Exploring User Behavior in Email Re-Finding Tasks

Joel Mackenzie*
RMIT University
Melbourne, Australia
joel.mackenzie@rmit.edu.au

Kshitiz Gupta
Microsoft
Bellevue, WA, United States
kshitiz.gupta@microsoft.com

Fang Qiao
Microsoft
Bellevue, WA, United States
fang.qiao@microsoft.com

Ahmed Hassan Awadallah
Microsoft Research
Redmond, WA, United States
hassanam@microsoft.com

Milad Shokouhi
Microsoft
Bellevue, WA, United States
milads@microsoft.com

ABSTRACT

Email continues to be one of the most commonly used forms of online communication. As inboxes grow larger, users rely more heavily on email search to effectively find what they are looking for. However, previous studies on email have been exclusive to enterprises with access to large user logs, or limited to small-scale qualitative surveys and analyses on limited public datasets such as Enron[1] and Avocado[2]. In this work, we propose a novel framework that allows for experimentation with *real* email data. In particular, our approach provides a realistic way of simulating email re-finding tasks in a crowdsourcing environment using the workers' personal email data. We use our approach to experiment with various ranking functions and quality degradation to measure how users behave under different conditions, and conduct analysis across various email types and attributes. Our results show that user behavior can be significantly impacted as a result of the quality of the search ranker, but only when differences in quality are very pronounced. Our analysis confirms that time-based ranking begins to fail as email age increases, suggesting that hybrid approaches may help bridge the gap between relevance-based rankers and the traditional time-based ranking approach. Finally, we also found that users typically reformulate search queries by either entirely re-writing the query, or simply appending terms to the query, which may have implications for email query suggestion facilities.

KEYWORDS

email search; user behavior; search interface; search result page; result degradation

ACM Reference Format:
Joel Mackenzie, Kshitiz Gupta, Fang Qiao, Ahmed Hassan Awadallah, and Milad Shokouhi. 2019. Exploring User Behavior in Email Re-Finding Tasks. In *Proceedings of the 2019 World Wide Web Conference (WWW '19), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3308558.3313450

1 INTRODUCTION

For 50 years, email has been the mainstay of online communication. Email continues to be used widely for both business and personal communication, with a projected 4.2 billion worldwide users sending over 330 billion emails *per day* by the year 2022 [25]. As cheaper data storage has resulted in an increase in email storage quotas, users are becoming less inclined to delete email [12]. This has implications on the way in which users interact with their email client; they need to be more persistent in re-finding tasks, especially if the email is ranked by time. To this end, recent work has explored the use of relevance ranking in email search, showing effectiveness improvements with respect to ranking by time [2, 12, 14]. Although this is a promising step towards improving the email search experience, there is still a large gap in understanding how users interact with email search systems, particularly those that use relevance ranking. This is due to the difficult nature of experimenting within this context – email data is highly personal, private, and exclusive to commercial email providers. Additionally, users are often interested in finding a single known message [18].

Based on this knowledge, we propose a novel offline evaluation framework that allows us to examine how users behave on realistic email search tasks using their personal email data. Our approach allows for a vast number of experiments to be conducted across realistic email search scenarios, using real data from real users. For instance, we examine user behavior in email search tasks using both the traditional rank-by-time result presentation, as well as more modern rank-by-relevance and hybrid ranking approaches as illustrated in Figure 1. Our framework also allows us to compute implicit success metrics such as Session Success Rate (SSR), in addition to relevance-based metrics such as Mean Reciprocal Rank (MRR) [46], for different settings. We also compare the query reformulation of users across various experimental settings. We focus on the following research questions:

RQ1: *How does user behavior differ with respect to different email ranking approaches?*

RQ2: *How do the characteristics of email impact re-finding success?*

RQ3: *How does search quality affect user behavior in email re-finding?*

In answering these questions, our main contributions are as follows:

1. We describe a novel approach for simulating email re-finding tasks that can be used for user behavior studies and offline evaluation experimentation in crowdsourcing environments (Section 2), and

[*] Research was conducted while at Microsoft Research.
[1] https://www.cs.cmu.edu/~./enron/
[2] https://catalog.ldc.upenn.edu/LDC2015T03

WWW '19, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6674-8/19/05.
https://doi.org/10.1145/3308558.3313450

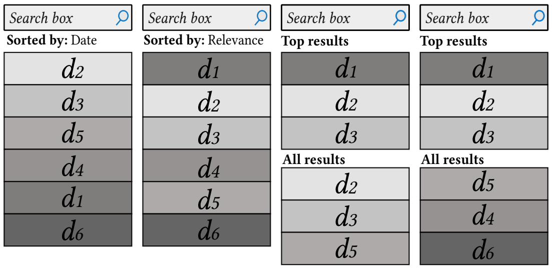


Figure 1: A pictorial example of the four different systems that we compare using the offline evaluation framework introduced in this paper. From left to right: Time (T_0): ranked by time, Rel (T_1): ranked by relevance, HybDup (T_2): hybrid relevance-time with duplicates, and Hyb (T_3): hybrid relevance-time without duplicates. Following the example of Carmel et al. [14], we denote message freshness by the shade of the cell, and we denote the relevance ranking as the document subscript.

2. We leverage this approach to characterize user behavior in email re-finding tasks across a wide range of factors such as ranking quality, SERP format, and email type (Section 4).

The remainder of the article is organized as follows: Section 2 provides some brief background and outlines our novel framework for simulating email re-finding, Section 3 describes the analysis we conduct, Section 4 reports on the experimental results, Section 5 validates our experimental framework, Section 6 outlines related work in user behavior analysis and email ranking, and Section 7 summarizes and concludes this article.

2 SIMULATING EMAIL RE-FINDING

Personal search tasks (e.g. email search) have been very difficult to study in realistic settings due to the nature of the email search process and the limitations on using and sharing email data. Email search is often characterized as a re-finding task [18] where a user is looking for a *specific known message* rather than looking for general information or exploring [4]. Additionally, since email data is highly personal, queries and relevance annotations are not readily available for both learning to rank email and evaluating email search systems. While synthetic approaches for data generation have been explored previously [1], they do not provide a realistic setting since the email data does not belong to the user conducting the search task. Instrumenting email clients was also proposed [20, 22, 30]; while allowing for a realistic setting, this approach provides no control over the experimental system. Hence, it is more suitable for logging user interactions rather than experimentation. This is analogous to using online experimentation and log gathering [12, 14] in large scale search settings, which is not possible without a real system and thousands or millions of users. Here, we introduce a novel approach for simulating email re-finding tasks. Our approach is deployed as crowdsourced *human intelligence tasks* (HITs), making it suitable for academic settings, while also allowing *real* email interaction data to be studied in a realistic and privacy-preserving manner.

