Contextual Support for Collaborative Information Retrieval

Shuguang Han[1], Daqing He[1], Zhen Yue[2] and Jiepu Jiang[3]
[1] University of Pittsburgh, 135 N Bellefield Ave., Pittsburgh, PA, USA
[2] Yahoo! Labs, 701 First Ave., Sunnyvale, CA, USA
[3] Center for Intelligent Information Retrieval, University of Massachusetts Amherst
{shh69,dah44}@pitt.edu, zhenyue@yahoo-inc.com, jpjiang@cs.umass.edu

ABSTRACT

Recent research shows that Collaborative Information Retrieval (CIR), in which two or more users collaborate on the same search task, has become increasingly popular. The presence of both search and collaboration behaviors makes CIR a complex search format, which further drives a critical need to understand CIR's search context. The contextual support for CIR should consider search contexts derived from both team members' search histories (including users' own search histories and partners' search histories) and their explicit collaboration (e.g., chatting). As it stands, existing studies on contextual search support only focus on Individual Information Retrieval (IIR) and only utilize individuals' own search histories. In this paper, we examine the unique search contexts (e.g., partners' search histories and team collaboration histories) in CIR. Based on a user study data collection with 54 participants, we find that compared to the use of individuals' own search histories, CIR contextual support is more effective when utilizing partners' search histories and teams' collaboration behaviors. More interestingly, though the explicit communication information (i.e., chat content) often involves massive noisy information, involving such noise does not affect the ranking of relevant documents since it also does not appear in relevant documents.

Keywords

Collaborative information retrieval, context-sensitive information retrieval, relevance feedback

1. INTRODUCTION

Despite the fact that information retrieval is often viewed as an individual behavior, recent studies [14, 15] report that collaborative information retrieval (CIR), in which two or more users collaborate and coordinate on the same search task, has increased from 0.9% in 2006 to 11% in 2012. A typical example of CIR is: two friends communicate over Facebook and search for nearby restaurants to go for dinner. In this task scenario, they may search and communicate in multiple rounds until reaching a final agreement. Therefore, unlike Individual Information Retrieval (IIR), CIR not only includes search activities, but also involves a significant amount of team collaboration such as sharing visited information and communicating with partners (i.e., chat) [19, 28]. In addition to these differences, researchers observe that users in CIR often develop new search tactics and strategies, compared to that in IIR [28].

Compared with IIR, CIR tasks are usually more complex and require more user exploration [15, 28]. This makes the inference of users' search contexts and the contextual support of CIR more difficult, and it is unclear whether or not the previously proposed IIR-based contextual support approach [23] is still useful. To handle these challenges, we investigate the contextual support for CIR in this paper.

The most common approach for inferring users' search context is to utilize users' search histories, which often include query history and click-through history [23]. Both have been demonstrated to be effective in improving search performance in IIR [1, 3, 23], whereas the contextual support for CIR is not well studied [17]. A straightforward CIR contextual support method is to ignore user collaboration and consider each team member as an individual user. In this way, the contextual support algorithms developed in IIR can be directly employed. However, the current literature has not yet provided a clear conclusion for the validity of this approach, and this is the focus of our first research question. In addition, considering that this approach does not take advantage of the unique contextual information only available in CIR, our further research questions examine how the new contextual information will support CIR search.

CIR involves more than one user so that the search histories can either come from the given user him/herself or from his/her partners. Although existing studies already explore the ways of incorporating search histories from other users [6, 26], the involved users are not designated as the given users' search partners. This is critically different from CIR because users in the above two studies have no direct or indirect collaborations, whereas collaboration has its unique importance in CIR. Consequently, an effective contextual support for CIR should be studied further, with particular focus on analyzing the utility of users' own and partners' search histories. This is a relatively new research topic; existing literature has examined it as algorithm-mediated CIR [17, 22, 24], where search results are customized based on users' search roles. The roles are either predefined or mined from their search histories. These studies have demonstrated the overall effectiveness of algorithm-mediated CIR, but it is

CHIIR '16, March 13-17, 2016, Carrboro, NC, USA

DOI: http://dx.doi.org/10.1145/2854946.2854963

unclear whether the boost of search performance comes from utilizing users' own or partners' search histories or if it comes from users' role-based result customization. Therefore, we will examine the approach that purely utilizes users' own and partners' search histories and not consider user roles.

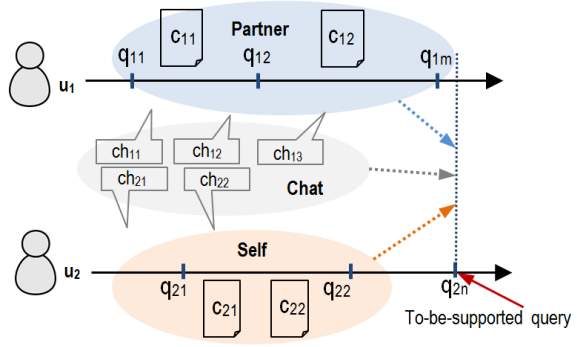

Figure 1: Context-sensitive retrieval framework for IIR and CIR. IIR only employs users' own search history whereas CIR can utilize self-history, partner-history and also chat messages.

CIR also contains considerable user collaboration behaviors (i.e., chat). Chat was found to have significant impact on the formation of CIR search strategies and their query reformulations [29]. However, modeling search context with chat information remains an open research topic with numerous challenges. The first challenge is that chats usually contain a massive amount of noise [27], some of which is not even task-related. It is also unclear that what effective methods can be applied to remove the noise and whether such noise will exert significant negative impacts on supporting usersâĂŹ search queries. The second challenge is that users may chat for different purposes - some may discuss a specific subtopic for the search task, some may coordinate different sub-tasks, while still others may be chatting for social purposes. It is unclear which type(s) of chat could be useful for contextual supports.

