Understanding and Modeling Success in Email Search

Jin Young Kim*
Snap Inc.
Venice, CA
jink@snap.com

Nick Craswell
Microsoft
Redmond, WA
nickcr@microsoft.com

Susan Dumais
Microsoft Research
Redmond, WA
sdumais@microsoft.com

Filip Radlinski
Microsoft
Redmond, WA
filrad@microsoft.com

Fang Liu
Microsoft
Redmond, WA
fangliu@microsoft.com

ABSTRACT

Email has been a dominant form of communication for many years, and email search is an important problem. In contrast to other search setting, such as web search, there have been few studies of user behavior and models of email search success. Research in email search is challenging for many reasons including the personal and private nature of the collection. Third party judges can not look at email search queries or email message content requiring new modeling techniques.

In this study, we built an opt-in client application which monitors a user's email search activity and then pops up an in-situ survey when a search session is finished. We then merged the survey data with server-side behavioral logs. This approach allows us to study the relationship between session-level outcome and user behavior, and then build a model to predict success for email search based on behavioral interaction patterns.

Our results show that generative models (MarkovChain) of success can predict the session-level success of email search better than baseline heuristics and discriminative models (RandomForest). The success model makes use of email-specific log activities such as reply, forward and move, as well as generic signals such as click with long dwell time. The learned model is highly interpretable, and reusable in that it can be applied to unlabeled interaction logs in the future.

CCS CONCEPTS

•Information systems → Evaluation of retrieval results;

1 INTRODUCTION

Email remains an important form of communication, with each email account becoming a private corpus of potentially highly personal or business-critical information. Email systems typically have a search interface that retrieves a list of messages matching the query, sorting the messages from newest to oldest. Some interfaces also offer a relevance ranking and multiple metadata-based sortings options, although users may seek out the date-sorted option even when it is not the default [10].

The role of relevance metrics in email search is to evaluate the quality of different ordering and interface options. Do users prefer a particular sort order, or a hybrid ordering with a short relevance-ranked list followed by a longer date-ordered one? Does the preferred sort order vary by query? By user? When there is relevance ranking, what ranking function is best? The matching function that is used to select messages could also be evaluated. Should it use stemming? What email fields should be matched and should there be measures in place to eliminate spurious matches from the list? Matching is be particularly important for date-ordered lists, since a large number of recent spurious matches has the effect of demoting older results that the user may be trying to find.

The metrics developed in this paper can be used to study these different choices. They can also be used to study different users, such as different levels of expertise, different types of information needs, which may lead to differences in desired system interface and results. Although it is important to develop evaluation metrics for email search, there are also challenges in doing so. Because of the sensitive nature of email corpora, it is usually impossible or undesirable to create a traditional test collection, with shared queries, message content and relevance judgments. A notable exception is Gord Cormack's work [9] on email spam detection in which they use an evaluation method that brings the model to private data.

In this work, we turn to server-side email search logs for building evaluation metrics. Modern email systems may operate in the web browser or in a client-server setup that allows the centralized collection of activity logs. For evaluating email search, these logs may be comparable to logs studied in web search.

However, there are some important differences between web and email search logs. Email search logs should be managed in a way that can preserve the privacy of users. The logs studied in this paper do not have any text from the user's query or messages, to maintain the privacy of users, unlike in web search logs where the content of queries and web pages are often studied. This adds extra challenges in finding signals that will be useful for evaluating email search logs.

On the other hand, email activity logs are richer than web search activity logs, because there are a wider range of activities. In web search the main activities are to enter a query and to click a search result, and unless the user returns to the search engine, the search

*Work done while this author was at Microsoft.

SIGIR'17, August 7–11, 2017, Shinjuku, Tokyo, Japan.
 ISBN 978-1-4503-5022-8/17/08…$15.00
DOI: http://dx.doi.org/10.1145/3077136.3080837

logs do not indicate how long the user spent on the clicked document before moving on or closing the browser. By contrast in email search the user remains in the system and their activities can be logged. For instance, after querying and selecting (clicking) a message, the timing and type of their next actions may indicate the relevance of the selected message. Perhaps reading a message for a long time is positive, but another signal of satisfaction would be whether they respond to the message. This may be true even for a short dwell time, in contrast to web search where a short dwell time tends to be a bad sign [12].

Given this availability of rich signals, the next challenge is finding a good interpretation of them, which is a nontrivial task due to the variety of signals (i.e., how would you characterize the value of 'message deletion' during a email search session) and the anonymity requirement for the log data. Our main approach here is to build a model that can predict the success (or failure) of a search session, which in turn can be used to characterize the performance of a search engine.

Building a model of success has been extensively studied in Web search settings [13], mostly relying on annotation of search sessions by human judges. To build such model for email search we need to switch from third party annotation of sessions to in-situ labeling by the searcher [12]. This allows us to understand email search with precision, because all our implicit and explicit data come from the same people in-situ. We develop a metric that can be used directly to evaluate success and effort in email search, and our insights can also be used as a comparison point for more traditional lab or test collection studies of email search.