2.1 Process

The task is broken into three distinct stages, each shown in Figure 2. First, the user is shown an email message e_t, known as the *target* message, which is sampled from the most recent N emails in their *inbox* folder (we set N to 1000 in our experiments). This screen is shown to the user for t_{read} seconds, in which the user can read the body of the message (Figure 2, left). Based on some preliminary experimentation, we set $t_{read} = 5$ seconds. This provides enough time for the user to 'get the gist' of the target email without learning or memorizing particular phrases from it. After t_{read} seconds have lapsed, the user is then shown a second screen with a questionnaire (see Section 2.2). Although we do not limit the amount of time a user can spend answering the questionnaire, we denote the time they take in this phase as $t_{question}$ (Figure 2, center). In the final stage of the task, the user is presented with a standard email search screen. On the top, a search bar is present, with the search engine results page (SERP) on the left side of the page and the body of the *current* email shown on the right (Figure 2, right). The user may scroll down the SERP and click on any of the emails to make them appear in full on the right pane. Once the user locates and opens the target message, a pop-up message informs the user that they successfully located the message, and the task is ended. For this phase of the task, the user may spend up to t_{search} seconds. If they do not find the target email in this time, then the search task is deemed as failed. We set $t_{search} = 120$ seconds. When showing the SERP, we show 20 documents in total. Although commercial providers generally provide more results to the user (such as the 50 per page[3], or an infinite scroll[4]), serving just 20 results (rather than 50 or more) improves the efficiency of the API calls, minimizing the impact of latency on user behavior [7, 10]. We leave exploration of alternative SERPs as future work. Note that we also allow the user to enter free-form feedback upon the completion or failure of a task if they wish.

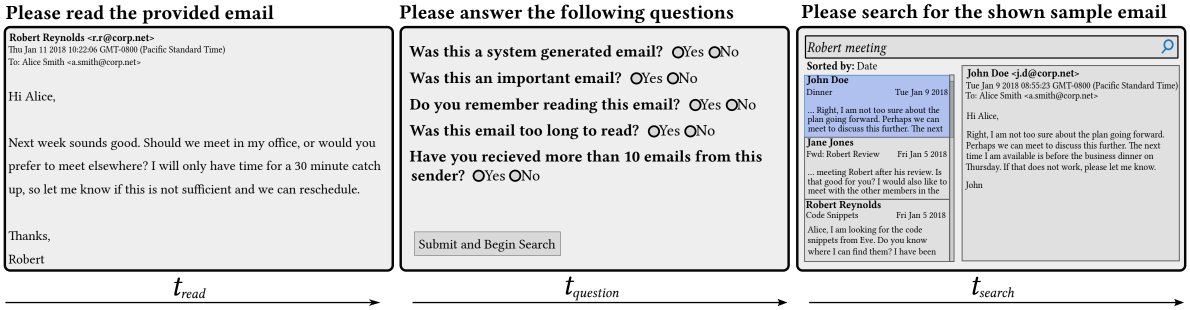
2.2 Questionnaire

For completeness, we now outline the questions presented to the users. All questions require *yes* or *no* answers, making them less cognitively expensive than other formats (such as those employing a Likert scale). We asked the worker seven questions in total about aspects of the target message:

- Did you reply to this email?
- Is this the type of email that you would likely search for?
- Was this email a system generated email?
- Was this email important to you?
- Do you get more than 10 emails from this sender in a week?
- Do you remember reading this email before?
- Was the email too long to read?

The aim of this questionnaire, besides offering a momentary distraction from the target email content, is to gain insight into the importance of the target email as perceived by the user. Furthermore, this allows additional quality control of the judges, as answers to some of the questions can be validated using the search API, such as whether the message was read, replied to, marked as important, and so on.

[3] https://mail.google.com/
[4] https://outlook.office.com/owa/



Please read the provided email

Robert Reynolds <r.r@corp.net>
Thu Jan 11 2018 10:22:06 GMT-0800 (Pacific Standard Time)
To: Alice Smith <a.smith@corp.net>

Hi Alice,

Next week sounds good. Should we meet in my office, or would you prefer to meet elsewhere? I will only have time for a 30 minute catch up, so let me know if this is not sufficient and we can reschedule.

Thanks,
Robert

Please answer the following questions

Was this a system generated email? ○Yes ○No

Was this an important email? ○Yes ○No

Do you remember reading this email? ○Yes ○No

Was this email too long to read? ○Yes ○No

Have you recieved more than 10 emails from this sender? ○Yes ○No

Please search for the shown sample email

Robert meeting

Sorted by: Date

John Doe
Dinner Tue Jan 9 2018
... Right, I am not too sure about the plan going forward. Perhaps we can meet to discuss this further. The next

Jane Jones
Fwd: Robert Review Fri Jan 5 2018
... meeting Robert after his review. Is that good for you? I would also like to meet with the other members in the

Robert Reynolds
Code Snippets Fri Jan 5 2018
Alice, I am looking for the code snippets from Eve. Do you know where I can find them? I have been

John Doe <j.d@corp.net>
Tue Jan 9 2018 08:55:23 GMT-0800 (Pacific Standard Time)
To: Alice Smith <a.smith@corp.net>

Hi Alice,

Right, I am not too sure about the plan going forward. Perhaps we can meet to discuss this further. The next time I am available is before the business dinner on Thursday. If that does not work, please let me know.

John

Figure 2: A pictorial example of our experimental framework. First, the user is shown an email sampled from their inbox for t_{read} seconds. Next, they must answer a short questionnaire. Finally, the user is taken to the search window where they have t_{search} seconds to find the email that they were shown. Once the user clicks on the target email, the task is ended, and the user is taken to a new task.

2.3 Metrics

Since we are interested in characterizing user behavior, we employ a range of measurements for comparing behavior across different treatments. Although these types of measurements are quite standard [29, 36, 40], we elucidate them for clarity. We also note that all time-based measures are calculated from the time in which the user enters the *search* phase of the HIT.

Task level interactions.

- *Success rate:* The percentage of successful tasks for the given configuration.
- *Time to success:* The time taken for the user to find the target message.
- *Number of queries:* The total number of issued queries.
- *Rate of abandonment:* The rate of reformulation without any message accesses.

Query level interactions.

- *Query action:* Reformulate/Succeed/Fail.
- *Time per SERP:* Average time spent per SERP.
- *Number of clicks:* The average number of clicks per SERP.
- *Query length:* The length (number of terms) of the submitted query.
- *Rank of clicks:* The average click depth.
- *Lowest ranked click:* The average rank of the lowest document accessed per SERP.
- *Time to first click:* The time taken for the user to click the first accessed document.
- *Time to deepest click:* The time taken for the user to click the deepest accessed document.

These interaction signals can also be regarded as online metrics, which would commonly be collected and analyzed in online systems. One disadvantage in real online systems is that the ground truth is not known, meaning that typical offline effectiveness metrics are not easy to deploy. Since we have access to the ground truth, we can deploy offline effectiveness metrics here. We use *Mean Reciprocal Rank* to measure the performance of the various systems across all provided queries, which is defined as

$$\mathrm{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{e_i} \tag{1}$$

where Q is the set of queries, and e_i is the rank of the target email in the SERP for query i. MRR is a suitable metric for a re-finding task as there is only *one* email that the user is looking for.

2.4 Prerequisites

We rely on the Microsoft Graph API[5] as an interface between the workers' mailbox and our task. The API provides all query matching and ranking functionality. Therefore, a crowdworker must have a Microsoft managed email address such as one at `outlook.com, live.com, hotmail.com,` or `msn.com` to be eligible to enroll in our task. Alternatively, they are provided with an option to import their inbox into an Outlook inbox such that the API is available for use. Other email systems have similar APIs. For example, our system also supports the Gmail API[6], but we only present results based on the Microsoft Graph API here for simplicity.

3 SETTINGS

In the previous section, we described our approach for simulating email re-finding tasks. In this section, we describe the experimental settings for the study we conducted to answer the research questions outlined earlier.

3.1 Search Results Grouping

We explore three ranking conditions that are found in real email systems. The first refers to the traditional approach of ranking all email that matches the query by *time*, the most common technique for ranking email. The second approach refers to ranking email by *relevance*, whereby the results are ranked by their estimated relevance to the user query. Finally, the *hybrid* approach refers to providing the top-3 results according to the relevance model, followed by the standard ranking of matched emails by time [14, 42].

