaborate explicitly (intentionally) with each other in small, focused search teams. Collaboration goes beyond the user interface: Information that one team member finds is not just presented to other members, but it is used by the underlying system in real-time to improve the effectiveness of all team members while allowing each to work at their own pace.

This is an important new direction for search collaboration that can lead to innovation in information retrieval algorithms and in user interfaces. Toward this end we present an initial implementation, the first of many possible systems, integrating algorithmic mediation and intentional collaboration, and apply it to the *ad hoc* information retrieval task. The design comprises a set of user interfaces, a middleware layer for coordinating traffic, and an algorithmic back-end optimized for collaborative exploratory search.

We evaluated the effect that algorithmic mediation has on collaboration and exploration effectiveness. Using mediated collaboration tools, searchers found relevant documents more efficiently and effectively than when working individually, and they found relevant documents that otherwise went undiscovered.

2. BACKGROUND

We distinguish several kinds of collaboration in the context of information retrieval. Collaborative filtering is an example of asynchronous and implicit collaboration; aggregate crowd behavior is used to find information that previous users have already discovered [14]. Collaborative filtering for search has two weaknesses: first, there are often many documents in a collection that have received little prior user attention, reducing the likelihood that they will be retrieved; and secondly, the aggregate information need of the crowd might not match the specific needs of the current searcher(s). For example, collaborative filtering algorithms in online shopping web sites may recommend products that you already have, or that are not appropriate to your climate or well-suited to your tastes.

The term "collaboration" has also been used to refer to

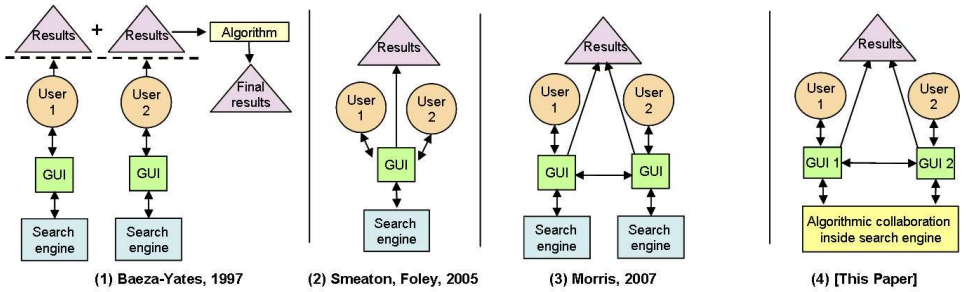

Figure 1: System Architecture Comparisons

synchronous, intentionally-collaborative information seeking behavior. Such systems range from multiple searchers working independently with shared user interface awareness [9] to multiple people sharing a single user interface and cooperatively formulating queries and evaluating results [13]. Collaborative web browsing and real-time sharing of found information, through purpose-built user interfaces rather than through email, is common in these systems [15]. A major limitation of existing synchronous approaches is that collaboration is restricted to the user interface. Searchers are automatically notified about the on-going activities of their teammates, but to take advantage of that information to improve their searches, each user must manually examine and interpret teammates' queries and found documents. While awareness of one's co-searcher(s) is an important first step for collaborative retrieval, user interface-only solutions still require too much attention to others' results. In these user interface-only systems, searchers must manually reconcile and integrate their activities with their co-searcher(s).

Nevertheless, the synchronous, intentional approaches mentioned above are closely related to the system described in this paper. Figure 1 shows the structural differences in architecture between three user interface-only collaborative IR systems ([4], [13], and [9]) and the algorithmically mediated approach. All three earlier systems use search engines that are not aware of the ongoing collaboration. As each query arrives at the engine, it is treated as a new, separate search. Although searchers may collaborate at the user interface and interpersonal level, the search engine itself does not support collaboration. This is true whether each searcher uses a separate search engine, or if they share a search engine as in Físchlár-DiamondTouch [13]. In SearchTogether [9], a searcher's activity is not used by the underlying engine to influence the partner's actions; all influence happens in the interface or live communication channels. In contrast, an algorithmically-mediated collaborative search engine coordinates user activities throughout the session.

3. SINGLE-PASS COLLABORATION

Collaborative exploratory search is an iterative, interactive process by nature. Searchers leverage each others' results as they explore a particular topic. To understand each cycle of this iterative process, we performed an experiment to characterize the effect of fusing search results from multiple searchers.

To create our test collection, we extracted terms from the description field of TREC topics 301-450, and ran these terms as queries to identify potential topics. We then selected all topics of moderate difficulty (precision@10 values between 0.1 and 0.5). This yielded 53 topics, from which we randomly removed 3 to produce 5 groups of 10 topics. For each group, we generated a paper form with topic descriptions and instructions asking users to write the query they would issue to a typical search engine for the specified topics. Fifteen subjects were asked to fill forms at their leisure; three queries were collected for each topic.