Our in-situ collection of survey and behavior log data provides three main research contributions. First, we characterize the overall differences between successful and unsuccessful email search, at the user level, session level and the message level. Second, we characterize, for the variety of email actions (such as reply, forward, flag, delete), which actions are associated with successful rather than unsuccessful search sessions. Third, we develop predictive models to distinguish between successful and unsuccessful email search sessions. These insights and success models for email search are new, and should offer a useful comparison point in future email lab studies, test collections and real-world controlled experiments.

2 RELATED WORK

Several areas of prior research are relevant to our current paper, including different forms of offline and online information retrieval evaluation and the analysis of email management strategies and behavior.

Information retrieval evaluation. The dominant form of evaluation for information retrieval research is the test collection, comprising query topics, a corpus of content and relevance judgments [19]. The relevance judgments are usuallt made by third party judges rather than in-situ users as they interact with the search engine. The queries can come from search logs of a real system, but this raises the problem that the third party judge may not be able to understand the context of a real user, such as their real information need and the types of results they prefer. More critically, in our case we can not show email messages to third party judges for privacy reasons.

For this reason it is more desirable to build a behavior-based evaluation metric. In the Web this has been done by getting third party annotators to replay logged user sessions [13] and provide success annotations. Then Markov Chain models are built to characterize the relationship between logged activity and annotations. Later work [2, 3, 17] used web search data and labels collected from a browser toolbar or a game-like interface, and built more elaborate models. Recently the work was also extended to multi-level annotations [14]. Their best explanation for levels of satisfaction was a combination of the outcome of the session and the level of effort. We ask about both outcome and effort in our in-situ survey.

In this work we adapt models from [13] which have the advantage of simplicity and interpretability. However, unlike these studies, in our case we can not replay sessions for third party annotators for privacy reasons. In fact replay based on the server-side logs used in this study is impossible, because the logs do not contain message and query text. Our only source of query and message text (such as message subject) in this study is from the in-situ survey, not the logs.

Because of these privacy concerns, there is a hard tradeoff between having real users with real email and having third party relevance judging. It is not possible to have both, without violating user privacy. In this study we focus on the case with real email and without third party judges, although later it may be possible to validate the other approaches (third party judges with a public email corpus) against our findings. We now consider evaluation approaches that do not rely on third party judging. Table 1 summarizes characteristics of the users, types of labels collected and evaluation metrics for five studies.

In our setting it could be possible to use interleaving [8] to evaluate two retrieval systems, for example comparing query-message matching with and without stemming. We do not study interleaving here, but our findings about dwell time and email actions could inform the interleaving design. Interleaving makes use of implicit behavior to elicit a preference between rankers, and our findings indicate which of the many logged actions in email is associated with search success.

Interactive lab experiments could be a way of studying the effectiveness of email search [11]. Such an approach would usually require some simulated work tasks to be set up. This differs from our in-situ setup, where search is motivated by a real user rather than a simulated task. However, it would be interesting to use the analysis from this paper to interpret behavior in lab studies, and compare.

The approach used in this paper is closest to the curious browser developed by Fox et al. [12], an app which was deployed to hundreds of users. The app logs search behavior such as queries and clicks. It also regularly asks users in-situ for session-level satisfaction labels, as well as result-level labels, for both clicked results and a sample of skipped results. For predicting result-level satisfaction the most important implicit variables were dwell time, position of click and end action (back to SERP, close browser, timeout, address bar navigation, new query). For predicting session-level judgments the most important variables were the result-level judgments, the number of results visited and end action.

Since result-level judgments are not known in practice, removing them made average dwell time, number of results and end action the

Table 1: Different forms of evaluation and their characteristics. Q-D indicates a relevance label on a query-document pair. SAT indicates a satisfaction label at the overall session or task level.

	Real users	In-situ labels	3rd party labels	Metric Type
TREC ad hoc [19]	No	No	Q-D	Average precision
Hassan et al. [13]	Yes	No	Q-D, SAT	SAT predict via behavior
Interleaving [8]	Yes	No	No	Ranker preference via behavior
TREC interactive [11]	In-lab	Yes	No	SAT
Curious browser [12]	Yes	Q-D, SAT	No	SAT predict via behavior

most important signals. Judgments were on three levels: Satisfied, partially satisfied, and dissatisfied. Grouping together satisfied and partially satisfied into one 'positive' label, users were positive on 78% of result-level labels and 77% of session-level labels. Since Fox et al. studied web search rather than email search, the curious browser did not have access to the rich set of user signals studied in this paper such as reply, move, forward and delete.

Email management and retrieval. Email is method of exchanging digital messages online, where messages accumulate over time unless deleted. Besides its use for communication, email is also used for task and time management [16] as well as for personal archiving [21]. Email management strategies have been identified [21] according to the number of folders used and the frequency of their use. *No filers* do not use folders except to archive or delete messages every few months. *Frequent filers* use folders, making daily passes of move and delete actions. *Spring cleaners* do the same but every 1-3 months rather than daily. Cecchinato et al. [7] found that the same person can use different strategies for their personal and work accounts.