In this paper, we are interested in understanding the effectiveness of two unique types of contextual information (i.e., a partner's search history and a team's chat history) in CIR. Following the context-sensitive retrieval algorithm [23] developed in IIR, we refer to the contextual support for CIR as Context-sensitive Collaborative Information Retrieval (CCIR). Figure 1 illustrates CCIR in a collaborative search with two users u_1 and u_2. The search contexts for supporting the new query q_{2n} of u_2 can be inferred from the historical search queries - q_{11}, q_{12} ... for u_1, and q_{21}, q_{22} ... for u_2; the clicked documents - c_{11}, c_{12} ... for u_1, and c_{21}, c_{22} ... for u_2; and the chat between u_1 and u_2 - c_{11}, c_{12} ... for u_1, and c_{21}, c_{22} ... for u_2. In IIR, context-sensitive IR will aim to provide a better document ranking using u_2's search histories, whereas CCIR can employ three types of histories to infer users' search contexts: u_2's own search history, u_2's partner search history (i.e., u_1's search history) and their chat histories.

However, our research goal in this paper does not aim to propose an optimal contextual-based CIR ranking model; instead, we are more concerned with exploring the utilities of different types of search contexts. Thus, this paper represents an exploratory work for demonstrating the necessity of utilizing the contextual factors in CIR. Consequently, we are interested in the following three research questions (RQs):

- RQ1: Can the context-sensitive retrieval algorithms developed in IIR [23] be directly applied in CIR without handling complex CIR user interactions?
- RQ2: Can partners' search histories in CIR be employed for better modeling of users' search contexts and to further improve the search performance?
- RQ3: Can team members' chat histories in CIR be employed for modeling users' search contexts and further improve the search performance?

2. RELATED WORK

The related work of our paper occupies three areas. The first one involves studying users' CIR search behaviors. Many CIR studies [20, 21, 28] observe that team members do collaborate very frequently, and the collaboration does indeed bring in diverse expertise [28]. However, users' collaboration also intervenes with and complicates an individual user's normal search process. Therefore, users in CIR also exhibit several new search patterns. For example, through the exploration of Kuhlthau's ISP model [13] for IIR, Hyldegard [10] suggests that the ISP model in CIR should incorporate social and contextual factors. Similarly, through mapping Kuhlthau's ISP model to CIR, Shah and Gonzalez-Ibanez [20] declare that social elements are missing. Yue et al. [28] compare the search tactic/strategy differences between IIR and CIR, to find that CIR involves more sense-making related search tactics. These above-mentioned studies all indicate the high complexity of CIR search processes.

The second related area is concerned with search supports in CIR. Pickens et al. [17] mention two types of supports in a CIR system: user-mediated support and system-mediated support. The user-mediated search support provides simple and user-transparent functions, often at the interface level, to assist users in completing their search tasks [19]. Most of the existing CIR-support systems, including SearchTogether [16], Coagmento [18], and CollabSearch [28], fall into this category. System-mediated support focuses on merging relevant documents by integrating inputs from partners. Then, the relevant documents are redistributed to different users based on certain strategies. CIR systems, such as Cerchiamo [17] and Querium [5], belong to this category. Current studies on system-mediated CIR support only explore the strategy of regrouping search results based on user roles [17, 22, 24], which are either manually predefined [17, 22] or automatically learned from users' behavior logs [24]. However, these studies restrict user roles into predefined categories (e.g., Prospector or Miner, and Gatherer or Surveyor [24]), while the collaboration styles of real users may vary significantly. Thus, to provide a more flexible CIR support, either a robust role-mining algorithm or a new result differentiation method is needed.

The last related area involves search context modeling in CIR. Previous researchers have identified that users' search histories can provide effective feedback to improve retrieval performance [3, 11, 23]. However, this effectiveness was only tested in IIR. There are indeed several studies that take into account the search histories from other users. For example, White et al. [26] found that it is beneficial to include the search history information from other users who performed

similar search tasks to support the given users' queries. Similar ideas have also been explored as social navigation [2, 6, 9, 8], where social cues (e.g. the highlighted content) generated from other users are utilized to assist current users' information-seeking processes. However, unlike the users in CIR, these users are not explicit search partners. Therefore, the effectiveness of utilizing other users' search histories needs to be reexamined. In addition, because chat information is unique contextual information in CIR, its effectiveness as a CIR search context should also be properly studied. These are also our research focuses in this paper.

3. OBTAINING EXPERIMENTAL DATASET

3.1 User Study Design

To examine the above-mentioned research questions, we need to build a collaborative web search data collection. In this paper, the data collection was obtained through a laboratory-controlled user study. We do acknowledge that lab study may introduce certain artificial factors since search tasks are not originated from the users themselves but are instead preassigned. However, this approach with simulated tasks is an effective way to evaluate interactive information retrieval systems [4] and is commonly used in CIR studies too [22, 19, 28].

Our study employed CollabSearch[1], a collaborative web search system developed by Yue et al. [28]. The two exploratory web search tasks were also borrowed from [28] - one was an academic task (T1) that asked participants to collect information about a social networking service and the other was a leisure task (T2) that asked participants to collect information for planning a trip. We chose these two tasks because they are representative collaborative web search tasks that are commonly adopted in many other CIR studies [20, 22, 24, 28].

To simulate a real collaborative web search scenario, we recruited participants as pairs so that two of them could work together on the same task. The pairs were asked to have past collaboration experience (e.g., past course collaboration experience) before attending our studies. Following the majority of CIR studies [17, 19, 28], we only worked on the CIR teams with two persons. Previous literature also showed that pairs are the most common team format involved in collaborative web searches [15].

Our study also included an IIR scenario for comparison. We took a between-subject design for this comparison so that the participants for IIR were recruited separately instead of asking the collaborative teams to continue working on IIR tasks. The first motivation for this design is to use the same tasks for both IIR and CIR to eliminate task effect. Here, we cannot adopt within-subject design because CIR participants cannot further work on IIR since they already completed the tasks in the CIR condition. Secondly, each team usually spent about two hours on CIR tasks. They would be exhausted and generate low quality results if we followed the within-subject design and continued asking them to perform IIR search tasks.

We do acknowledge that the current design of the study (i.e., using only two search tasks and a relatively small number of participants) may cause the conclusion of insufficient generalizablility. It is still useful for this exploratory work to point out insights for further research directions. More importantly, this current design helps to significantly reduce the complexity and time span of the user study, which enables researchers to avoid the limitation that users in a larger and longer study may see diverse search results at different time periods since CollabSearch depends on Google to return real-time search results that can change over time.