We used Borda count fusion [2] to merge different queries on the same search engine, an approach also suggested by Shaw and Fox [12], and Belkin *et al.* [5]. As with most fusion approaches, the goal is to increase the range of information from which relevance can be inferred.

To simulate single-iteration collaboration among a set of users we first ran the three user queries (A, B, and C) for each topic individually. Next, we fused the ranked lists two ways (AB, AC, BC) and three ways (ABC). For each condition we computed the average recall and precision scores, as shown in Table 1. In line with Shaw and Fox [12] and Shah *et al.* [11], these results show that query fusion led to improved performance and that more formulations of an information need led to further improvements. This suggests that although ranked list fusion is not the only possible foundation for collaborative algorithms, it is a reasonable starting point.

4. ITERATIVE COLLABORATION

While Borda fusion of query results showed good performance, by itself this algorithm cannot be used effectively in an iterative setting, because both searchers would need to issue their queries at the same time, to finish browsing their results lists at the same time, and then to issue the next query at the same time. This is the inherent limitation faced by user interface-only architectures, such as the

	Indiv	2-Way	%Chg	3-Way	%Chg
Average precision (non-interpolated)					
	0.0931	0.1064	**14.26⋆**	0.1100	**18.14⋆**
Precision:					
At 5:	0.3133	0.3493	+11.5	0.3360	+7.2
At 10:	0.2707	0.2967	+9.6	0.3000	**+10.8⋆**
At 15:	0.2391	0.2671	**+11.7⋆**	0.2653	**+11.0⋆**
At 20:	0.2157	0.2443	**+13.3⋆**	0.2460	**+14.1⋆**
At 30:	0.1856	0.2049	**+10.4⋆**	0.2133	**+15.0⋆**
At 100:	0.1078	0.1175	**+9.0⋆**	0.1264	**+17.3⋆**
At 200:	0.0734	0.0800	+9.0	0.0872	**+18.7⋆**
At 500:	0.0427	0.0479	**+12.4⋆**	0.0523	**+22.6⋆**
At 1000:	0.0265	0.0305	**+15.0⋆**	0.0327	**+23.5⋆**

Table 1: Precision for single-query runs vs. precision for 2-way and 3-way fused runs. ⋆ indicates t-test significance at $p < 0.01$.

Físchlár-DiamondTouch system [13] and the SearchTogether system [9] used in split mode.

An algorithmically-mediated collaboration framework should allow people to work at their own pace but still be influenced in real-time by their partners' search activities. *Influence should be synchronized, but workflow should not.* If one user decides to issue a new query, the second user should not be interrupted in his or her activity. At the same time, the second user should start to see the influence of the first user's new search activity when the second user makes a request to the collaborative back-end, and vice versa.

To test our ideas about algorithmic mediation for collaborative exploratory search, we implemented a proof of concept system built for video search. The design of the system was based partially on lessons learned from best-of-breed instances of video search interfaces [8], and partially from observations and studies we performed as part of the design process. We chose the TRECvid search task partly because it provides an interesting complex search task involving several modalities (text, image, and concept similarity) and partly to leverage existing experience (e.g. the MediaMagic interface, described below) within our laboratory.

While the system was built for the 2007 TRECvid interactive video search track, we emphasize that the underlying mediation algorithms supporting this task are generic and may be applied to all types of retrieval: text, video, images, music, etc. Our search engine (based on Adcock *et al.* [1]) algorithmically mediates a wide variety of queries, including text queries, fuzzy text queries (text-based latent sematic concept expansion), image similarity queries based on color histograms, and image-based concept similarity, via statistical inference on semantic concepts of images.

In this section we will describe how the system combines multiple iterations from multiple users during a single search session. The system consists of three parts: (1) user interfaces that implement the roles, (2) the architecture to support these roles, and (3) algorithms used to perform collaborative search.

4.1 Search Roles

The synchronous and intentional nature of the collaboration enables searcher specialization according to roles and/or tasks. Many roles and associated task types are possible; these may shift over time or during different parts of the search task. Roles may be equal, hierarchical, partitioned (separated by function), or some combination thereof. User interfaces, tools, and algorithms may offer commands or perform actions specific to particular roles.

Our current system allows collaborating users to assume the complementary roles we dubbed Prospector and Miner. The Prospector opens new fields for exploration into a data collection, while the Miner ensures that rich veins of information are explored. These roles are supported by two different user interfaces and by underlying algorithms that connect the interfaces. Unlike approaches in which roles are supported manually or only in the user interface [9], these roles are built into the structure of the retrieval system. The regulator layer (described below) manages roles by invoking appropriate methods in the algorithmic layer, and routing the results to the appropriate client.