Email search may occur under any of the identified email management strategies. Narang et al. [18] performed a large-scale analysis of email search logs, finding that people who organize their email less tend to search more. A related study [1] found that people perform a variety of activities during search, including organization actions such as move and delete. That study used both search logs and a survey, but it was not an in-situ survey paired with the log data, so it was not possible to analyze whether users who organized their email during search would report that their search was successful. Such analysis is possible here (see Table 1).

Carmel et al. [6] used both logs and labeling to evaluate email ranking. The log-based evaluation was a retrospective experiment, considering the rank position of a message selected by the user and how that would change under offline reranking. The relevance labeling evaluation was carried out by judges searching their own email, who each invented 25 queries based on a pre-identifed set of common query patterns. In both cases, a machine learned ranker could outperform the default date-sorted list of email search results.

The ranking approach could be further validated using a controlled trial online, assigning users to different rankers at random and seeing which group has activities that look most successful according to one of our models such as MarkovChain(1). Unlike the log-based evaluation in [6], users would see the ranked list, so the test would take into account whether date sorting makes a results list easier and faster to scan. Unlike the relevance labeling evaluation, the test would involve real users with real information needs, in-situ. It would be based on behavior rather than gathering new labels.

3 DATA COLLECTION

In this section, we described the data collection process. We first describe the in-situ survey we ran for client-side data collection, followed by the description of server-side activity logs and the merging of client-side and server-side data.

3.1 In-situ Survey Methodology

We implemented an in-situ survey application to collect the outcome of each email search session. Once a user installs the app on their PC, the app monitors search activity in the user's email client. When the end of a search session is detected, the app pops up an interface for a short survey about the success and ease of the search session. We used the following two conditions to detect the end of a search session:

- Explicit ending: the user clears the query input control and results by navigating away from the search interface
- Implicit ending: the user abandons the email search interface, which we defined as 3 minutes of inactivity

These conditions are reasonable but not perfect. In both cases there is some chance that the user may reenter the search interface and start another query soon after. However, during the development of the survey app we found these conditions work well in most cases. Note that we chose 3 minutes threshold in implicit ending which is shorter than 10 minutes in sessionization threshold so that people can respond to survey while their memory is fresh.

Figure 1 shows the user interface for the survey app. The user first answers a question about the success of their search session which included the query (or queries) shown. If the user responded 'yes' to the first question, then they are asked to answer additional questions about effort and relevance of clicked messages. The first additional question is the ease of finding what they were looking for. The second question asks about the relevance of each email message that was clicked. Finally, there is an optional box for additional comments about the search.

Often an email search happens in the middle of a task, and responding to the survey may cause an unwanted interruption. In order to make the survey participation as unobtrusive as possible, we also allowed the user to skip the survey once (Skip), or the for the rest of the day (Skip & disable for the day). After the user responded to a survey or Skipped the survey, we did not show the survey for the next 30 minutes to avoid making two many requests.

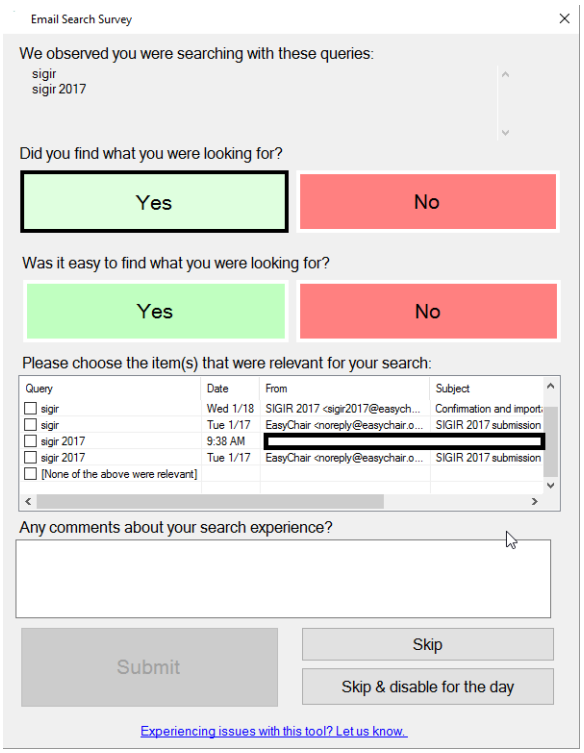

Figure 1: In-Situ Email search survey client interface. Some text has been removed for privacy reasons.

Once a survey is submitted, all the information *visible* in the survey pop-up window was sent to a remote database. This included the search queries, the subjects of clicked messages, and any textual comments they provided. For the search query and email reading activities, we also saved timestamps that were detected on the client. Participants were informed about the data being shared with researchers at the time of signing up for the study, and had the opportunity to opt out if they did not want to share this information. In addition, in the case of Skip, no information was sent to the server.

We ran the survey within our organization. The participants were recruited via internal mailing list, and we informed participants about the usage of the application and the data being shared with researchers. The participants were mostly IT professionals working in various locations in US and abroad. In order to keep the participation going, we also ran weekly raffle for a gift card throughout the duration of the study in which any active participant (i.e., those who kept the application running during the week) was eligible.