3.2 Search Results Quality

Our experimental framework allows us to purposefully degrade the quality of the search results [27, 45], which is useful for both validation and experimental analysis. Firstly, it allows us to validate that our metrics align with the user experience of the system; when the quality of the results is degraded, we expect to observe the user taking longer to find the target email, inputting more searches per

[5] https://developer.microsoft.com/en-us/graph/
[6] https://developers.google.com/gmail/api/

task, and so on. Secondly, it enables us to add another condition to our study by degrading the quality of the search results and observing how users behave when they are failing. For our system, we employ a very simple quality degradation mechanism. On a per-task basis, we set some probability P that we will remove the target, e_t, from the SERP. Then, on a per-query basis, we remove e_t with probability P. We plan to look at alternative degradation approaches in future work, where the target email can be demoted in various ways [45].

3.3 Search Results Duplication

Since the hybrid ranking uses both relevance and time ranking, it is possible that a document that is surfaced as being in the top-3 most relevant is also present in the most recent documents too. As such, we decided to study another condition to help us understand how duplicating the documents in the hybrid ranking affects user behavior [14, 42]. We instantiate two versions of the hybrid system with respect to the duplication condition: one version allows documents appearing in the "top results" pane to also appear in the "all results" section of the SERP, whereas the other version removes any duplicates found further down the ranking.

Based on the aforementioned conditions, we experiment with four unique systems, as shown in Figure 1. In practice, the two hybrid systems are very similar, though the Hyb system is guaranteed to have the same or better performance as compared to the HybDup system in terms of MRR, given the exclusion of duplicate documents.

Furthermore, note that our systems are similar but not directly comparable to those discussed by Carmel et al. [14], and we refer the interested reader to this work for both textual and graphical explanations of their systems. For our experiments, we treat Time as the control system, and measure differences between Time and all other systems using Dunnett's multiple comparison test [19].

4 EXPERIMENTS

We now employ our experimental approach to gain insight into how various rankings and interfaces impact user behavior and success in email re-finding tasks.

4.1 Users and Data

We employ 53 unique *trained* [28] judges to perform the given tasks; all judges were required to complete a set of *training tasks* before they were enrolled in the data collection tasks. Judges are not paid per HIT, but are paid an hourly rate, which we believe reduces incentive to game the tasks. Each judge could complete up to 90 tasks, and task configuration was randomized between tasks, meaning that each judge was equally likely to see each variation of the system. We also vary the probability of degradation, P, on a per-task basis, with $P \in \{0.0, 0.25, 0.50\}$. The judges completed a total of 3,678 tasks across a two-week period in August, 2018. Table 1 summarizes the data on both a per-judge level, and on a per-task level. Clearly, there is some variance in each measurement, as task difficulty can vary depending on the mailbox of the user, their search strategy, and so on. Although these statistics reflect expectation, we found that the age of the sampled emails is much older than anticipated, with a mean age of 142 days, and a median age of 67 days. Since we sample from the most recent 1,000 emails, this indicates that the judges do not receive a high volume of email (on average) to these personal accounts. Table 2 summarizes the questionnaire data across each HIT. Most emails are classified by users as machine generated [34], which is not surprising since we are using personal web email accounts. Interestingly, users recalled reading around 10% of the sampled messages, whereas the API reported that 31% of these had been read. This can be explained in several ways. Firstly, users may not have perfect recall of whether they read an email or not. This could be related to the importance of the email, receiving a large volume of similar email, etc. Secondly, the API can only tell us if the email is marked as read, not whether the user actually reads the email. Another interesting observation is that only around 4% of the sampled mail was replied to, as contrasted with other studies that report reply rates of around 15% [49] and 20% [38]. Note that these studies were focused on enterprise data, where responses are more likely as compared with personal mail scenarios, where a lower reply rate be explained by the high volume of machine generated messages that saturate the user's inbox [34].

Table 1: Summary statistics on a per-judge level (top) and on a per-task level (bottom) for 53 unique judges across a total of 3,678 records.

	Mean (SD)	Median	Min	Max
Tasks	69.4 (12.8)	67	30	83
No. Successful	61.0 (13.1)	64	22	81
TTS per task	45.4 (11.0)	44.5	22.4	76.9
Queries per task	1.8 (0.4)	1.8	1.1	2.9
Clicks per task	2.6 (2.4)	1.7	0.9	12.9
Age [days]	141.9 (219.0)	64	0	1,911
TTS	45.3 (36.4)	30.4	3.7	120.0
No. queries	1.8 (1.4)	1	1	17
No. clicks	2.4 (4.9)	1	0	100

Table 2: Distribution of the questionnaire data collected from our HITs, shown as percentages. Each question pertains to the target email for the given task.

Email property	Percentage marked true
Machine Generated	72.6
Read [User Response]	9.8
Read [API]	30.6
Replied	4.2
Important	16.3
Popular Sender	30.1
Likely to Search	16.8

4.2 Ranker Effectiveness

Our first experiment evaluates the effectiveness of the various rankers we deploy. For the first query from each task, we retrieve the ranked list for each ranker, and evaluate the reciprocal rank for that particular ranked list. Figure 3 shows the MRR for each system for target emails of different ages and different input query lengths.

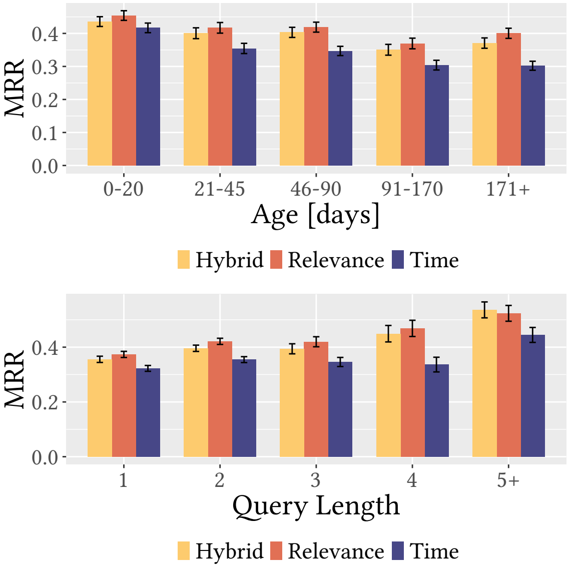


Figure 3: MRR and standard error for each system with respect to email age (top) and query length (bottom). Email ages are grouped such that ≈ 20% of all tasks are shown in each bucket.

Since we found that both hybrid systems have the same performance across this query set, we show only one *hybrid* bar in this analysis. Relevance based ranking provides the most effective ranking, followed closely by the hybrid ranking. As expected, ranking by time is most effective for new mail, as there is less chance that the ranker will surface matching (but irrelevant) messages. Even so, ranking by time is still inferior to the alternatives, at least with respect to MRR. Additionally, as query length increases, so too does the MRR for each ranker, which confirms the results from Carmel et al. [14]. The reason that longer queries are more effective in the email search paradigm is because email ranking is still based mostly on matching, whereas other applications such as web search rely heavily on click-through data and anchor text which is not readily available for email ranking. Interestingly, we found that most queries submitted in this experiment are either one (36%) or two (36%) terms, with only 15% containing three, 6% containing four, and 7% containing five or more terms, respectively. This supports observations from prior work; even though longer queries result in a better ranking, users opt to submit very short queries [13, 14, 30]. Using a two-tailed t-test with a Bonferroni correction, we find that both the hybrid and the relevance rankers significantly outperform the rank-by-time system ($p < 0.01$).