[1]http://crystal.exp.sis.pitt.edu:8080/CollaborativeSearch/

3.2 User Study Procedure

After being introduced to the study and completing an entry questionnaire to establish their search experience background, the participants then worked on a training task for 15 minutes to get familiar with the system. Then, each team worked for 30 minutes on each of the two assigned tasks. The task order was rotated and preassigned to each team to avoid learning and fatigue effects. During the search for each task, the participants were instructed to save as many relevant web pages as possible. At the end of each task, the participants were asked to rate the relevance (at the task level) of each saved web page on a 5-point Likert scale, with 1 denoting non-relevant and 5 being highly relevant. These scores will be used for building the ground-truth.

3.3 User Study Data

Our study recruited 36 collaborative participants (i.e., 18 pairs) and 18 individuals. These 54 participants are comprised of 26 females and 28 males. Twenty-four participants are undergraduate students and 30 are graduates, all from University of Pittsburgh or Carnegie Mellon University. They have strong computer experience and conduct web search on a daily basis. In response to a question regarding their search experience, where 1 denotes the least experience and 7 the most experience, all of the participants rated their experience from 4 to 7.

In total, the participants issued 970 search queries (537 for T1 and 433 for T2), clicked 1,384 web pages and saved 909 pages. We downloaded the HTML sources of the top 100 Google search results for all 970 queries to build a data collection. In total, the collection includes 54,364 unique web pages. Additionally, we applied a state-of-the-art content extraction algorithm [12] to extract the full text of each web page and to remove the advertisements and copyright information for each web page.

4. CONTEXT-SENSITIVE CIR

4.1 Experiment Setup

To answer the research questions raised in the Introduction, we design the following simulation experiment. After obtaining users' search logs from CollabSearch system, we extract each of an individual's search queries q, its issuing time t and the user u in a temporal order. The goal of the simulation experiment is to provide a better document ranking for q by combining the content of q and different types of contextual information inferred from users' search and chat histories (before time t). In this manner, we simulate the contextual support for all user-issued search queries in a sequential order.

As shown in the CCIR framework (Figure 1),it is possible to utilize different types of contextual information in CIR. Depending on the sources and types of users' histories that are used to infer contextual information, we can obtain six types of search contexts, as shown in Figure 2. We

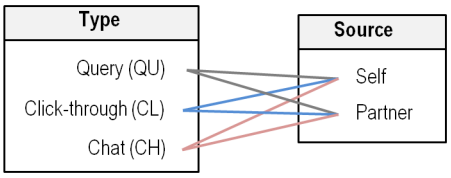

Figure 2: Different types of contextual information available in CIR

denote the query, click-through, and chat histories as H_{QU}, H_{CL} and H_{CH}, respectively. They can be further differentiated by their sources - H_{XS} for self and H_{XP} for partner, where X∈{QU,CL,CH}. Since chat usually requires extensive involvement from both users, we do not further differentiate the users' own and partners' chats in our experiments. Therefore, in total, there are five types of contextual information. Following the context-sensitive retrieval algorithm in [23], users' behavior histories are converted to contextual language models and then used for search result re-ranking.

4.2 Estimating Contextual Language Models

After obtaining users' search histories, the next step is to estimate their corresponding contextual language models (θ_{HX}) [11, 23]. In this paper, we estimate a unigram language model for each type of contextual information, which can be formalized as Formula (1) and (2). H_X can be any type of user history (i.e., H_{QU}, H_{CL} and H_{CH}). Note that H_X can either come from the given user or his/her partners. $c(w, X_i)$ denotes the count of word w in the search history element X_i, $|X_i|$ is the word count of X_i and k is the total number of user histories (i.e., the number of historical queries, chat messages or clicked documents) of type X.

$$p(w|H_X) = \frac{1}{K}\sum_{i=1}^{K} p(w|X_i) \quad (1)$$

$$p(w|X_i) = \frac{c(w, X_i)}{|X_i|} \quad (2)$$

4.3 Document Re-ranking

We apply the following procedure to re-rank relevant documents for each to-be-supported query. For each candidate document, we estimate its document language model using Dirchilet smoothing [31], where we set the smoothing parameter $\mu = 100$. The similarity between each candidate document and the contextual model is measured by the KL divergence between their estimated language models [30]. The matching between a candidate document and the query is determined by Google rank position of the given candidate document. This is because our experimental system uses Google results as the default.

Instead of using linear interpolation proposed by Shen et al. [23], we employ LambaMART in RankLib[2] to build a pairwise learning-to-rank approach for combining different features. To be specific, for each query-document pair, we employ two features: (1) Google rank position (G) of the candidate document for the given query, and (2) KL divergence between a contextual language model (i.e., θ_{HX}) and the candidate document language model. The parameters of LambaMART are set as default in RankLib, except the number of leaves is set to 10. We use 10% of our training samples as validation data, and the features are normalized using the sum of all feature values to make different features in a same scale. Our parameter settings are set as the same to a previous study [9], which tries to handle a similar re-ranking problem.

[2]http://sourceforge.net/p/lemur/wiki/RankLib/

4.4 Ground-truth and Evaluation

4.4.1 Building Ground-truth

To evaluate the effectiveness of the above-mentioned contextual information, we need the ground-truth data. The ground-truth is built at the task level, in which we aggregate the relevant documents saved by all participants. Furthermore, we assume that the goal for each user is to find the new relevant documents for the whole team (not including the already-explored documents from all team members). Therefore, the ground-truth removed all of the already-saved documents from both team members. Figure 3 illustrates the ground-truth building procedure. To support the search query q of user u at time t, we aggregate the saved documents from all participants except u and v. We also remove the already-saved documents from u and v (u's search partner) up to time t.