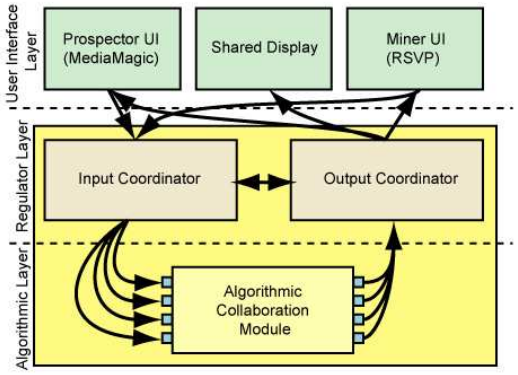

Figure 2: Collaborative System Architecture

4.2 System Architecture

The system architecture consists of three parts: the User Interface Layer, the Regulator Layer, and the Algorithmic Layer (Figure 2). System components communicate through a web service API, and can be combined in different ways: the single shared display in a co-located setting can be replaced by separate displays in remote locations, showing the same information.

4.2.1 User Interface Layer

Our system contains three user interfaces: (1) A rich query user interface (MediaMagic [1, 6]) for use of the Prospector, (2) a rapid serial visualization result browsing user interface (RSVP) for use of the Miner, and (3) a shared display containing information relevant to the progress of the search session as a whole.

The MediaMagic user interface contains tools for issuing queries (text, latent semantic text, image histogram, and concept queries), displays ranked results lists and has an area for viewing and judging retrieved shots. The RSVP user interface is primarily designed for relevance assessment of video shots, which are presented in a rapid but controllable sequence. However, the RSVP user interface also includes the capability for Miners to interrupt the flow of shots to issue their own text queries.

Finally, a shared display shows continually-updating information about issued queries, all shots marked as relevant by either user, and system-suggested query terms based on activities of both users. In our setting, the shared display was shown on a large screen easily viewed by both the Prospector and the Miner (Figure 3, top center).

4.2.2 Regulator Layer

The regulator layer consists of an input regulator and an output regulator. The input regulator is responsible for capturing and storing searcher activities, such as queries and relevance judgments. It contains coordination rules that call the appropriate algorithmic collaboration functions. The input regulator implements policies that define the collaborative roles. Similarly, the output regulator accepts information from the algorithmic layer and routes it to appropriate clients based on their roles. The regulator works autonomously, and does not interact directly with users.

4.2.3 Algorithmic Layer

The algorithmic layer consists of a number of functions for combining searchers' activities to produce documents,

Figure 3: A collaborative search session. Each user's UI is suited to their role: Prospector (left) and Miner (right). Large side screens show sample relevant shots for the current topic; center screen shows the shared query state.

rankings, query suggestions, and other information relevant to the search. It performs basic searches, and generates raw search results, transformed search results based on input from multiple users, and query terms that characterize the current state of the collaboration. Details of these algorithms are discussed in the following section.

4.3 Algorithmic Mediation

In Section 3 we described a single-pass algorithm for merging ranked lists; in this section, we extend the approach to support iterative use. We define two weighting variables, relevance $w_r(L_k)$ and freshness $w_f(L_k)$. These are functions of a L_k, a ranked list of documents retrieved by query k.

$$w_f(L_k) = \frac{|unseen \in L_k|}{|seen \in L_k|} \quad (1)$$

$$w_r(L_k) = \frac{|rel \in L_k|}{|nonrel \in L_k|} \quad (2)$$

The query freshness weight w_f is given by the ratio of unseen (retrieved by the engine, but not yet manually examined) to seen (retrieved and manually examined) documents in L_k. The query relevance weight w_r is given by the fraction of seen documents that were judged relevant for that query. These two factors are designed to counterbalance each other. If a query retrieved many relevant documents, it should have a high relevance weight, but once most of the documents from a query have been examined, other queries should have higher priority given by the freshness weight. These weights are updated continuously based on searchers' queries and judgments of relevance. The weights are then used to affect the information shown to each searcher, as appropriate to their roles.

4.3.1 Miner Algorithm

As mentioned in Section 4.1, the RSVP client acts in the role of Miner. The regulator accumulates documents retrieved by all team members during a session. Documents not yet examined by the Prospector are queued for the Miner based on freshness and relevance weights. The queue is ordered by a score that uses Borda fusion to merge the contributions of all queries, as shown in Equation 3.

$$score(d) = \sum_{L_k \in \{L\}} w_r(L_k) w_f(L_k) borda(d, L_k) \quad (3)$$

The Prospector continually adds new ranked lists L_k to the set L, and views and judges documents. Meanwhile, the Miner judges highly-ranked unseen documents d. These documents are likely to have appeared in more than one list L_k; therefore, relevance judgments made on these documents affect the w_f and w_r weights of more than one list and further change overall priorities. The Miner does not have to manually decide which documents to comb through, nor does the Prospector have to decide which documents to feed to the Miner.