4.3 Behavior Comparison

We now evaluate the four different systems with respect to the interactions discussed in Section 3, thereby addressing RQ1. For this experiment, we are interested only in evaluating the systems when there is no quality degradation; after filtering out tasks where $P > 0$, we retain approximately 400 tasks per system.

User Behavior. Table 3 summarizes the observations we collected from the search task. Firstly, we observe that the success rates across the systems are very similar, with most tasks being completed successfully, while also aligning with rates observed in prior work [6]. It is also clear that the time to success does not vary widely between the systems, though the standard deviations are quite large. This suggests some variance in the difficulty of the search tasks. Users tend to issue less queries on the HybDup system, although not statistically significantly so. Users also submit less queries per task on systems other than Time, though the difference is small. Interestingly, the rate of abandonment is higher for Time than the other systems, which may suggest that users are more aware of *when* they should abandon and reformulate in a rank-by-time scenario, especially since this is what most users are accustomed to.

Looking more deeply at the query level analysis, we observe that tasks on Rel and Hyb result in statistically significantly shallower clicking on average compared to those on Time. Furthermore, the deepest clicks on Hyb were statistically significantly shallower than those on Time. Intuitively, this makes sense, because the MRR for these systems is significantly higher than on the rank-by-time system (Time). However, it is surprising that the same observation is not true for HybDup. One possibility is that, since HybDup is simply the same as Time with the addition of three ‘champions’, users can more easily fall back to the same search style they employ for Time, whereas the same notion is not true for Rel and Hyb. To further investigate this phenomenon, we analyze the tasks from HybDup which contain e_t in both the top-3 ‘champions’ section of the ranker, as well as the rank-by-time section. It is expected that users begin at the top-ranked result and work down the ranked list, one document at a time, until they find the target document. However, we found that 6% of the time, users would click the *lower-ranked* instance of e_t, that is, in the rank-by-time part of the SERP. This suggests that some users may occasionally either miss or glance beyond the target document, or may just prefer to not examine the top-results. Although rarely used, this provides evidence of some benefit of duplication of top-results, and helps explain why users preferred duplication in the study from Carmel et al. [14].

In summary, we expected that the relevance and hybrid systems would result in significantly different behavior as compared to the system that ranks documents by time, as they are significantly better rankers based on MRR. However, we found that the ranking employed by the search systems does not largely impact user behavior (RQ1). One explanation for this outcome is that MRR is more sensitive than users are, in the sense that a statistically significant change in MRR does not necessarily mean that a user will notice such change.

Failure Analysis. While it is clear that users are generally quite good at successfully completing the provided task (Table 3), there are a small number of cases where users were unable to do so. To learn more about what caused users to fail, we conducted a failure analysis across these tasks (where $P = 0.0$). We define two types of failure: *Unforced failures* occur when the target document was surfaced on at least one of the SERPs that was viewed by the worker. That is, the user would have been able to successfully complete the task with at least one of the SERPs they were shown. The remainder of the failures are characterized as *forced failures*, where the user had no chance of completing the task since their queries never surfaced the target. There were 25, 33, 25, and 29 failed tasks for systems

Table 3: Summary table of user behavior across the four systems using mean and SD. Only tasks with no degradation are considered. The rank of clicks, and deepest clicks, consider only queries in which at least one document was clicked.

Measure	Time (T_0)	Rel (T_1)	HybDup (T_2)	Hyb (T_3)
		Task Level		
Success Rate	93.8	92.2	93.9	92.7
TTS	36.70 (32.08)	36.37 (32.40)	34.15 (29.70)	37.22 (32.12)
Number of queries	1.52 (1.00)	1.50 (1.27)	1.40 (0.98)	1.45 (1.05)
Rate of abandonment	0.17 (0.29)	0.13 (0.26)	0.13 (0.28)	0.14 (0.28)
		Query Level		
Time on SERP	25.42 (18.91)	26.55 (20.72)	26.02 (19.38)	27.56 (21.63)
Number of clicks	1.36 (2.84)	1.60 (3.67)	1.39 (2.24)	1.18 (1.74)
Length of queries	2.28 (1.47)	2.23 (1.46)	2.07 (1.28)†	2.28 (1.39)
Rank of clicks	4.19 (4.19)	3.51 (3.98)†	4.02 (4.38)	3.33 (3.73)‡
Deepest click	4.75 (4.89)	4.16 (4.89)	4.58 (4.99)	3.77 (4.18)‡
Time to first click	11.90 (15.39)	12.24 (14.98)	10.99 (13.54)	13.03 (16.75)
Time to deepest click	12.94 (15.39)	13.25 (14.97)	12.23 (13.90)	13.65 (16.99)

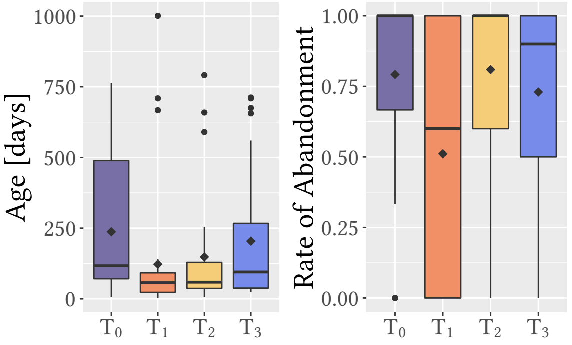


Figure 4: The age of the target email (left), and the abandonment rate within the task (right) for unsuccessful tasks.

Time, Rel, HybDup, and Hyb, respectively, which represents around a 6−7% rate of failure. Of these, 4, 6, 4, and 0 were unforced, meaning that the majority of failures were due to users not being able to surface the target document at all. Figure 4 (left) shows the age of the target emails from the failed tasks on a per-system basis. Clearly, users struggle to surface older documents on Time. The converse is true for the relevance based ranking in Rel, where most failure cases were emails that were received within the last ≈ 125 days. The two hybrid systems appear somewhere between Time and Hyb. Figure 4 (right) shows the rate of abandonment across the failed tasks. Users abandoned the resultant SERPs at a very high rate, suggesting that they were quite confident that their query had not surfaced the target email, opting to reformulate instead. Interestingly, users were less likely to abandon the relevance based SERP (Rel) than those of the other systems. One explanation is that users have a much stronger intuition for when a time-ranked SERP has not provided the desired result in comparison to a relevance-based SERP.

Table 4: Examples of the different types of reformulation that were observed from our log and the percentage in which they are observed. 0.3% of cases did not fall into any of our categories, and are thus omitted here.

Type	Example	% Obs.
Full reform.	*grammarly* ⟶ *writing*	35.7
Partial reform.	*cv attached* ⟶ *cv provided*	11.6
Generalize	*out for delivery* ⟶ *delivery*	6.9
Specialize	*amazon* ⟶ *amazon order number*	23.3
Typo.	*fornt desk* ⟶ *front desk*	18.6
Revert	*new account* ⟶ *bank* ⟶ *new account*	3.6

4.4 Reformulation

Continuing our analysis of user behavior across the varying systems, our next experiment investigates how users reformulate their queries when they cannot find what they are searching for. We refer to a reformulation as the modification of query q which results in the next successive query q'. For this experiment, we consider all tasks where a reformulation was made, irrespective of the ranking system and degradation probability. In total, there were 2,794 reformulations from 1,445 tasks.

Categorizing Reformulations. In order to categorize the reformulation strategies, we first examine the query trace from each task with more than one query to understand what users are doing when they reformulate. Based on this analysis, we came up with six major strategies as defined in Table 4, similar to those defined by Jansen et al. [26] and Hassan [23]. Note that the strategy defined as *partial reformulation* refers to a combination of adding and removing terms where q' contains at least one term from q.