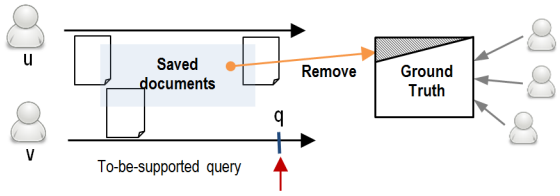

Figure 3: CCIR ground-truth building procedure

After obtaining the ground-truth pool, we compute the relevance of each document in the pool. Note that the participants are asked to rate the relevance of their saved documents on a 5-point Likert scale at the end of each task (see Section 3.1). These relevance scores are used to generate the final aggregated relevance. The simplest way to aggregate multiple scores from different users is to compute the average of these scores. However, the average can be biased when only a small number of participants rate the web page. A Bayesian smoothing approach [9] is adopted to remove this bias. Specifically, the smoothed average relevance score $\hat{r'}(d_i)$ for document d_i is computed based on the interpolation between its average score $\hat{r}(d_i)$ and the group average, as shown in Formula (3). C is the average rating for all documents, and v denotes the number of participants who saved the document d_i.

$$\hat{r'}(d_i) = \frac{\hat{r}(d_i)v + C}{v + 1} \quad (3)$$

4.4.2 Evaluation Setup

In the evaluation, we randomly split the data collection into training and testing datasets using a five-fold cross-validation. The random division is repeated 10 times to remove biases. Our model parameters are learned from the training dataset, and then these parameters are applied to the learning-to-rank model in the testing dataset for evaluation. In our evaluation, the effectiveness of the learned

models based on different contextual information was measured by MAP (Note that MAP only cares about the binary relevance of a document, i.e., relevant and non-reelvant. In our study, a document is defined as a relevant document if the score computed using Equation 3 is bigger than 3.0) and nDCG@N (N was set to be 1, 2,..., 20 in this paper). The reported MAP and nDCG values were the averages across five folds and over all 10 runs.

5. RESULT ANALYSIS

5.1 Experiment Overview

Our result analysis includes extensive statistical tests using non-parametric Wilcoxon signed-rank test (with Bonferroni correction) since the data is not normally distributed. Because we will consider two tasks and two evaluation metrics and compare multiple systems in result analysis at the same time, we do not provide Z-scores and P-values in this paper since there are too many values to report. Particularly, there are 20 different cutoffs for nDCG evalution metric. Therefore. we set the statistical significance at 0.05 level and only report significant results. We design the following three experiments to answer our research questions mentioned in the Introduction.

- Although query and click-through search histories have been proved to be useful in supporting complex IIR tasks [9, 11, 25], they have not been extensively studied in the context of CIR. Therefore, our first experiment explores the effectiveness of applying IIR-based contextual support in CIR. This experiment targets RQ1. The details are provided in Section 5.2.
- The second experiment addresses RQ2, which is to better understand the utility of applying partners' search histories for CIR contextual support. The details of this experiment are presented in Section 5.3.
- The third experiment answers RQ3, and we focus on studying the effectiveness of utilizing chat information to support CIR search processes. Since not all chat messages contain useful task-related information, we further categorize chat information into different groups and compare each group's effectiveness. The details of this experiment are presented in Section 5.4.

5.2 Utilizing Users' Own Search Histories

5.2.1 *Overall Results*

The goal of our first experiment is to understand whether the IIR-based contextual support still works in CIR. We develop two search contexts based on users' own query histories and click-through histories, as shown below.

- H_QUS: query history for user u;
- H_CLS: click-through history for the given user u.

These two search contexts are then combined with the baseline Google ranking feature (G) to produce two contextual models (i.e., the search result re-ranking models), which are noted as G+H_QUS and G+H_CLS. The MAP and nDCG evaluations on these two models are provided in Figure 4, where we find that both models perform significantly better than the Google baseline on MAP. For nDCG, we also observe statistically significant results at almost all cutoffs (we do not provide Z-scores here since there are too many values to report), which reveals that the context-sensitive retrieval model developed in IIR [23] also works well in CIR tasks. Although users' collaborations may potentially disrupt their individual search processes, users' search histories are still useful contextual information and should be properly considered for better contextual support.

Second, the statistical test shows that G+H_QUS is significantly better than G+H_CLS. That is, the search context inferred from query history is more effective than that from click-through history. This result is different from several previous studies about IIR [9, 11, 23], in which click-through is found to be more effective than query history. We think that there are two possible reasons. The first one is that the tasks involved in our study are different from the tasks included in previous IIR studies; our studyâĂŹs tasks are more complex and contain diverse sub-tasks. If task difference is the reason, the query history will also be more effective in IIR. The other potential reason is that users' normal search processes might be interrupted by the involvement of collaborative activities. This may drive people to issue different types of search queries and click different kinds of web pages. If this is true, the performance difference may be more related to the synergy effect between search and collaboration. We will examine them in the next section.

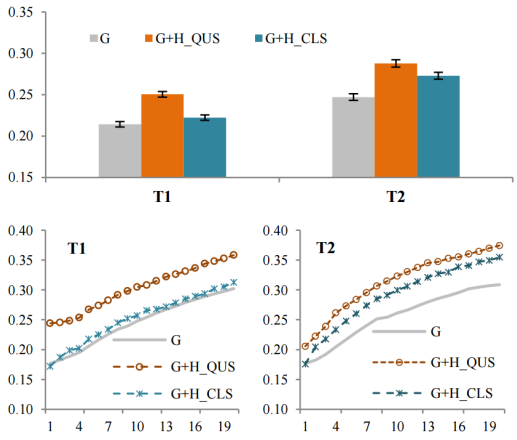

Figure 4: MAP (top) and nDCG (bottom) evaluation on the effectiveness of different CIR search contexts. X: task/cutoff. Y: MAP/nDCG values.

5.2.2 *Comparing with IIR*

Our user study also includes an IIR search condition, where the participants search individually for the same tasks as those in CIR. Using such data, we can compare the effectiveness of IIR-based search contexts with the CIR search contexts. The support for IIR search queries follows the same procedure used for CIR, except that we cannot access partners' search histories. Like in CIR, we also consider three ranking models, including the Google baseline (G) and two contextual models: G+H_QUS and G+H_CLS.