The Miner algorithm is similar to the on-line hedge algorithm for ranked list fusion [3]. Both approaches share the intuition that attention should shift to those lists that show themselves to be more "trustworthy." In our work, however, the ranked lists being combined are from different queries, rather than from different search engines. Furthermore, the number of ranked lists is not static. As both users issue queries, the number of rank lists grows over time.

4.3.2 Prospector Algorithm

The previous section describes how algorithmic mediation allows the Miner to work with the Prospector. But how does the Prospector algorithmically influence the Miner? The unseen documents priority score in Equation 3 only affects the ordering of documents for the Miner. We chose not to apply the same transformation to the Prospector's search results because we wanted the Prospector to see the raw effectiveness of each query. If previously retrieved but unseen documents were retrieved again by a new query, that would boost their priorities in the Miner's queue; those documents would likely soon receive attention by the Miner rather than the Prospector.

We choose to let the Prospector focus on coming up with new avenues for exploration into the collection. This is accomplished by a real-time query term suggestion feed from which the Prospector can get a sense of how the overall search is progressing and draw ideas about new avenues to explore. The basic idea is similar to the Miner algorithm. However, instead of a Borda count on unseen documents d, we use a "ranked list frequency" count on terms t: $rlf(t, L_k)$. This is defined as the number of documents in L_k in which t is found. Using rlf, we define a score for every term, t:

$$score(t) = \sum_{L_k \in \{L\}} w_r(L_k) w_f(L_k) rlf(t, L_k) \quad (4)$$

This function updates continuously. Terms used in previous queries are filtered out to produce a list of top ten terms that are shown on the shared display. As the Miner's activity affects the w_r and w_f weights, the system re-orders or replaces term suggestions. The more the Miner digs into fresher and more relevant pathways, the more terms associated with those pathways appear. Once a particular avenue loses freshness or does not exhibit relevance and the Miner switches from that path, the automatically-suggested terms switch accordingly. No explicit actions are required for the

Miner to suggest terms to the Prospector, just as no explicit actions are required from the Prospector to feed documents to the Miner; the collaborative retrieval algorithm handles the flow of information between the users. The Miner and Prospector are self-paced in their respective workflows, but the influence that each exerts on the other is synchronous.

5. EXPERIMENTAL EVALUATION

We conducted an experiment to evaluate our collaborative exploratory search system. This experiment was conducted within the framework specified by NIST's 2007 TRECvid interactive track. We hypothesized that collaborative search would produce better results than *post hoc* merging (see the Baeza-Yates [4] architecture in Figure 1) with respect to average precision and average recall. We also discovered that collaborating users found more unique relevant documents. A thorough investigation of the effects of collaborative system roles on search performance is beyond the scope of this paper; for now we only compare the end effects of two-person collaboration against two-person *post hoc* results merging. Section 6.2 suggests a few directions that additional role analyses might take.

5.1 Method

We used a mixed-design experimental method, where teams of searchers performed one of two search conditions (merged or collaborative); all 24 TRECvid interactive retrieval topics were used in both conditions. In total, eight people participated in the experiment. All are researchers in our lab, but only half of them conduct research related to information retrieval. The experiment was run during regular work hours and the participants did not receive additional compensation for their participation. No participants had prior experience with the topics on which they searched. In both conditions teams had 15 minutes to complete a topic; each condition consisted of two team members working for 15 minutes, for a total of 30 person-minutes per topic.

The teams in the collaborative condition consisted of two co-located participants with different levels of experience in multimedia information retrieval; one had prior experience (EXP) and one did not (NEX). The EXP used the MediaMagic client, while the NEX used the RSVP client. Prior to the runs, both team members received training on their respective user interfaces, general instruction on their expected roles, and how the system implemented each role. Verbal and non-verbal communication between participants was not controlled; participants were free to discuss the topic as necessary. Each EXP user worked and equal amount with each NEX user; switching team members balanced the influence of individual differences on team performance. Although it would be interesting to explore role interaction effects, we did not have enough subjects to pursue that analysis.

The merged condition consisted of *post hoc* unions of pairs of search runs with no interaction or algorithmic mediation. This condition mirrored the collaborative condition in terms of the composition of the teams: each topic was completed by a team with one EXP and one NEX user, using the same amount of overall time. All team members received training on, and used, the MediaMagic client. Teams were also swapped to balance individual influences.

In both conditions the same underlying indices and similarity functions were used. The primary difference was that the interfaces in the *post hoc* merged condition were connected to the stand-alone retrieval engine, whereas in the collaborative condition the interfaces were connected to the algorithmically-mediated collaborative engine.