Next, we use a semi-automated process to categorize each subsequent query given all previous queries within the task. This process uses features such as the bag-of-words from each query, the edit distance from q to q', and the term ordering to determine the category of the reformulation. The output was then validated manually.

Table 4 also shows the proportion of the strategies that were employed. The most popular reformulation action is to generate an entirely new query, which occurs around 35% of the time. After further examination, it became clear that these full reformulations often came from users changing their search strategy. For example, many of the full reformulations involved users initially trying to use a keyword oriented query to find relevant results, and then reformulating to use a date instead (eg, *'amazon delivery'* → *'june 2018'*). The next most popular reformulation action is to specialize the previous query. This suggests that users were trying to improve precision. This is also a likely strategy for users of email search systems since queries are generally very short. Another common action was to fix typographical errors, which made up around 19% of all reformulations. It would be interesting to see how the proportion of typographical errors changes if query autosuggestion [13] was made available, but we leave this for future work. We also note that the relative proportions of reformulation strategies did not change with respect to the ranking algorithm used, which indicates that users are not sensitive to the ranker when considering their reformulation approach.

Comparing these findings to web search scenarios, we find that users typically reformulate typographical errors and generalize query terms with similar rates to those seen in web search, but are more likely to construct full reformulations in email r-finding [23]. Additionally, while other works have reported a reasonably large (up to 20%) use of advanced search operators such as '`from:`' or '`to:`' [4, 30], we observed almost no such occurrences (< 1% of submitted queries). We postulate that this is because these studies were from corporate environments, where users may be more accustomed to searching using operators. We must also note that we did not give the users any information on whether or not our system could handle such operators.

Behavior when Reformulating. We also examine the user behavior measures across successive reformulations to see if there is evidence of users becoming frustrated (or of any other changes in behavior). We found that, in general, user behavior did not change markedly across successive reformulations with one exception; as users continue to reformulate, they typically spend less time on the resultant SERP (Figure 5, left). Although we initially believed this to be due to increased time pressure [17], examining the time spent on each SERP reveals that users (generally) have used less than 50% of their allocated time at the time of their final reformulation. Next, we investigated the number of *unseen* documents that are surfaced in the SERP for each subsequent reformulation (Figure 5, right). Evidently, as users continue to reformulate, they are observing fewer new documents, which explains why they spend less time on the SERP. Similar trends were found across all system configurations.

Supplementing the answers to RQ1, our findings show that users typically submit short queries, and prefer to either specialize them, or to write entirely new queries when they cannot locate the target document, irrespective of the ranking quality or system in use. Furthermore, we observed that users tend to spend less time per successive SERP as they reformulate. This is caused by the ranker surfacing many documents the user has already seen, which in turn is caused by the query strategy taken by the user. These findings have implications for ranking in email and personal search; diversification or techniques for re-ranking that account for documents that the user has already seen [44] may be valuable in improving both the user experience and the engagement of the user, especially since it is known that users prefer to submit short queries and specialize as they reformulate.

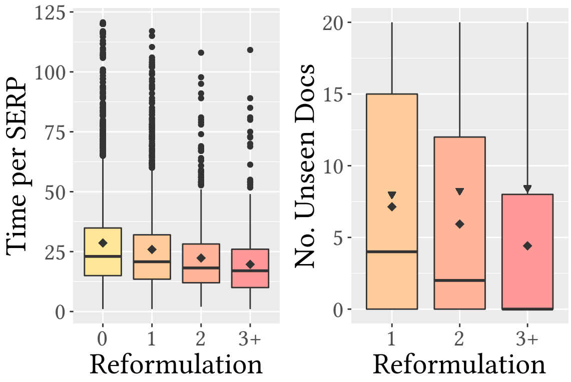


Figure 5: Time per SERP (left) and the number of unseen documents (right) for successive reformulations. On the right plot, the triangles denote the *average rank* of the unseen documents.

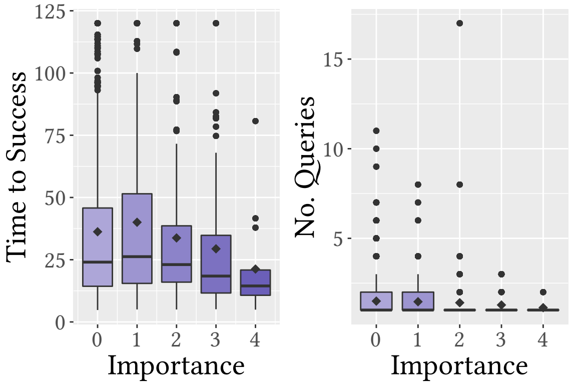


Figure 6: Time to success (left) and number of queries submitted (right) for emails of varying levels of importance.

4.5 Email Properties and Importance

For our final experiment, we explore whether user behavior changes with respect to the properties of the sample email, thus answering RQ2. We define important emails as those that are marked as important, replied to, read, and those that the user is likely to search for. For each message, we use the sum of these importance criteria to define an overall importance score. For example, a message that was read, replied to, and marked as important would receive an importance score of 3. Figure 6 shows the time to success and the number of submitted queries across these levels of importance. We observe that the more important a message is to a user, the easier they are able to find it, issuing less queries and spending less time to succeed. This is because users are more likely to remember certain aspects of messages that are important to them, the converse being true for unimportant messages.

Table 5: Summary of user behavior measures using mean and SD where the target email was sent by a human, or was an automated email. No significance was detected for any of the metrics shown.

Measure	Machine	Human
Task Level		
Success Rate	92.7	93.4
TTS	36.84 (32.45)	35.71 (31.14)
Number of queries	1.45 (1.07)	1.48 (1.09)
Rate of abandonment	0.14 (0.27)	0.15 (0.28)
Query Level		
Time on SERP	26.76 (22.30)	26.17 (18.94)
Number of clicks	1.47 (2.62)	1.34 (2.80)
Length of queries	2.39 (1.61)	2.14 (1.26)
Rank of clicks	3.51 (4.09)	3.89 (4.08)
Deepest click	4.13 (4.89)	4.41 (4.70)
Time to first click	12.62 (16.78)	11.73 (14.29)
Time to deepest click	13.66 (16.90)	12.68 (14.43)

We were also interested in whether users found machine generated emails harder to search for. Our hypothesis is that since the majority of the mail received by the workers is machine generated, this would make searching for such mail more difficult (due to a high volume of similar messages). Table 5 shows the behavior across tasks, divided by whether the target message was generated by a human or a machine. We observe that users tend to submit shorter queries, click less, but go deeper in the ranking when the email is sent from a human, although none of these observations are statistically significant. Answering RQ2, users are able to find important emails faster and with less effort than those that are less important. Furthermore, whether the email is sent by a machine or by a real user tends to have no impact on the effort or success rate of re-finding, indicating that users are quite capable of searching for machine generated emails.

5 VALIDATION

We now perform some analysis to validate the robustness of our experimental framework and collected data.