After 2 months, we had 1875 email search sessions collected from 65 users. Most people in our sample used the Microsoft Outlook email client application for their search, although some search sessions were from web email clients. Both the Outlook email application and the web email clients have standard email search capabilities, which includes: a search box at the top left corner of the interface, search results displayed in the middle of the screen and sorted by date by default, and the full-text email message in the right pane by default.

Table 2: Activities we captured in server-side log.

Activity	Remark
Send	Send a message
Move	Move a message
Reply	Reply to a message
Forward	Forward a message
Flag	Flag a message as a task
Mark	Mark a message as read/unread
Delete	Delete a message
Search	Issue a search query
ReadingLong	Read a message (dwell time >= 30 seconds)
ReadingShort	Read a message (dwell time < 30 seconds)
Create	Create a folder or message

Table 3: Summary of survey results for each outcome in overall (above) / after merging with activity log data (below).

	Failure	Success/Easy	Success/Hard
Survey Only	258(14%)	1534(82%)	83(4%)
Survey+Log	145(18%)	621(77%)	45(5%)

3.2 Activity Log Collection

In order to obtain a full view of users' search activities, we also collected the server-side log data from our organization's email server. For privacy reasons, the log data does not contain any textual information, yet it includes the activity user engaged in along with timestamp. The list of activities are shown in Table 2.

3.3 Data Processing

Our goal is to get a full picture of the users' email search including users' subjective impression along with activity records, therefore we merged the survey data with server-side activity logs. Since we had only the account owner information and timestamp for merging, we first aligned server-side (activity) and client-side (survey) data for each user by timestamp. We found that the server and client-side timestamp does not always align for various reasons, which resulted in some data loss.

Once we have the time-aligned merge data, we then split the data by session. We used the ten minutes of inactivity to sessionize the data, following the convention from previous studies [1]. Once we sessionized the data, the boundary for each search activity is detected within each session. We used the first search activity as the beginning of the session, and the user's submission of survey as the end of the session.

Table 3 summarizes the statistics of data before and after merging with the activity log. Note that the data decreased by 57% after the merging. We use the *Survey Only* data for the analyses in Section 4, and the *Survey+Log* data for predictive modeling in Section 5.

4 UNDERSTANDING SUCCESS IN EMAIL SEARCH

In this section, we analyze the in-situ survey data to better understand success and effort in email search. While the survey data does not have full activity logs, it has all the search and reading activities, along with timestamps. We first look at behavioral patterns associated with search session with different outcomes. We then look at the differences at the message-level.

4.1 Understanding User-level Success

The usage of email is personal in nature, and previous work [1] shows that there is wide variety in how people use email search. Some people seldom use search functionality, instead relying on various categorization mechanism provided by email clients or scanning the inbox, while other use search much more frequenty. A recent study[18] found this complementary nature of search and organization strategies for email. Among the people who tend to use search for email access, there may still be a wide range of difference in information needs and search strategies.

Our survey data provides a way to understand these individual differences in email search behavior and how they affect success in email search. We first compute six measures of search activity for each user and the look at the distribution across users. Figure 2 shows the distribution of six per-user session measures as box plots, where each plot presents the distribution of statistics aggregated for each user.

The first two plots show that the median number of queries per session is slightly less than 2 and the median number of read messages is about 3. The third plot shows that proportion of viewed messages that were relevant was slightly less than 50%. The fourth plot shows that the proportion of queries with advanced operators (e.g., from:) was approximately 20%, but the variance across users is large. The fifth plot shows that search success is high (median 90%) and there is little variance across users. The final plot shows the number of judgments submitted per user.

Email clients provide various search operators as shown in Table 4. Most query operators correspond to different email fields and are therefore self-explanatory, except for the 'Conversation:' operator which provides a way to find the messages in the same thread. This operator is automatically invoked in the interface by selecting a message and then a menu item called 'Find Related Messages in this Conversation'. How much are these operators used in practice? Figure 2 shows that 20% of queries had an operator in median, although the variability was quite high.

A recently study of email search log [1] found similar trends, showing that 18% of queries had the operator, and that the percentage of the *from:* operator is 75% and the *to:* operator is 13%, respectively.

We now consider the extent to which these per-user session measures are correlated. Table 5 presents the Pearson correlation coefficient across different user-level measures. The correlation is generally low, except for that between message count and query count, which is expected since if a person queries a lot they have more opportunity to view messages. There is an interesting correlation (0.25) between the usage of query operators and success rate, which seems to suggest that users who use query operator more are also more successful in their searches.

Table 4: Summary of query operators.

Query Operator	Count	Percentage
from:	704	72%
to:	123	13%
Conversation:	119	12%
subject:	14	1%
contents:	5	1%
hasattachments:	5	1%

4.2 Understanding Session-level Success

Table 6 summarizes the measures of session-level behavior. The table is grouped by sessions with different outcomes – success with low effort (*Success/Easy*), success with high effort (*Success/Hard*), and failure (*Failure*).