According to the MAP and nDCG evaluations for IIR in Figure 5, we find that the search history-based contextual information also helps in IIR - G+H_QUS (H_CLS) has significant performance boosts over the Google baseline (G). However, the improvement greatly depends on the

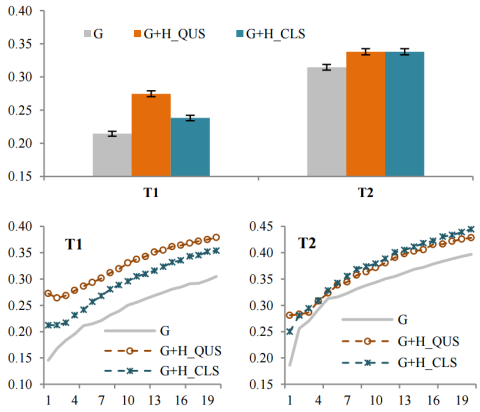

Figure 5: MAP (top) and nDCG (bottom) evaluation on the effectiveness of different IIR search contexts. X: task/cutoff. Y: MAP/nDCG values.

task. Wilcoxon signed-rank tests on T1 show that the query history significantly boosts performance more than the click-through history on both MAP and nDCG. This is consistent with the CIR results for T1. However, the results for T2 show that utilizing the click-through history is equivalent or even better than query history, particularly for nDCG at high cutoffs (bigger than 10). This is inconsistent with the results for CIR shown in Figure 4. One possible explanation for this is that the synergy between search and collaboration in CIR affects users' query and click behaviors, and further causes inconsistency. Since the inconsistency only happens in T2, if our hypothesis is correct, there might be a stronger synergy effect in T2 than in T1.

Table 1: Mean (S.D.) for different measures in CIR

	T1	T2	Sig.
#chat messages	21.56 (20.98)	38.86 (27.24)	$p < 0.001$
#queries	12.97 (6.98)	9.25 (5.30)	$p < 0.001$
#click-through	24.00 (13.23)	14.44 (7.25)	$p < 0.001$

To test this hypothesis, we compute additional measures (including number of queries, number of click-through documents and number of chat messages) for each task and compare their differences. Table I shows that within the same amount of time (30 minutes for each task), users in T1 issued more queries and clicked more documents, but composed fewer chat messages than the users in T2. Thus, T1 can be viewed as a search-intensive task where people spend more time searching and reading search results, whereas T2 is a collaboration-intensive task, since users spend more time on communication and collaboration. Consequently, it makes sense that CIR for T1 is more similar to the patterns observed in IIR, because users spend less time on collaboration and behave more like they are searching individually. Linking back to the hypothesis in the last paragraph of Section 5.2.1, we find that both task nature and synergy effect between collaboration and search can affect the search performance. The synergy effect may have a greater impact when there are more collaborations.

5.3 Involving Partners' Search Histories

To answer RQ2, we develop two search contexts extracted from partners' search histories, including:

- H_QU: query histories for u and u's search partner v;
- H_CL: click-through histories from both u and v.

We also include the search contexts from the above sections for comparison. In total, five ranking models are used, including the Google baseline (G) and four contextual models (G+H_QUS, G+H_CLS, G+H_QU and G+H_CL). Partners' search histories are not used separately, but instead are integrated with users' own histories. This is because different team members may have different search traces. Solely applying partners' search histories could distort users' own search traces, which is a risky approach to apply in a real search system. In contrast, aggregating search histories from all team members can highlight the relevant information without losing content diversity.

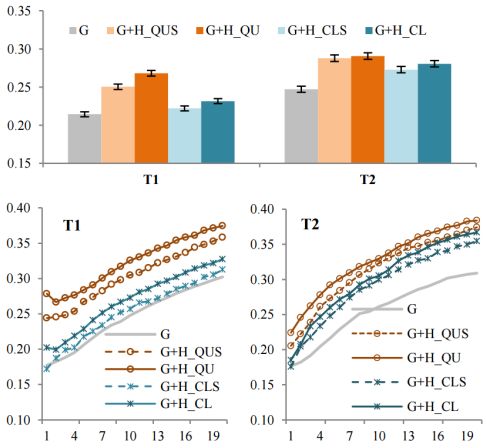

Figure 6: MAP (top) and nDCG (bottom) evaluation on the effectiveness of different search contexts in IIR. X: task/cutoff. Y: MAP/nDCG values.

The MAP and nDCG evaluations on these five ranking models are provided in Figure 6. Compared with users' own search histories, aggregating partners' search histories provides a better contextual support. This is based on the significant MAP and nDCG improvements on G+H_X over G+H_XS, where X∈{QU,CL}. The nDCG improvement may come from the fact that utilizing both users' own and their partners' histories can help identify the most relevant information. However, combining information from both users may potentially reduce information diversity. This means that the MAP of our exploratory web search tasks may be reduced because they usually cover multiple relevant or partially relevant subtopics. However, the significant MAP increases instead of decreases in Figure 6 further eliminates our concerns.

5.4 Utilizing Chat Histories

5.4.1 *Overall Results*

Another important distinction between CIR and IIR is that CIR involves explicit communication (i.e., chat) among users. To better understand the utility of applying chat content in supporting CIR queries, we conduct the following experiments. Firstly, we compare four different ranking models, including the Google baseline (G) and three contextual

models (G+H_QU, G+H_CL and G+H_CH). H_CH refers to the search context inferred from chat messages. H_QU and H_CL are two additional baselines because of their superior performances in the above experiments. Stemming and stop-word removal are adopted when utilizing H_CH.

According to Yue et al. [27], around 30% chat messages are not directly related to the task content in CIR. We initially anticipated a poor result by applying the chat-based contextual support. However, as shown in Figure 7, G+H_CH surprisingly outperforms G+H_CL on MAP and nDCG in both tasks. Furthermore, it even significantly outperforms G+H_QU on T2 for MAP and nDCG. This implies that chat-based search context is highly effective in CIR support. In addition, we also observe the joint influence between task and chat-based search context. Chat-based search context achieves the best performance in T2, whereas it is only the second-best in T1. We think that the collaboration-intensive task (T2) may have more chat messages on discussing task content, and thus can provide better support than search intensive task (T1). To further understand the reasons, our next experiment attempts to separate different types of chat messages and compares their impact on search effectiveness.