5.1 Sampling Target Email

A key aspect of our experimental framework is the way in which we sample the target email. As discussed in Section 2 and 3, we sample a random email from the most recent $N = 1{,}000$ emails in a workers inbox, ensuring an adequate sample size. However, this results in a large proportion of email being categorized as email that is not likely to be searched for by the worker (Table 2). Looking further at Table 2, it is clear that many of the sampled emails are in fact generated automatically by machines. While these emails can be useful and important, such as flight tickets, bills, or order invoices, they can also be general newsletters, repeated notifications, and other possibly annoying or low-utility content [34]. It is also important to consider the context in which the user is viewing the email. While some emails are unlikely to be searched *now*, the very same emails may have been sought by the user previously. One such example is a user who is waiting for a delivery from an e-commerce provider. Before the delivery is received, the user may search for the email to find tracking details, for instance. However, once the delivery is received by the user, they are unlikely to be interested in finding this email again. To ensure that the target email sampling does not bias our results, we compared the user behavior and ranking robustness where users classified emails as being either 'likely' or 'unlikely' to be searched for. We did not find any significant differences in user behavior or the performance of the tested systems. Hence, we believe that although our targets may not represent an email that is likely to be searched *now*, the process of re-finding is not impacted by our sampling method.

5.2 Judge Consistency and Quality Control

An important step in analysis of data collected from users is ensuring the quality of the data. Therefore, we conduct a number of analyses to confirm the data quality. First, we explore the click traces of the workers, looking for specific patterns of gaming (such as users clicking each document down the ranking with minimal time delay between clicks). We found very few instances of such behavior (5 occurrences) and removed them from further analysis. Since the target email was randomized and rotated, there were some occasions where a worker had a repeated task at a different point in time (possibly using a different ranking system). From all tasks that we deployed, 271 (7%) of these were repeated. To analyze the consistency of the judges, we compare the submitted queries for the initial task h with the repeated task h'. Our hypothesis is that if judges are consistent, their queries will be similar for the repeat task(s) as they were to the original task. To this end, we consider a judge to be inconsistent if there is no overlap between the submitted queries in h' as compared to h. We found that of the 271 repeat tasks, 51 were classed as inconsistent. Looking further at the inconsistent tasks themselves, it is clear that these inconsistencies arise from the worker employing a different email re-finding strategy, for example, *'2018 conference tickets'* → *'tentative program schedule'*. Furthermore, we only report analysis using trained judges who are trained to conduct our task. However, we did also experiment with employing workers on a per-HIT basis at 40 cents per task. We found the behavior to be consistent with the findings from the trained judges, except that crowdworkers employed on a per-hit basis were generally faster to succeed, which is intuitive given the difference in incentive.

5.3 Controlled Degradation

As discussed in Section 3, we are able to intentionally degrade the search ranking. When collecting data, we opted to run a few different configurations for result degradation, where we remove the target document with probability $P \in \{0.0, 0.25, 0.50\}$. This experiments allows us to study the impact of lower search quality on user behavior and it allows us to validate that our behavior measures move in the expected directions when the ranking quality changes.

Degradation Analysis. Figure 7 shows some of the key measures for each system as P is increased. As expected, the number of submitted queries per task, the time to success, and the rate of abandonment increases steadily as the result quality becomes worse.

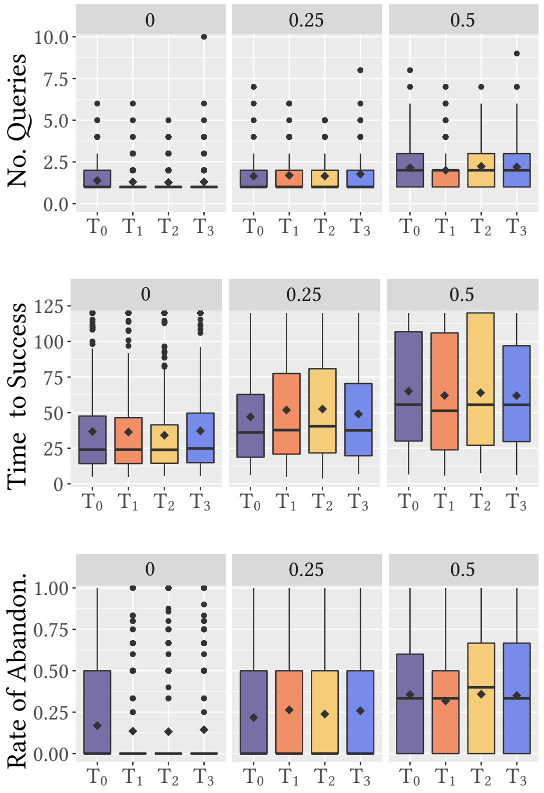


Figure 7: The effect of removing the target email e_t from the SERP with probability P on both the number of submitted queries, the time to success, and the rate of abandonment. Systems are shown in Figure 1. Degradation results in significantly more queries being submitted per task, a significantly longer time to success, and a significantly higher rate of abandonment.

Furthermore, users tend to click deeper as the quality becomes worse. These trends holds across all systems, and are statistically significant under a two-tailed paired t-test with a Bonferroni correction ($p < 0.01$). In addition, the success rates fall across the board as P increases, from around 93% ($P = 0.0$) to 87% ($P = 0.25$) and then to 76% ($P = 0.50$). On the other hand, no statistical significance was measured for the number of clicks, or the time spent on the SERP. This indicates that these metrics are not sensitive to ranking quality, and are unlikely to be informative in the experiments conducted in Section 4. So, in answering RQ3, we have shown that user behavior is correlated with the quality of the search experience, and that a degraded search experience will lead to users taking longer to succeed, searching more often, and abandoning the SERP more often. Contrasting these results to those observed in Section 4 (RQ1) suggests that while users are sensitive to the quality of the search experience, the quality difference (as measured by MRR) must be quite pronounced to observe significantly changed behavior. In any case, the clear correlation between quality and these metrics provides additional evidence supporting the validity of the collected data.

6 RELATED WORK

A multitude of prior work has investigated how users interact with the many aspects of Information Retrieval systems, including presentation, ranking, search task, temporal influences, and many others. Here, we review the relevant studies, including those that focus on user behavior in web search tasks, user behavior and characterization on email tasks, crowdsourcing and gamification, and ranking for email search.

6.1 Understanding User Behavior

Web Search. There are a multitude of studies that focused on web search environments and the many components therein. For example, recent studies have quantified how users behave with respect to the number of documents included in a SERP [29], the information content and length of snippets [36], the difficulty of the querying interface [9], and other aspects of web search such as result presentation, visual interfaces, result coherence, and user typing ability [5, 8, 17, 33, 41]. However, these studies are far less common in personal search environments due to the difficult nature of experimentation on personal data [11]. Our work follows this line of work but focuses on email re-finding tasks.

Email Search. While email use is at an all-time high, email search has not received as much attention from the IR community compared to other areas such as web search. As Carmel et al. [12] pointed out, this is likely due to the lack of publicly available data. While some collections are available, they often have limitations such as using synthetic queries [1], which are not reflective of true user intent on email finding tasks. Most of the prior work on user behavior for email tasks has come from query log analyses, user studies, and surveys. One line of work investigates how users organize their mail and how this impacts their re-finding strategies or actions [4, 6, 21, 38, 47]. Another line of related work explores modeling and predicting various aspects of email search such as how successful a user will be in their search task [30], how likely user will be to respond to new mail [49], how long a user's reply will be [31], and whether a user will defer an email [43], among many others. Elsweiler et al. [20] studied re-finding behavior by instrumenting the Mozilla Thunderbird[7] email client to capture user behavior. Their study uncovered several interesting findings such as that re-finding is difficult for users, and that users can become disoriented while performing re-finding tasks. Other studies have also observed the use of very short queries in email search [4, 13, 14]. Interestingly, users are generally successful at re-finding mail [6, 15], indicating that users are somewhat comfortable with re-finding even if such tasks are thought to be difficult. Our work addresses some of the major limitations of previous work by proposing an approach for simulating email re-finding tasks using real data in a crowdsourcing environment.