The results highlight the differences among sessions with different outcome. As for query counts, we find that *Success/Easy* sessions have the lowest number of queries. Interestingly, *Success/Hard* sessions had higher message count than *Failure* sessions. It is possible that the type of search tasks in *Success/Hard* sessions are very important, so users were willing to spend extra effort to complete the search compared to *Failure* sessions where they gave up.

Another measure of user effort is query length, which is also shown in the Table 6. Note that we removed all queries with operators from this calculation, since the query operators can be chosen from the menu instead of manually typing, thereby skewing the length measure. We can see that failure sessions have longer queries on average compared to successful sessions.

As we saw in the previous section, the usage rate of query operators is correlated with user-level success. The same trend holds true at session-level. From Table 6, we can see that the proportion of queries that contained query operators was highest in *Success/Easy* sessions in mean and median, followed by *Success/Hard* sessions.

Another angle to session-level behavior is the characteristics of messages read. In Table 6, the average number of messages read during a session is by far highest in *Success/Hard* sessions. *Failure* sessions have higher message count than *Success/Easy* sessions, yet the difference is small. This trend is consistent with those from the query counts.

We also looked at the ratio of messaged marked relevant by participants, and the ratio was 60% in *Success/Easy* sessions, which is more than twice as high as *Success/Hard* sessions. Since we did not ask people to judge the relevance of messages when they failed to find what they were looking for, this value is not available for *Failure* sessions.

Among the properties of messages read, message age is of particular interest. Message age is defined by the interval between when the email was sent or received and the time of search. The overall age of messages read during a search session could be affected by various factors.

Table 6 shows the comparison of median age for all messages and relevant messages across sessions with different outcomes. We

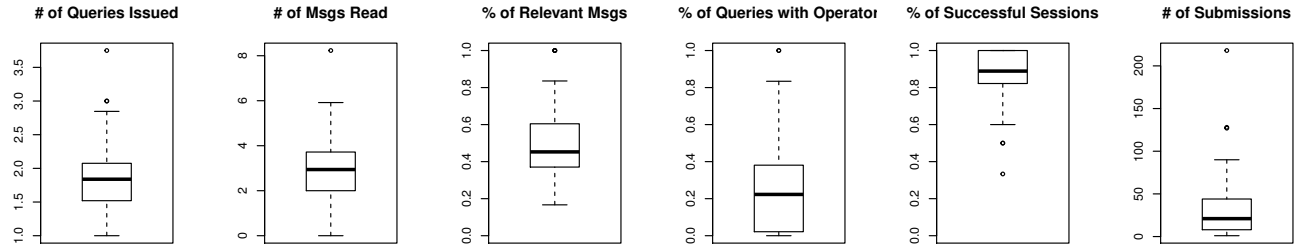

Figure 2: Distribution of per-user session statistics.

Table 5: Correlation among per-user session statistics. Statistics above 0.2 are boldfaced.

	AvgMsgCount	AvgQueryCount	SurveyCount	QueryOperator%
AvgQryCount	**0.60**			
SurveyCount	-0.11	-0.08		
QueryOperator%	0.13	**0.21**	0.19	
Success%	0.05	-0.03	-0.03	**0.25**

Table 6: Statistics of session-level behavior for sessions with different outcomes.

Session Outcome	Query Count	Query Length	QueryOperator %	MsgRead Count	RelevantMsg %	MedMsgAge (in Days)	MedRelevantMsg Age (in Days)	Survey Count
Success/Easy	1.71	1.40	32%	2.68	60%	3	3	1534
Success/Hard	2.87	1.47	21%	5.47	27%	12	10	83
Failure	2.20	1.59	17%	2.93	0%	9.5	NA	257

present the median instead of mean because the distribution of message age is heavily skewed toward right. Note that relevance labels are not available for successful sessions.

4.3 Understanding Message-level Success

In addition to session-level success labels, we also collected relevance labels for each message read. This allows us to understand success in email search at individual message level, which in turn can be valuable in collecting labels for training relevance-based ranking methods [6]. Here we delve into this, focusing on dwell time at each message, which is an important signal in understanding the users' engagement with individual messages [15].

Note that dwell time is reported for only non-last messages within each session because we do not have a reliable way to detect reading time for the last message is a session. Also, if a message is the last one read during a session, it is an important signal for relevance in and of itself. Table 7 confirms this intuition, where the ratio of relevant messages among last-read messages was 54%, compared to 24% for non-last read messages.

If being the last message read indicates relevance, what would the dwell time for non-last messages tell us about its relevance? Table 8 shows that the average dwell time for relevant messages is about 37 seconds, whereas the value is 22 seconds for non-relevant messages. Figure 3 shows the distribution of dwell time in the box plots, where the median dwell time for relevant vs. non-relevant messages are 12 and 6 seconds, respectively.