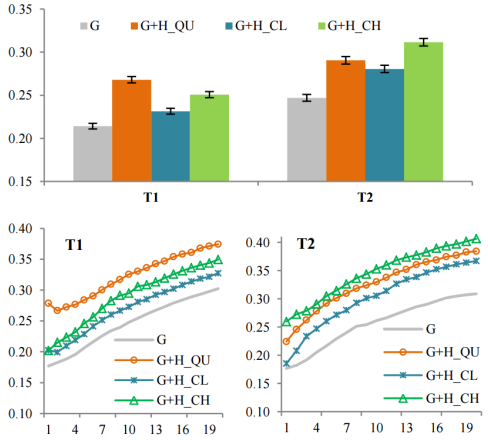

Figure 7: MAP (top) and nDCG (bottom) evaluation on the effectiveness of different search contexts in CIR. X: task/cutoff. Y: MAP/nDCG values

5.4.2 *Chat Functionality Analysis*

We categorize chat into four groups based on its functionality, which includes task content (TT), task coordination (TC), task social (TS) and non-task (NT). This schema (Table II) is directly adopted from a previous study on CIR [7] with only slight modifications. Task social chat messages usually attempt to provide social support between team members. Typical examples are greetings, encouragement between team members and opinions on the obtained information (e.g., *"well, hello there", " yeah! we are going to Helsinki!", "everything looks great so far!"*). Labor division and task progress checking are two major types of task coordination chat messages (e.g., *"you do stats and I'll do impacts on students and professionals", "have you done impact yet?"*). Chat messages about task requirements and assessments of the obtained information belong to task content (e.g., *"ok so outdoor activities will be hard", "in December they set up tons of markets and stuff in the streets"*). And, there are also several non-task chat messages that are not related to the task itself (e.g., *"Can we eat after this?", "I wish there was a notification every time we saved a page"*).

Table 2: Categorization schema for chat

Category	Description
Task social	Chat messages concerning group effort or attitude to the obtained information
Task coordination	Chat messages regarding the coordination of the search task, including division of labor and checking task status
Task content	Chat messages related to the content of the search task, including task requirement and information resource assessment
Non-task	Chat messages that are not related to the search task or the user study

To ensure data quality, we manually labeled the category for each chat message. Two coders went through all chat messages and manually assigned a category for each. The first round of coding was performed by each coder independently, with an agreement of 86.1% for T1 and 83.3% for T2. Then, a second round of coding was performed to resolve the disagreements. Based on the labeled data, we computed the percentages of different chat types. As shown in Table 3, T2 had a relatively large number of chat messages that discuss task content, whereas T1 has more chat messages focusing on task coordination. We also observe that there is a large number of chat messages that are irrelevant to the current task (22% for T1 and 13% for T2).

Table 3: Percentages (S.D.) of chat messages in different categories.

	T1	T2	Sig.
Task social	16.46 (16.24)	12.93 (9.16)	p=0.293
Task content	25.43 (21.66)	58.77 (22.00)	p<0.001
Task coordination	36.51 (24.27)	15.42 (8.49)	p<0.001
Non-task related	21.60 (24.48)	12.88 (17.10)	p=0.060

We think different chat types may have different impact, which motivates us to compare the difference among different chat types. We consider the following five types of chat messages, each inferred from one chat type. Combining each of these search contexts with the Google ranking feature, we propose five contextual models, represented as G+H_CH, G+H_TT, G+H_TC, G+H_TS and G+H_NT, where:

- H_CH: using all chat messages;
- H_TT: using the chat messages of task content;
- H_TC: using the chat messages of task coordination;
- H_TS: using the chat messages of task social;
- H_NT: using the chat messages of non-task.

MAP and nDCG evaluations of the five contextual models, along with the Google baseline (G), are provided in Figure 8. We have several interesting observations based on the results. Firstly, utilizing any type of chat message, even the non-task chat, can help improve search performance over the Google baseline (for both MAP and nDCG). Secondly, the involvement of all chat messages achieves the best performance. This implies that different types of chat messages all contain useful information. Wilcoxon signed-rank tests show that, at most cutoffs, MAP and nDCG of G+H_CH have significant performance improvement over the use of other types of chat messages. However, different types of

chat messages may help with different aspects. For example, task coordination (TC) and task content (TT) are the two most effective search contexts. TC has better search performance in T1, whereas TT performs better in T2. We think this is related to the amount of messages for each type of chat in each task (see Table III) - TT occurs the most in T2, whereas TC occurs the most in T1. Note that the contextual feature used in our re-ranking model is already normalized by the number of chat messages (see Formula 2), a better search performance of a contextual model means that the amount of useful information contains in one unit of chat message is richer than that in other models.

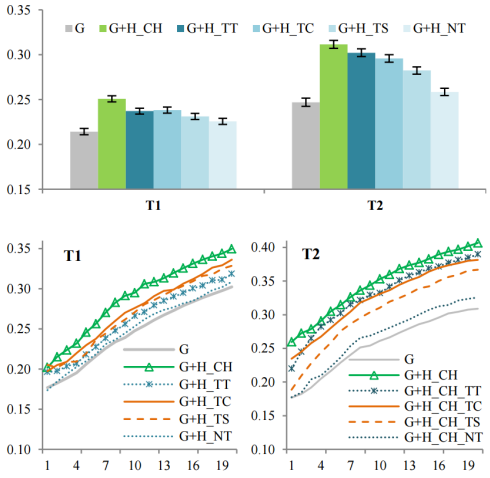

Figure 8: MAP (top) and nDCG (bottom) evaluation on the effectiveness of different search contexts in CIR. X: task/cutoff. Y: MAP/nDCG values.

Since non-task (NT) chat contains a large amount of noise information, unsurprisingly, G+H_NT is the least effective chat-based contextual model. Despite this, we still observe a significant MAP and nDCG (at most of the cutoffs) increase when utilizing the NT chat, which may due to the following two reasons. First, one chat message in our study refers to all content typed into the chat box before a user hits the submit button. Therefore, one chat message may contain more than one type of chat information. However, in our manually-labeled dataset, one chat message is forced to be put into one of the four chat types. It is possible that the effectiveness of the NT chat comes from these chat messages that contain multiple chat types. Second, the noise information in the NT chat messages probably does not hurt the search result ranking either. We hypothesize that the noisy information from NT chat messages (e.g., *lol*, *hah*) is so different that it has little chance to be included in both relevant and non-relevant documents. Therefore, the noise has no impact on the result re-ranking. We will test this in the next section.

5.4.3 Analyzing Non-task Chat Messages

This section tries to answer the following two questions: (1) whether the involvement or removal of non-task chat messages will influence the effectiveness of chat-based contextual support; and (2) why the noise information involved in non-task chat messages does not hurt search performance.