5.1.1 Relevance judgments

The collaborative condition was originally designed for the TRECvid 2007 competition; its results were submitted to the NIST relevance judgment pool. However, the *post hoc* merging runs were completed later. In these later runs, individual searchers selected a small number of documents that had not been found by any other search system and therefore had not been judged by NIST. In order to avoid incorrectly penalizing these latter runs through incomplete ground truth, four judges independently assessed documents unique to the merged condition for topical relevance. Disagreements among judges were resolved in a joint judging session. In all, 41 relevant documents were added to the ground truth[1] through this process.

5.1.2 Ranked List Padding

During a 15 minute interactive run, searchers manually select only a limited number of documents, far below the allowed TRECvid submission limit of 1,000. It is therefore common to do a last-second relevance feedback run to fill the final results list. However, we believe the results from this procedure do not reflect true interactive performance. Therefore, while we performed this step as part of our competition submission, for the analysis in this paper we only include manually identified documents.

5.1.3 Metrics

In evaluating the effectiveness of interactive search, we need to distinguish between documents returned by the search engine, documents actually seen by the user, and documents selected by the user [7]. Thus, we use *viewed precision* (P_v, the fraction of documents seen by the user that were relevant), *selected precision* (P_s, the fraction of documents judged relevant by the user that were marked relevant in the ground truth), and s*elected recall* (R_s) as our dependent measures.

5.2 Results

We wanted to test the hypothesis that mediated collaboration offers more effective searching than *post hoc* merging of independently produced results, as was done, for example by Baeza-Yates *et al.* [4].

5.2.1 Collaboration

To assess the teams' performance, we removed duplicate documents from the merged result set, and kept track of which documents each person saw (whether they judged them or not). Participants in the merged condition saw an average of 2978 distinct documents per topic (both relevant and non-relevant); participants in the collaborative condition saw an average of 2614 distinct documents per topic. For each topic, we subtracted the merged score from the collaborative score and divided by the merged score. We also split runs up over time (3.75, 7.5, 11.25 and 15 minutes).

We found that collaborative search consistently outperformed merged search on our metrics as shown in Table 2

[1]20 new documents for topic 199, 11 for topic 213, 6 for 206, 3 for 214, and 1 for 209; the total went from 4704 to 4745

and in Figure 4. For example, at the end of the 15 minute session, $\bar{R}_s$ was 29.7% higher for collaborative search than for merged results. Collaborative search exhibited better performance throughout the session.

	3.75 min	**7.5 min**	**11.25 min**	**15 min**
	Avg%Chg	Avg%Chg	Avg%Chg	Avg%Chg
P_s				
Overall	+9.8	+21.5	+22.4	+30.2
Plentiful	-2.6	+6.1	+4.2	+0.4
Sparse	+22.4	+36.8	+40.7	+60.1
R_s				
Overall	+15.2	+35.7	+19.2	+29.7
Plentiful	+13.9	+13.5	+3.8	-4.4
Sparse	+16.4	+57.9	+34.7	+63.8
P_v				
Overall	+13.6	+65.4	+41.1	+51.1
Plentiful	+16.6	+9.1	+2.3	-9.7
Sparse	+10.6	+121.6	+79.9	+111.9

Table 2: Average percent improvement of collaborative over merged, at various time points

Overall results of the experiment indicate a consistent advantage for collaborative search over merged results from independent single-user searches. We wanted to understand the differences in more detail, and thus looked at the effect that the number of relevant documents for a topic had on our results. We divided the 24 topics into two groups based on the total number of relevant documents available for that topic. Topics that fell below the median (130) were deemed "sparse" (average of 60 relevant documents per topic) and those above were "plentiful" (average of 332 relevant documents per topic).

For "sparse" topics, users in the merged condition saw on average 3787 unique documents vs. 2877 in the collaborative condition; for "plentiful" topics, users in the merged condition saw on average 2168 documents vs. 2352 for the collaborative condition. We then repeated our analysis on the two groups independently; results are shown in Table 2, and compared in Figure 4.

We now see that for plentiful topics, collaborative search is comparable to merging individual results: the white bars in Figure 4 are small, and the error bars span 0. If relevant documents are abundant, anybody can find them! When the topics are not so obvious, however, collaborative search produces dramatically better results on average. The gray bars are consistently above 0 and are consistently larger in magnitude than the solid "overall" bars. For viewed precision (P_v) in particular, we saw an increase of over 100% compared to the merged condition, despite the fact that participants in the merged condition saw 910 more documents overall than in the collaborative condition. A repeated-measures ANOVA of P_v confirmed that this interaction between time and topic sparsity ($F(3, 66) = 3.69$, $p < 0.025$) was significant, indicating that improvements over the course of a session were unlikely to be due to chance.