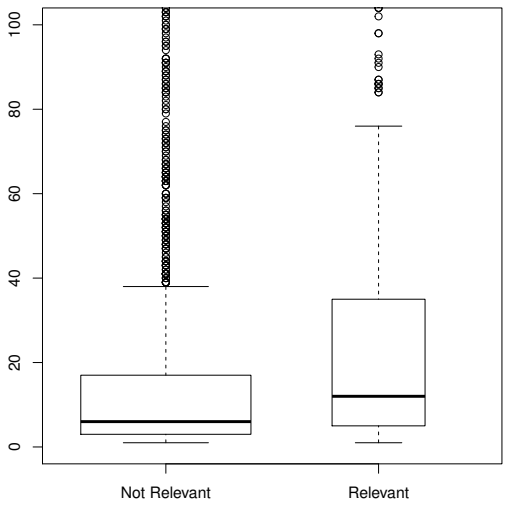

Figure 3: Comparison of dwell time among messages with different relevance labels.

Table 7: Cross-tabulation of the relevance label for each message read and whether the reading was the last activity.

Message Relevance	Last Activity	Non-Last Activity
Not Relevant	681	2468
Relevant	793	737
$P(Relevant)$	54%	24%

Table 8: Average dwell time for relevant vs. non-relevant reading activities.

MsgRelevance	DwellTime(mean)	SampleSize
Not Relevant	22.38	2468
Relevant	37.23	737

5 MODELING SUCCESS IN EMAIL SEARCH

In the previous section, we looked at the characteristics of success in email search at the user, session and message levels. While these are informative in and of themselves, it would be even more valuable if we could build a model that could predict the success of an email search. Such model would further contribute to our understanding of success in email search, and could be used to evaluate a new search method against baseline.

There are many approaches to building models for predicting search success, and here we provide the rationale for approaches taken in this work. First, one can model success at session-level vs. message-level, and we decided to model session-level success because that's the level at which we have cleanest labels and features. Second, the modeling choice can focus on multiple aspects, including prediction accuracy, interpretability and efficiency. We focused on interpretability in this work because the goal was to build a transparent model which provides insights about its outcome upon inspection.

5.1 Experimental Design

Here we describe the details of experiments we ran. We used the merged data set of the client-side survey and the server-side activity log as described in Section 3.3. The data contained the list of activities that occurred during each search session, along with various session-level properties and labels from the survey. Since the data was collected over 2 months, we used the data from the first month for training models and the data from the second month for evaluation. Using random sampling or k-fold cross validation would mean the that the future might be used to predict the past.

The evaluation setup is as follows: each model takes in various session-level features and calculates a score for session-level success. The score for each model is then compared against the ground truth session-level labels we obtained from the survey. This is a binary classification task which predicts the success vs. failure of a given session. Since reading time has previously been shown to be an important feature in predicting search success [13], we built every model with or without reading time.

Now we describe the evaluation metric and models we used. As a metric we used AUC (area under curve) of the ROC (receiver operating characteristic) curve [4]. There are a few reasons for this choice: Since we employed different modeling approaches, the scores from each model are not necessarily comparable. Also, there is a skew in the distribution of target labels, which the ROC curve is known to be robust tp.

Table 9: Markov Chain models and activity examples.

	MarkovChain (0)	MarkovChain (1)
Activities	$P(Query_t)$ $P(Reading_t)$	$P(Reading_{t-1}, Query_t)$
Activities + Reading Time	$P(Query_t)$ $P(ReadingShort_t)$ $P(ReadingLong_t)$	$P(ReadingShort_{t-1}, Query_t)$ $P(ReadingLong_{t-1}, Query_t)$ ...

5.2 Models Compared

Here we compare a few approaches for predicting the success of email search. Each model combines different subsets of input features. Our first model is a heuristic baseline, which associates a subset of activities with session-level success, and outputs a positive label when any of these activities are found during a search session. The activities we considered as success for this baseline model includes 'Move','Reply', 'Forward', 'Flag', 'ReadingLong' (for the version that incorporates reading time).

While this simple approach takes into account the activities performed during a search session in order to make a prediction, it uses heuristics in assessing the contribution of each activity toward success. A more data-driven approach is to use the survey data we collected to determine the weights. Since we have a sequence of activities along with success label for each search session, a natural choice is a discrete Markov Chain of user behavior.

We experimented with Markov Chain models based on different Markov properties, where the zero-order Markov Chain assumes the independence among activities, and the first-order Markov Chain assumes that the present activity is dependent on the previous activity. The first-order Markov Model can potentially model the context in which an activity is triggered more carefully, although the expanded state space may result in sparsity in estimation.

The Markov Chain models we experimented with can be further distinguished based on whether reading time is considered or not. Previous work on modeling search success [13] notes that the time spent on reading is an important indicator of success, and here we decided to incorporate time by splitting the reading activities into short reading vs. long reading, where the short reading is defined by the reading activity with duration less than 30 seconds.

Table 9 shows the example activities for four varieties of Markov Chain model we experimented with. Note that the state space is defined by individual activity for the zero-order Markov Chain models (*Markov Model (0)*), whereas the state space is defined by activity pairs for the first-order Markov Chain models (*Markov Model (1)*). Also, the *Reading* activity is divided into *ReadingLong* and *ReadingShort* for Markov Chain models with reading time.