In the first experiment, we introduce an additional search context, which includes all chat messages except NT chat messages. We name it H_NT_R. Combining this search context with the Google ranking feature (G), we can obtain a new contextual model, G+H_NT_R. Then, we compare its performance with two baselines - G and G+H_CH. As shown in Figure 9, the MAP and nDCG evaluations of G+H_NT_R are almost identical to that of G+H_CH except for a slightly better performance over G+H_CH on nDCG (though no statistical significance is detected). This implies that despite providing useful contextual information (see Figure 8), NT chat is already covered by the chat messages of other types.

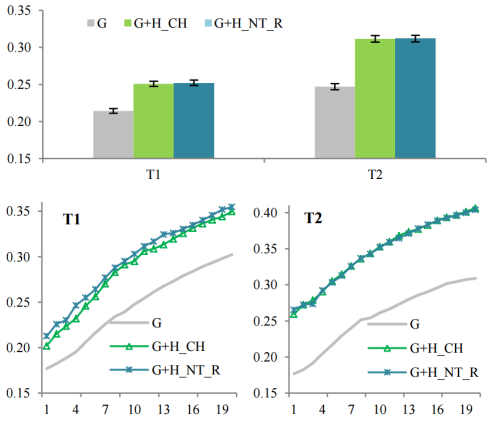

Figure 9: MAP (top) and nDCG (bottom) evaluation on the effectiveness of different search contexts in CIR. X: task/cutoff. Y: MAP/nDCG values

Next, we examine the impact of noisy information in the NT chat messages. Specifically, we want to know whether non-chat information is equally likely to occur in relevant documents and in the whole data corpus, which can be seen as a representation of non-relevant documents (for each query, the relevant document is usually only a small proportion of the whole data corpus). If NT chat messages are equally likely to occur in relevant and non-relevant documents, we can think that the NT chat messages have no impact on retrieving relevant documents. We use the term occurrence probability to measure this likelihood. $p(t|R)$ is the term occurrence probability in the relevant document pool, and $p(t|C)$ is the term occurrence probability in the whole data corpus, where t is a single term, R denotes the relevant document pool, and C denotes the whole data corpus. The relevant documents are decided based on the ground-truth relevance, which is calculated using Formula (3). A document with a ground-truth relevance score of more than 3.0 is used to form R.

We compute both the term occurrence probability for NT chat messages and for task-related chat messages (TT, TS and TC). As shown in Table IV, the NT chat has almost the same term occurrence probability in the relevant document pool and in the whole data corpus. However, the chat term occurrence probability of task-related chats in the relevant document pool is almost 1.5 times of the whole data corpus. This explains why including noise information from the NT chat does not influence the search performance.

5.5 Discussions and Insights

Our current evaluation procedure prefers the models that produce more relevant documents. However, not all CIR tasks are concerned on saving relevant documents. User

Table 4: Term occurrence probabilities for NT and non-NT chat (TT+TS+TC) for both two tasks.

Task		NT (10^{-3})	TT+TS+TC (10^{-3})
T1	p($t|C$)	0.520	1.936
T1	p($t|R$)	0.509	2.422
T2	p($t|C$)	0.333	2.120
T2	p($t|R$)	0.346	3.139

communication and collaboration may play more important roles in defining task success. So far, a robust evaluation metric to quantify all of these effects is still missing, and this remains an open issue in the CIR research community [19]. In CIR, existing evaluation metrics and the methods for ground-truth construction are mostly derived from IIR. Therefore, it is unsurprising to observe that the pure Google search performance for IIR is higher than CIR in T2 (see Figure 4 and Figure 5, for both the MAP and nDCG), which means that the ground-truth data may be more biased towards the individuals' search queries. Once the new CIR-based evaluation metrics are defined, we can further re-examine the effectiveness of different search contexts. This is one of our current limitations.

Another limitation is about the task generalizability. So far, this study only considers two types of tasks - an academic task and a leisure task. Although they are commonly adopted in CIR studies [20, 22, 24, 28], these tasks cannot cover the wide variety of the tasks employed in CIR. We plan to develop more search tasks and recruit more users to test the generalizability of our conclusions.

Our experimental results demonstrate that the contextual supports developed in IIR can be directly applied in CIR. However, their effectiveness highly depends on task types. In our study, the contextual support for CIR is consistent with that for IIR in search-intensive tasks, while differing in collaboration-intensive tasks. The result difference between two task types is due to the degree of user collaboration. Having more collaboration from team members may potentially change users' own search traces so that he/she may issue different types of search queries and click different types of web pages. This suggests that researchers should be more careful when applying IIR models in CIR, particularly when the search tasks require significant user collaborations.

We also find that the unique contextual information in CIR, including both partners' search histories and team members' chat histories, can significantly boost performance. However, the chat-based contextual support in CIR is also easily affected by task types. The utilities of chat-based search contexts may be related to the quality and quantity of chat messages that are available in CIR. For example, we found that chat-based contextual support can produce better search performance in chat-intensive tasks than in search-intensive tasks. When both search and chat histories are not enough, a proper leverage (e.g., combining) of both is needed.

6. CONCLUSIONS AND FUTURE WORK

Observing the increasing popularity of Collaborative Information Retrieval (CIR) in modern search systems [15], our study in this paper targets the support of CIR using contextual information. To be specific, we study: (1) whether the contextual supports developed in IIR can also be applied in CIR; and (2) if the unique information available in CIR, including chat information and partners' search histories, can be applied for better modeling of users' search contexts. These two questions are answered based on a list of our properly designed contextual support experiments, which are built on the data obtained through a user study with 54 participants working on two CIR search tasks.

Using the collected dataset, we set up a Context-sensitive Collaborative Information Retrieval (CCIR) framework to examine the effectiveness of different types of contextual information. Based on our experimental results, we find that the contextual support developed in IIR can be directly applied into CIR without significant adjustments. Specifically, the contextual search support for CIR is similar to IIR in search-intensive tasks, while it differs in collaboration-intensive tasks. We also find that the CIR contextual support can benefit from the unique information that is only available in CIR - i.e., partners' search behaviors and explicit collaboration among team members through chat. Particularly, although the chat often includes a massive amount of noisy information, such noise does not affect the document re-ranking because they do not occur in relevant documents.