Although we did not design the experiment to quantitatively measure the effect of oral communication between team members, we analyzed the video record of experiments to assess the degree to which non-mediated communication channels (e.g. gaze) were used by our participants. We found that the Miner spent on average 5.3 seconds (SD=6.4) of a 15 minute session looking at the Prospector's screen, while the Prospector spent 4.2 seconds (SD=3.7) looking at the Miner's screen. Given that the system's algorithmic mediation, not the Prospector, determined the documents and their order of presentation to the Miner, and given that Miner found on average 38% of the relevant documents, it is unlikely that non-algorithmic channels played a large part in the overall team performance.

These results suggest that algorithmically-mediated collaborative teams were much more efficient than people working individually at detecting relevant documents.

5.2.2 Exploration

In addition to effectiveness, we wanted to see how well exploration was supported by algorithmic mediation. We wanted to assess the comparative effectiveness of collaborative versus *post hoc* merged search in finding unique relevant documents, documents that only one system was able to find. While some retrieval tasks are considered a success if any relevant document is found, sometimes it is appropriate to examine *which* relevant documents are found.

Head-to-head comparisons between two systems for uniqueness are problematic. Without an external baseline, uniqueness is equivalent to the difference in the number of relevant documents retrieved. Thus we chose to use the data submitted by the 10 other TRECvid participants as the baseline.

If two interactive runs are based on the same low-level indexing and retrieval code, they will likely retrieve similar documents because of those shared retrieval mechanisms. Each TRECvid group was allowed to submit several runs, all of which were used by NIST to determine relevant documents. Some of these runs were produced by parametric variation and retrieved many of the same documents. Therefore, to get a more accurate count of unique documents, we kept only the best-performing (in terms of MAP) run from each group. This gave us a "background" set of 10 runs, the union of which served as our baseline.

Our two runs, collaborative and *post hoc* merged, are also based on the same low-level indices and document similarity functions. Therefore, we compare each condition separately against the baseline. We first evaluate the collaborative system against the baseline by computing for each topic N_u the number of unique relevant documents identified by each of the 11 systems: 1 collaborative and 10 background systems. We also compute F_u, the number of unique relevant documents as a fraction of the number of relevant documents found by a system. We repeated the analysis using the merged run results.

The 2007 TRECvid competition for *ad hoc* search consisted of 24 topics (0197-0220); the number of relevant documents varied greatly among the topics ($range = [6, 1150]$, $\bar{x} = 196$, $median = 130$, $\sigma = 234$ for the original TRECvid data, $range = [6, 1170]$, $\bar{x} = 198$, $median = 130$, $\sigma = 238$ for the augmented ground truth). This large variability made it problematic to average performance across topics, so for each topic, we ranked the scores (N_u and F_u) of the various runs, and averaged ranks rather than the raw scores across topics. Finally, we compared the average ranks we obtained from the merged runs with the collaborative runs, as shown in the rightmost columns of Table 3.

We were also interested in the temporal profile of our sessions with respect to finding unique documents. Did people find unique relevant documents after they found documents found by other systems, or were they sprinkled throughout the session? For this analysis, we again used the other

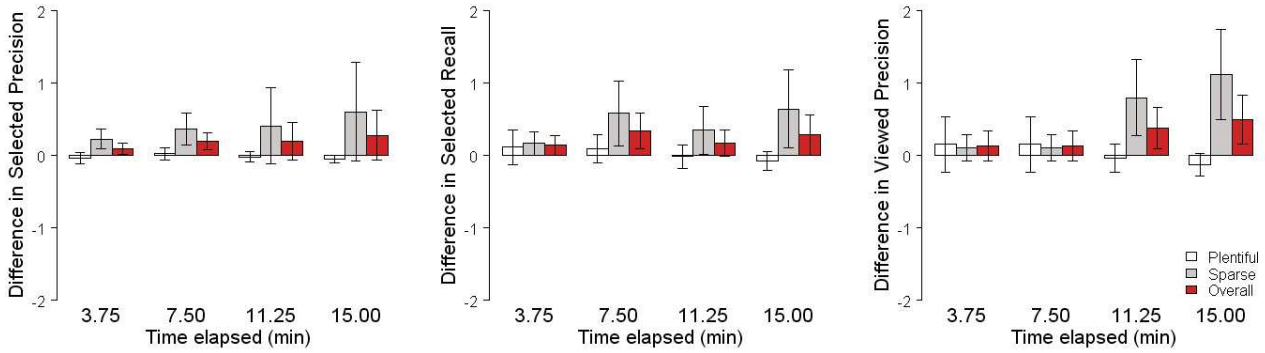

Figure 4: Plentiful/Sparse Split: Selected Precision, Selected Recall, Viewed Precision. Error bars represent ± 1 standard error.

groups' data as a baseline, but rather than comparing our complete runs, we used subsets of our data collected through 3.25, 7.5 and 11.5 minutes of a 15 minute session. We compared our fractional data to the full data of other systems because we were not interested in comparing directly against other systems, but in comparing between the collaborative and merged systems. The baseline data served as background, as a relatively unbiased estimate of easy to discover documents that were relevant to each topic. Given that baseline, how do our two systems compare when finding new unique documents?