Now we focus on how we derive success score from models. Given the state space in Markov Chain as above, we can estimate two models (M_s and M_f) corresponding to successful sessions and failure sessions, respectively. We can calculate the chance of success

for a new model using these two models. The detailed calculation varies depending on the type of Markov Chain used. $P_{M0}(S_i)$ represent the probability of observing the ith state in the sample space of activities for zero-order Markov Chain $M0$. $P_{M1}(S_i, S_{i-1})$ represent the probability of observing the transition from $i-1$th state to ith state in the sample space of activities for first-order Markov Chain $M1$.

$$L_{M0}(S) = \prod_{i=1}^{n} P_M(S_i) \tag{1}$$

$$L_{M1}(S) = \prod_{i=2}^{n} P_M(S_i|S_{i-1}) \tag{2}$$

Given a Markov Chain model M, we can calculate the likelihood of observing a sequence of activities for a new session $S = (S_1, S_2, ..., S_n)$ by multiplying the probability for each action pairs. Using the the likelihood values from model of success P_s and failure P_f, we can calculate the odds ratio of the likelihood from success model L_s over L_f. Finally, we take the log of the odds ratio to avoid numerical underflow. The equation below shows the overall calculation of log odds for zero-order Markov Chain (above) and first-order Markov Chain (below).

$$LogOdds_{M0}(S) = \sum_{i=1}^{n} log(P_s(S_i)) - \sum_{i=1}^{n} log(P_f(S_i)) \tag{3}$$

$$LogOdds_{M1}(S) = \sum_{i=2}^{n} log(P_s(S_i|S_{i-1})) - \sum_{i=2}^{n} log(P_f(S_i|S_{i-1})) \tag{4}$$

The Markov Chain approach described above can be considered as a generative model in that the scoring is based on the likelihood of a sequence generated from probabilistic models. Alternatively, one can think of a discriminative approach where the goal is to find a model which directly optimize for the classification performance.

We use RandomForest [5] for this reason, which is a discriminative model known to perform well out-of-the-box. We experiment with two models with Markov Chain features – activity counts from zero-order Markov Chain (RandomForest(0)), activity counts from first-order Markov Chain (RandomForest(1)). These are models based on the same set of features as Markov Chain models, except they they are fed into RandomForest model.

The third RandomForest model we tested employed session-level features along with activity counts from first-order Markov Chain (RandomForest(1+S)). Session-level features we used include the count of queries, count of messages, ratio of operators in session queries and average message age. Note that the Markov Chain model does not accommodate these features since they do not have a probabilistic interpretation.

The models we tested have distinctive characteristics. The baseline model has the simplest calculation procedure (i.e., whether the given set of activities exist in the new session or not), and does not require any training data in the form of activities and labels.

Markov Chain models have high interpretability in that the scores from individual activity contribute linearly to the final score. This means that we can also calculate the odds ratio at the level of individual activities, as we show in Section 5.4.

Table 10: Comparison of Area under ROC (AUC) values across models we evaluated.

	Activities	Activities + Reading Time
Baseline	0.602	0.659
MarkovChain(0)	0.659	0.693
MarkovChain(1)	0.719	0.753
RandomForest(0)	0.623	0.655
RandomForest(1)	0.681	0.709
RandomForest(1+S)	0.739	0.769

In contrast, the RandomForest models employ a large collection of decisions trees to produce a prediction, which makes the results harder to interpret. It has the advantage of being able to incorporate arbitrary features, as well as the ability to model complex and possibly non-linear interaction among features.

5.3 Experimental Results

Here we present the outcome of prediction for session-level satisfaction. Table 10 summarizes the Area under ROC values for the models we evaluated. Note that the results are grouped into 1) baseline 2) variants of Markov Chain models 3) variants of RandomForest, with the columns showing the results with or without taking into account the reading time.

The first overall trend from Table 10 is that both Markov Chain and RandomForest models are much better than the baseline and that the Markov Chain model is slightly better than the RandomForest model with the same feature set. Since RandomForest is a more powerful model, it may outperform Markov Model given enough training data.

In terms of feature sets used, it is also clear the first-order models that use activity pairs are always better than the zero-order models that consider each activity independently. In addition, it is clear that adding reading time provides a further boost in AUC in all cases. Finally, adding session-level features to RandomForest(1+S) model improves the performance as well.

Now, let's look at ROC curve for each model to interpret the results in more detail. An ROC curve characterizes the performance of a classifier by plotting Sensitivity (true positive rate) against Specificity (1 - false positive rate). We first compare the Markov Chain models against the baseline model. Figure 4 presents the ROC curve for two Markov Chain models, along with a baseline model. It shows that the use of activity pairs improves the sensitivity in all but the leftmost side of the plot (high specificity / low false positive rate region). This shows that the use of activity pairs provides additional evidence when the signal from individual activity is weak.

Next, we compare RandomForest models against the baseline model. Figure 5 presents the ROC curve for three RandomForest models, along with the baseline model. It shows that the use of activity pairs improves the true positive rate in all but the leftmost side of the plot (high specificity / low false positive rate region), as we saw in Markov Chain model. Also, the inclusion of session-level features improves the performance across the board.