We will explore several new topics in the future. First, chat is demonstrated to be an effective contextual resource, particularly in the chat-intensive search tasks. However, as we mentioned in Section 5.4.2, there is no clear definition for the unit of a chat message. For now, one chat message is defined as *all of the content a user types in the chat box before he/she hits the submit button*. Under this definition, however, it is very commonly observed that multiple messages focus on the same topic and/or subtopic or one message talks about multiple topics and subtopics. Segmenting chat messages into multiple semantic units may provide a better understanding of the chat-based search contexts.

Second, we want to develop an innovative contextual support algorithm to accommodate different search strategies for different users. CIR is usually associated with complex information needs, where both an individual user and a whole team usually develop various search strategies. For example, some teams may start their searches with general topics for two users, and then discuss and narrow down the search scopes. Other teams may prefer to divide the search tasks into several small subtopics and assign these subtopics for different team members at the very beginning. Identifying different search strategies and providing proper support for each one is an important future consideration.

Third, we have thus so far explored several search contexts and observed that most of them can provide positive contributions for supporting CIR search processes. However, we have not taken advantage of combining multiple search contexts into one unified model. This is another potential research topic for the future.

7. REFERENCES

[1] R. Baeza-Yates, B. Ribeiro-Neto, et al. *Modern information retrieval*, volume 463. ACM press, 1999.

[2] S. Bateman, C. Gutwin, and G. McCalla. Social navigation for loosely-coupled information seeking in tightly-knit groups using webwear. In *Proceedings of the 2013 conference on Computer supported cooperative work*, pages 955–966. ACM, 2013.

[3] P. Bennett, R. White, W. Chu, S. Dumais, P. Bailey, F. Borisyuk, and X. Cui. Modeling the impact of

short-and long-term behavior on search personalization. In *Proceedings of the 35th ACM SIGIR conference on Research and development in information retrieval*, pages 185–194, 2012.

[4] P. Borlund. Experimental components for the evaluation of interactive information retrieval systems. *Journal of documentation*, 56(1):71–90, 2000.

[5] A. Diriye and G. Golovchinsky. Querium: a session-based collaborative search system. In *Advances in Information Retrieval*, pages 583–584. 2012.

[6] R. Farzan. *A study of social navigation support under different situational and personal factors*. PhD thesis, University of Pittsburgh, 2009.

[7] R. González-Ibáñez, M. Haseki, and C. Shah. Let's search together, but not too close! an analysis of communication and performance in collaborative information seeking. *Information Processing & Management*, 49(5):1165–1179, 2013.

[8] S. Han, D. He, Z. Yue, and P. Brusilovsky. Supporting cross-device web search with social navigation-based mobile touch interactions. In *User Modeling, Adaptation and Personalization - 23rd International Conference, UMAP 2015, Dublin, Ireland, June 29 - July 3, 2015. Proceedings*, pages 143–155, 2015.

[9] S. Han, Z. Yue, and D. He. Understanding and supporting cross-device web search for exploratory tasks with mobile touch interactions. *ACM Transactions on Information Systems (TOIS)*, 33(4):16, 2015.

[10] J. Hyldegård. Collaborative information behaviour - exploring kuhlthau's information search process model in a group-based educational setting. *Information Processing & Management*, 42(1):276–298, 2006.

[11] J. Jiang, S. Han, J. Wu, and D. He. Pitt at trec 2011 session track. In *TREC*, 2011.

[12] C. Kohlschütter, P. Fankhauser, and W. Nejdl. Boilerplate detection using shallow text features. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 441–450, 2010.

[13] C. C. Kuhlthau. Inside the search process: Information seeking from the user's perspective. *JASIS*, 42(5):361–371, 1991.

[14] M. Morris. A survey of collaborative web search practices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1657–1660, 2008.

[15] M. Morris. Collaborative search revisited. In *Proceedings of the 2013 conference on Computer supported cooperative work*, pages 1181–1192, 2013.

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

[17] J. Pickens, G. Golovchinsky, C. Shah, P. Qvarfordt, and M. Back. Algorithmic mediation for collaborative exploratory search. In *Proceedings of the 31st ACM SIGIR conference on Research and development in information retrieval*, pages 315–322, 2008.

[18] C. Shah. Coagmento-a collaborative information seeking, synthesis and sense-making framework. *Integrated demo at CSCW*, pages 6–11, 2010.

[19] C. Shah. Collaborative information seeking. *Journal of the Association for Information Science and Technology*, 65(2):215–236, 2014.

[20] C. Shah and R. González-Ibáñez. Exploring information seeking processes in collaborative search tasks. *Proceedings of the American Society for Information Science and Technology*, 47(1):1–7, 2010.

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

[22] C. Shah, J. Pickens, and G. Golovchinsky. Role-based results redistribution for collaborative information retrieval. *Information processing & management*, 46(6):773–781, 2010.

[23] X. Shen, B. Tan, and C. Zhai. Context-sensitive information retrieval using implicit feedback. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 43–50, 2005.

[24] L. Soulier, C. Shah, and L. Tamine. User-driven system-mediated collaborative information retrieval. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 485–494, 2014.

[25] B. Tan, X. Shen, and C. Zhai. Mining long-term search history to improve search accuracy. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 718–723, 2006.

[26] R. W. White, W. Chu, A. Hassan, X. He, Y. Song, and H. Wang. Enhancing personalized search by mining and modeling task behavior. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1411–1420, 2013.

[27] Z. Yue, S. Han, and D. He. An investigation of search processes in collaborative exploratory web search. *Proceedings of the American Society for Information Science and Technology*, 49(1):1–4, 2012.

[28] Z. Yue, S. Han, and D. He. Modeling search processes using hidden states in collaborative exploratory web search. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, pages 820–830, 2014.

[29] Z. Yue, S. Han, D. He, and J. Jiang. Influences on query reformulation in collaborative web search. *Computer, IEEE*, (3):46–53, 2014.

[30] C. Zhai and J. Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *Proceedings of the tenth international conference on Information and knowledge management*, pages 403–410, 2001.

[31] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 334–342, 2001.