As before, we performed the analysis on all topics, and then by topic size (plentiful/sparse). The results are summarized in Table 3. The data for overall uniqueness show a slight advantage for collaborative search at the end of the 15 minute session (a rank improvement of 2-3%), but that is a small gain. When we split the topics by sparsity, however, a different pictures emerges. For plentiful topics, there is no effective gain for the collaborative system (a rank improvement of 0-2%), whereas for the sparse topics there is a 12-14% uniqueness rank improvement.

Over time, there appears to be no strong trend for either metric either overall or for plentiful topics, whereas for sparse topics, improvements accrue quickly, and then reduce a bit at the end. This suggests that collaborative search teams find more unique relevant documents than merged results from individual searchers, that the advantage is greater for difficult topics, and that for those topics, there is more advantage earlier in the search session than later.

As with recall and precision measures discussed in the previous section, these results suggest that mediated collaboration is more effective when the search topics are sparse. While a more principled exploration of the effect of topic difficulty on system performance is warranted, the trend is encouraging.

6. FUTURE WORK

The work described in this paper represents some initial steps in exploring the design space of algorithmically-mediated information retrieval collaboration. While our initial results are encouraging, much remains to be explored. We are pursuing several broad directions, including understanding the range of roles and the sorts of algorithms and interfaces that support them. We are also looking at the relative contributions of team members in terms of roles and how best to allocate scarce human resources.

6.1 Roles and interfaces

Roles are a critical part of our system: they inform the design of user interfaces, and determine the algorithms that the regulator and back end use to retrieve and represent search results. Our initial implementation focused on the roles of Prospector and Miner, but other roles and associated algorithms and interfaces may also be useful.

6.2 Evaluation of role relationships

We observed that the collaborative system resulted in better performance than *post hoc* merged results of independent searches. But what exactly contributed to these improvements? Did the team find more relevant documents because the Miner to dug deeper and found them? Or was the Miner able to dig deeper, because some of the nuggets the Miner found allowed the Prospector to ask better questions? Or did the Prospector explore more because he or she knew that the Miner would find what he missed? These are important questions for future system design, and they may require additional experiments to answer definitively.

Another possible analysis involves determining the optimal size of a team. We ran pairs of participants; would adding a second Miner or Prospector improve the results? Would it improve the results compared to merging three independent runs? Are there situations in which a person is better off working alone, or having an asymmetric relationship with other roles (e.g. are there times when a Prospector's explorations should not affect the Miner's work)?

6.3 Algorithms

Our experiments were conducted on TRECvid data; the task was to find video shots that matched the given topic description. The data were indexed in a variety of ways: text, LSA text, image histogram, and image inferred-concept. Our initial collaboration algorithms therefore focused on a method that could handle queries and results across this wide variety of data types: results list fusion. Future work will explore possibilities that arise by restricting the search to a single data type, i.e. text-only or image-only. In those narrower situations it should be possible to create collaboration algorithms based on the intrinsic content of the infor-

	3.75 minutes			7.5 minutes			11.25 minutes			15 minutes		
	Merge	Collab	%Chg	Merge	Collab	%Chg	Merge	Collab	%Chg	Merge	Collab	%Chg
Average Rank of N_u												
Overall	7.61	7.00	+8.02	6.90	6.09	+11.7	5.88	5.74	+2.38	5.54	5.35	+3.43
Plentiful	8.92	8.83	+1.01	8.33	8.33	0.00	7.83	8.08	-3.19	7.67	7.50	+2.22
Sparse	5.00	4.56	+8.80	4.50	3.64	+19.1	3.92	3.18	+18.9	3.42	3.00	+12.28
Average Rank of F_u												
Overall	5.94	6.48	-9.09	5.60	5.26	+6.07	4.83	5.18	-7.25	5.08	4.96	+2.36
Plentiful	6.42	8.08	-25.9	6.33	6.92	-9.32	5.92	7.17	-21.1	6.75	6.75	0.00
Sparse	5.00	4.33	+13.4	4.50	3.45	+23.3	3.75	3.00	+20.0	3.42	2.92	+14.6

Table 3: Average ranks for uniqueness measures. Smaller scores represent better performance. The "%Chg" column represents the percent improvement (decrease in average rank) of collaborative over merged.

mation being queried and retrieved. For example, instead of the Miner digging through the best unseen documents that result from a Prospector's query stream, the Miner might instead see a continually-updating variety of semantic facets. The Prospector might not even be aware that a certain untapped, potentially relevant facet is accruing in his or her retrieval activities. But a collaborative algorithm might analyze the unseen content and surface those facets to the Miner, synchronously.