Crowdsourcing and Gamification. Crowdsourcing is a simple and cost effective approach for rapidly collecting data from users [32]. Prior work has involved using crowdsourcing to understand how users interact and behave with search tasks. The most relevant work to our approach is that of Ageev et al. [3], who use *gamification* [37]

[7] https://www.thunderbird.net/

to model user behavior in a crowdsourcing environment. This involves proposing a unique and scalable crowdsourcing task that allows user behavior to be captured across informational search tasks with a *known intent*. Subsequent works have focused on how user behavior can be studied via gamification [24], and the implications that such frameworks have on user behavior. Our approach is influenced from these ideas of leveraging crowdworkers with novel systems and tasks to collect user behavior for further analysis, allowing us to bypass many of the aforementioned issues such as experimentation on personalized data.

6.2 Ranking for Email Search

Some interest has been placed on ranking email results by *relevance* instead of the classic *rank-by-time* approach. Many of these ranking approaches were inspired directly from ad-hoc web search, including the use of language models [39] and BM25F [16]. Macdonald and Ounis [35] studied which combinations of email fields are useful as evidence for email ranking, suggesting that both the body and the subject of emails are crucial for effective retrieval.

Recently, the community has focused on the use of machine learning models for ranking email search, and the implications this has on users. Aberdeen et al. [2] outline the approach that Google uses for ranking email in their *Priority Inbox*. In particular, they focus on ranking documents by the predicted probability that the user will interact with the email within some time threshold after delivery. They make this prediction based on *social, content, label,* and *thread* features. They found that internal users of the priority inbox spent 13% less time reading unimportant mail, helping ease *email overload* [31, 48].

Advocating for rank-by-relevance, Carmel et al. [12] experimented with the implementation and deployment of a two-stage ranking model for effective email ranking, showing that a full relevance ranking significantly increases *Mean Reciprocal Rank (MRR)* as compared to rank-by-time.

Using a similar ranking system as described in [12], Ramarao et al. [42] described a new mail system called *InLook*. However, instead of ranking the entirety of the matched search results, InLook displays an *intent pane* which shows the top-3 most relevant matched emails, followed by the standard rank-by-time list. The authors argue that certain query types cannot be sufficiently matched using just rank-by-time or rank-by-relevance, providing justification for the hybrid ranking. Carmel et al. [14] further investigated hybrid rankings, arguing that users are not accepting rank-by-relevance in email search, likely because their search process is ingrained in the rank-by-time model. They propose a very similar approach to Ramarao et al. [42], where they provide the top-*H* results by relevance (called 'Heroes') above the remaining top-*T* by time, and explore a few variants of this hybrid approach. They found that providing the top-3 results based on relevance followed by the standard default time-ordered results, without removing duplicates, provided the best user experience (based on both MRR and a small-scale user study). This ranking, named *Heroes-Dup*, was subsequently deployed in Yahoo corporate and Yahoo web mail A/B testing scenario, significantly improving MRR over the baseline. Our work extended this line of research by exploring the implications that relevance ranking has on user behavior.

7 DISCUSSION AND CONCLUSION

A major contribution of this work is the proposed framework for studying email search using email data from real users in a realistic and privacy-preserving setting. We described an approach that is extensible, and can be modified to study a variety of different aspects of email re-finding, including alternative search interfaces and ranking approaches, among others. We also acknowledge some limitations and opportunities for future work. One interesting capability that was not explored in this work is to control the rank of the target email. Since the target is known before the SERP is shown to the user, the target document could be strategically placed *anywhere* in the ranking. While our approach aims to be as realistic as possible, the random sampling of target emails may not truly reflect an email that a user would search for (Section 5). It would be interesting to see if more robust sampling approaches could be used, such as sampling only emails that are marked as important, or come from specific senders. A final aspect of interest is further gamification of the system, whereby the task could be appropriately modified to include additional incentives for crowdworkers. This would ultimately lead to more robust data collection in situations where *untrained* judges are employed, such as in typical crowdsourcing platforms.

Another major contribution of this work is leveraging our email search experimentation framework for analyzing user behavior in email re-finding tasks. Our results confirmed that relevance based ranking significantly outperforms rank-by-time for email search in terms of MRR, and that rank-by-time performs worse for re-finding older emails. We also showed that although both hybrid and relevance ranked systems are significantly better than the rank-by-time system, in terms of MRR, user behavior (in terms of success rate and time to success) does not markedly differ between the systems. This indicates that MRR may not be entirely representative of utility to users in re-finding tasks (RQ1 and RQ3). We also showed that users tend to do better with deciding when they should abandon the SERP and reformulate the query when rank-by-time is used. Another interesting observation was that some users occasionally either miss or glance beyond the target document when shown in the top results list of a hybrid SERP. Alternatively, they may have just preferred to not examine the top-results and skip to the more familiar rank-by-time setup. These observations show the challenges of trying to introduce new features that require users to change existing behavioral patterns and search strategies.

When studying how users reformulate their searches, we found similar behavior to web search but with more full reformulations which usually result from change in strategy (keyword search to time-based search). Finally, we also explored how properties of email impact re-finding success (RQ2). Most notably, users have more success finding emails that are deemed important. We found no evidence that machine-generated emails are more difficult to find. Our findings have several implications on designing email clients and email search systems. Our proposed framework for experimenting with email search could enable additional work to further study and evaluate email search systems by overcoming the challenges around studying email search interactions in realistic settings.

REFERENCES

[1] S. AbdelRahman, B. Hassan, and R. Bahgat. 2010. A New Email Retrieval Ranking Approach. *IJCSIT* 2, 5 (2010), 44–63.

[2] D. Aberdeen, O. Pacovsky, and A. Slater. 2010. The Learning behind Gmail Priority Inbox. In *NIPS Workshop on LCCC*.

[3] M. Ageev, Q. Guo, D. Lagun, and E. Agichtein. 2011. Find It if You Can: A Game for Modeling Different Types of Web Search Success Using Interaction Data. In *Proc. SIGIR*. 345–354.

[4] Q. Ai, S. T. Dumais, N. Craswell, and D. Liebling. 2017. Characterizing Email Search Using Large-scale Behavioral Logs and Surveys. In *Proc. WWW*. 1511–1520.

[5] H. Ali, F. Scholer, J. A. Thom, and M. Wu. 2009. User Interaction with Novel Web Search Interfaces. In *Proc. OZCHI*. 301–304.

[6] T. Alrashed, A. H. Awadallah, and S. T. Dumais. 2018. The Lifetime of Email Messages: A Large-Scale Analysis of Email Revisitation. In *Proc. CHIIR*. 120–129.

[7] I. Arapakis, X. Bai, and B. B. Cambazoglu. 2014. Impact of Response Latency on User Behavior in Web Search. In *Proc. SIGIR*. 103–112.

[8] J. Arguello and R. Capra. 2014. The Effects of Vertical Rank and Border on Aggregated Search Coherence and Search Behavior. In *Proc. CIKM*. 539–548.

[9] L. Azzopardi, D. Kelly, and K. Brennan. 2013. How Query Cost Affects Search Behavior. In *Proc. SIGIR*. 23–32.

[10] X. Bai, I. Arapakis, B. B. Cambazoglu, and A. Freire. 2017. Understanding and Leveraging the Impact of Response Latency on User Behaviour in Web Search. *ACM Trans. Information Systems* 36, 2 (2017), 21:1–21:42.

[11] R. Capra and M. A. Pérez-Quiñones. 2006. Factors and Evaluation of Refinding Behaviors. In *SIGIR Workshop on PIM*. 16–19.