7. CONCLUSION

We designed, built, and evaluated a system that mediates search for a focused team of searchers. Our evaluation showed that this instantiation of mediated collaboration improved selected precision, selected recall, viewed precision, and the number of unique relevant documents found compared with naive merging of search results obtained independently by two searchers. Although different media types may require different user interfaces to elicit queries and to display results, the underlying mediation need not change because this particular mediation algorithm, supporting these roles, is content-domain independent.

This paper described a system that is one possible instantiation of a more general concept. Numerous challenges remain, including designing and comparing different real-time merging strategies for query results, defining additional roles, better understanding the tradeoffs between parallel and synchronized work, and designing appropriate user interfaces. Overall, we are confident that these first steps will lead to a fruitful research field, success in which will rely on the combined efforts of IR and HCI researchers.

This paper opens a novel area for Information Retrieval in that a user is interacting not only with a human partner, but with a search engine that is also interacting with the same partner, algorithmically taking into account that partner's actions to fulfill a shared information need. These streams of information (computer retrieval plus partner search activity) are combined algorithmically in real time; they alter and influence each other. How these intertwined streams are presented to each partner, and what effect that has on retrieval effectiveness of the system, is a challenging and fruitful open question, one with considerable practical and research value.

8. ACKNOWLEDGEMENTS

We would like to thank Lisa Anthony for numerous contributions to both design and system code, and Sagar Gattepally and John Adcock for frameworks and improvements to the code.

9. REFERENCES

[1] J. Adcock, M. D. Cooper, A. Girgensohn, and L. Wilcox. Interactive video search using multilevel indexing. In *CIVR 2005*, pages 205–214, 2005.

[2] J. A. Aslam and M. Montague. Models for metasearch. In W. B. Croft, D. J. Harper, D. H. Kraft, and J. Zobel, editors, *Proc. SIGIR 2001*, pages 276–284. ACM Press, September 2001.

[3] J. A. Aslam, V. Pavlu, and R. Savell. A unified model for metasearch and the efficient evaluation of retrieval systems via the hedge algorithm. In *Proc. SIGIR 2003*, pages 393–394, July 2003.

[4] R. Baeza-Yates and J. A. Pino. A first step to formally evaluate collaborative work. In *GROUP '97: Proc. ACM SIGGROUP Conference on Supporting Group Work*, pages 56–60, New York, NY, USA, 1997.

[5] N. J. Belkin, P. Kantor, E. A. Fox, and J. A. Shaw. Combining the evidence of multiple query representations for information retrieval. *Information Processing and Management*, 31(3):431–448, 1995.

[6] A. Girgensohn, J. Adcock, M. D. Cooper, and L. Wilcox. A synergistic approach to efficient interactive video retrieval. In *INTERACT*, pages 781–794, 2005.

[7] G. Golovchinsky. What the query told the link: the integration of hypertext and information retrieval. In *HYPERTEXT 1997*, pages 67–74, New York, NY, USA, 1997. ACM.

[8] A. Hauptmann, W.-H. Lin, R. Yan, J. Yang, and M.-Y. Chen. Extreme video retrieval: Joint maximization of human and computer performance. In *Proc. ACM Multimedia 2006*, pages 385–394, Santa Barbara, CA, 2006.

[9] M. R. Morris and E. Horvitz. Searchtogether: an interface for collaborative web search. In *Proceedings of UIST*, pages 3–12, 2007.

[10] J. Pickens and G. Golovchinsky. Collaborative exploratory search. In *Proc. 2007 HCIR Workshop*, pages 21–22, October 2007.

[11] C. Shah, D. Kelly, and X. Fu. Making mind and machine meet: a study of combining cognitive and algorithmic relevance feedback. In *Proc. SIGIR 2007*, pages 877–878, New York, NY, USA, 2007. ACM.

[12] J. A. Shaw and E. A. Fox. Combination of multiple searches. In *Text (REtrieval) Conference*, pages 105–108, 1994.

[13] A. F. Smeaton, H. Lee, C. Foley, S. McGivney, and C. Gurrin. Físchlár-diamondtouch: Collaborative video searching on a table. In *Multimedia Content Analysis, Management, and Retrieval*, San Jose, CA, January 15-19 2006.

[14] B. Smyth, E. Balfe, O. Boydell, K. Bradley, P. Briggs, M. Coyle, and J. Freyne. A live-user evaluation of collaborative web search. In *Proc. IJCAI 2005)*, pages 1419–1424, Ediburgh, Scotland, 2005.

[15] M. B. Twidale, D. M. Nichols, and C. D. Paice. Browsing is a collaborative process. *Information Processing and Management*, 33(6):761–783, 1997.