[12] D. Carmel, G. Halawi, L. Lewin-Eytan, Y. Maarek, and A. Raviv. 2015. Rank by Time or by Relevance? Revisiting Email Search. In *Proc. CIKM*. 283–292.

[13] D. Carmel, L. Lewin-Eytan, A. Libov, Y. Maarek, and A. Raviv. 2017. The Demographics of Mail Search and Their Application to Query Suggestion. In *Proc. WWW*. 1541–1549.

[14] D. Carmel, L. Lewin-Eytan, A. Libov, Y. Maarek, and A. Raviv. 2017. Promoting Relevant Results in Time-Ranked Mail Search. In *Proc. WWW*. 1551–1559.

[15] M. E. Cecchinato, A. Sellen, M. Shokouhi, and G. Smyth. 2016. Finding Email in a Multi-Account, Multi-Device World. In *Proc. SIGCHI*. 1200–1210.

[16] N. Craswell, H. Zaragoza, and S. Robertson. 2005. Microsoft Cambridge at TREC-14: Enterprise track. In *Proc. TREC*.

[17] A. Crescenzi, D. Kelly, and L. Azzopardi. 2015. Time Pressure and System Delays in Information Search. In *Proc. SIGIR*. 767–770.

[18] S. T. Dumais, E. Cutrell, J. J. Cadiz, G. Jancke, R. Sarin, and D. C. Robbins. 2003. Stuff I've Seen: A System for Personal Information Retrieval and Re-Use. In *Proc. SIGIR*. 72–79.

[19] C. W. Dunnett. 1955. A Multiple Comparison Procedure for Comparing Several Treatments with a Control. *J. American Stat. Assoc.* 50 (1955), 1096–1121.

[20] D. Elsweiler, M. Harvey, and M. Hacker. 2011. Understanding Re-finding Behavior in Naturalistic Email Interaction Logs. In *Proc. SIGIR*. 35–44.

[21] M. Grbovic, G. Halawi, Z. Karnin, and Y. Maarek. 2014. How Many Folders Do You Really Need? Classifying Email into a Handful of Categories. In *Proc. CIKM*. 869–878.

[22] M. Harvey and D. Elsweiler. 2012. Exploring Query Patterns in Email Search. In *Proc. ECIR*. 25–36.

[23] A. Hassan. 2013. Identifying Web Search Query Reformulation using Concept based Matching. In *Proc. EMNLP*. 1000–1010.

[24] J. He, M. Bron, L. Azzopardi, and A. de Vries. 2014. Studying User Browsing Behavior Through Gamified Search Tasks. In *Proc. GamifIR*. 49–52.

[25] The Radicati Group, Inc. 2018. Email Statistics Report, 2018–2022. https://www.radicati.com/wp/wp-content/uploads/2018/05/Email-Market-2018-2022-Executive-Summary.pdf. Online; accessed August, 2018.

[26] B. J. Jansen, D. L. Booth, and A. Spink. 2009. Patterns of Query Reformulation during Web Searching. *JASIST* 60, 7 (2009), 1358–1371.

[27] T. Jones, D. Hawking, P. Thomas, and R. Sankaranarayana. 2011. Relative Effect of Spam and Irrelevant Documents on User Interaction with Search Engines. In *Proc. CIKM*. 2113–2116.

[28] G. Kazai and I. Zitouni. 2016. Quality Management in Crowdsourcing Using Gold Judges Behavior. In *Proc. WSDM*. 267–276.

[29] D. Kelly and L. Azzopardi. 2015. How Many Results Per Page? A Study of SERP Size, Search Behavior and User Experience. In *Proc. SIGIR*. 183–192.

[30] J. Y. Kim, N. Craswell, S. T. Dumais, F. Radlinski, and F. Liu. 2017. Understanding and Modeling Success in Email Search. In *Proc. SIGIR*. 265–274.

[31] F. Kooti, L-C. Aiello, M. Grbovic, K. Lerman, and A. Mantrach. 2015. Evolution of Conversations in the Age of Email Overload. In *Proc. WWW*. 603–613.

[32] M. Lease and E. Yilmaz. 2011. Crowdsourcing for Information Retrieval. *SIGIR Forum* 45 (2011), 66–75.

[33] Z. Liu, Y. Liu, K. Zhou, M. Zhang, and S. Ma. 2015. Influence of Vertical Result in Web Search Examination. In *Proc. SIGIR*. 193–202.

[34] Y. Maarek. 2017. Web Mail is Not Dead! It's Just Not Human Anymore. In *Proc. WWW*. 5–5.

[35] C. Macdonald and I. Ounis. 2006. Combining Fields in Known-item Email Search. In *Proc. SIGIR*. 675–676.

[36] D. Maxwell, L. Azzopardi, and Y. Moshfeghi. 2017. A Study of Snippet Length and Informativeness: Behaviour, Performance and User Experience. In *Proc. SIGIR*. 135–144.

[37] M. Meder, F. Hopfgartner, G. Kazai, and U. Kruschwitz. 2016. Third International Workshop on Gamification for Information Retrieval (GamifIR'16). In *Proc. SIGIR*. 1239–1240.

[38] K. Narang, S. T. Dumais, N. Craswell, D. Liebling, and Q. Ai. 2017. Large-Scale Analysis of Email Search and Organizational Strategies. In *Proc. CHIIR*. 215–223.

[39] P. Ogilvie and J. Callan. 2005. Experiments with Language Models for Known-Item Finding of E-mail Messages. In *Proc. TREC*.

[40] K. Ong, K. Järvelin, M. Sanderson, and F. Scholer. 2017. Using Information Scent to Understand Mobile and Desktop Web Search Behavior. In *Proc. SIGIR*. 295–304.

[41] K. Ong, K. Järvelin, M. Sanderson, and F. Scholer. 2018. QWERTY: The Effects of Typing on Web Search Behavior. In *Proc. CHIIR*. 281–284.

[42] P. Ramarao, S. Iyengar, P. Chitnis, R. Udupa, and B. Ashok. 2016. InLook: Revisiting Email Search Experience. In *Proc. SIGIR*. 1117–1120.

[43] B. Sarrafzadeh, A. H. Awadallah, C. H. Lin, C-J. Lee, M. Shokouhi, and S. T. Dumais. 2019. Characterizing and Predicting Email Deferral Behavior. In *Proc. WSDM*. 627–635.

[44] M. Shokouhi, R. W. White, P. N. Bennett, and F. Radlinski. 2013. Fighting Search Engine Amnesia: Reranking Repeated Results. In *Proc. SIGIR*. 273–282.

[45] P. Thomas, T. Jones, and D. Hawking. 2011. What Deliberately Degrading Search Quality Tells Us About Discount Functions. In *Proc. SIGIR*. 1107–1108.

[46] E. M. Voorhees. 1999. The TREC-8 Question Answering Track Report. In *Proc. TREC-8*. 77–82.

[47] S. Whittaker, T. Matthews, J. Cerruti, H. Badenes, and J. Tang. 2011. Am I Wasting My Time Organizing Email? A Study of Email Refinding. In *Proc. SIGCHI*. 3449–3458.

[48] S. Whittaker and C. Sidner. 1996. Email Overload: Exploring Personal Information Management of Email. In *Proc. SIGCHI*. 276–283.

[49] L. Yang, S. T. Dumais, P. N. Bennett, and A. H. Awadallah. 2017. Characterizing and Predicting Enterprise Email Reply Behavior. In *Proc. SIGIR*. 235–244.