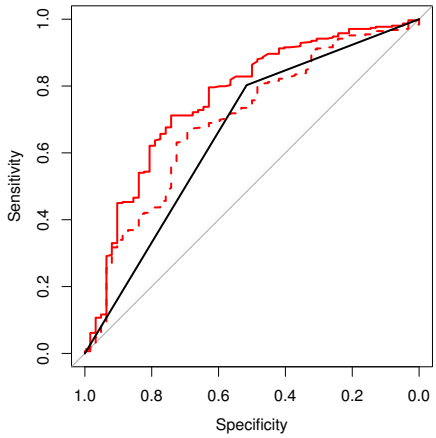

Figure 4: ROC curve for MarkovChain(1) (red solid line) vs. MarkovChain(0) (red dashed line) with reading time. Baseline model with reading time is shown in black solid line.

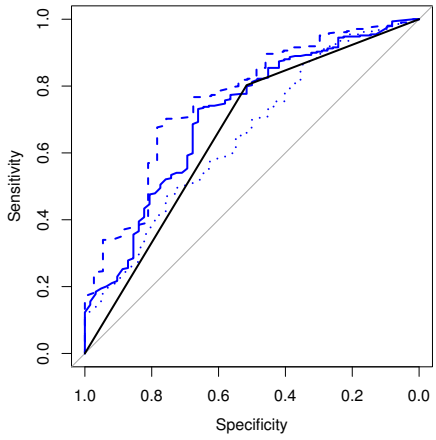

Figure 5: ROC curve for RandomForest(0) (blue dotted line) vs. RandomForest(1) (blue solid line) vs. RandomForest(1+S) (blue dashed line) models with reading time. Baseline model with reading time is shown in black solid line.

Finally, Figure 6 compares the ROC curve for MarkovChain(1) against RandomForest(1) model, along with the baseline. The Markov Chain model outperforms the RandomForest model in all but the leftmost side of the figure. This shows that the discriminative model is better at getting the first few instances right (i.e., high sensitivity at high specificity region) yet worse in overall.

5.4 Further Analysis

Here we present a few additional analyses to provide further insight into the results we presented in Section 5.3. These analyses are based on Markov Chain models, which provides a simple interpretation of session-level results in terms of individual activities.

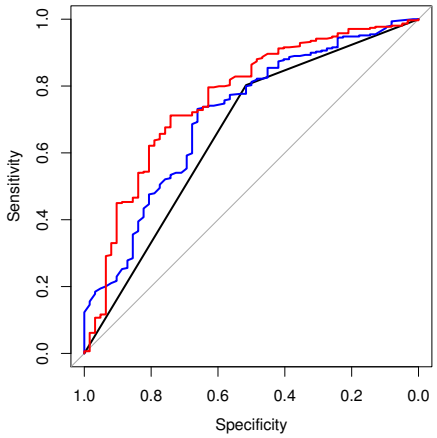

Figure 6: ROC curve for MarkovChain(1) (red solid line) against RandomForest(1) (blue solid line) model. Baseline model with reading time is shown in black solid line.

Table 11: Odds Ratio for Activities ($\frac{P_s(a)}{P_f(a)}$).

Activity	Odds Ratio	Sample Size
Send	2.87	324
Reply	1.99	194
Move	1.93	402
Forward	1.62	38
Delete	1.25	366
Mark	1.21	832
ReadingLong	1.13	1256
Flag	1.02	19
Search	0.94	3199
ReadingShort	0.88	5930
Create	0.24	75

Table 11 presents the odds ratio for each activity. The odds ratio is calculated as the probability of observing an activity in successful sessions divided by the probability in failure sessions. Note that top part of the table is dominated by typical message-level actions, which would be done for relevant message(s). Since *Search* and *ReadingShort* are two frequent activities with odds ratio less than one (and negative log odds ratio), any session with long sequences of search and short reading activities will be regarded as a failure.

Table 12 presents the odds ratio for each activity pair. The odds ratio is calculated in the same way as above, except with activity pairs instead of individual activities. Interestingly, we can see *ReadingShort* activity both at the top and bottom of the table, by which we can see that a short reading is positive when followed by reply or send activity, yet negative when it is found around search, delete, or another short reading activity. This result exemplifies the power of evaluating an activity in the context of other activities.

Table 12: Odds Ratio for Activity Pairs ($\frac{P_s(a_t|a_{t-1})}{P_f(a_t|a_{t-1})}$). Only top 5 and bottom 5 activity pairs are shown due to space limitations.

Activity Pair	Odds Ratio	Sample Size
Send → Reply	7.22	50
Move → Move	4.51	53
ReadingShort → Reply	3.38	89
ReadingShort → Send	3.09	160
Search → ReadingLong	2.51	170
...	...	...
Search → ReadingShort	0.82	888
Delete → ReadingShort	0.80	147
ReadingShort → ReadingShort	0.74	3540
ReadingShort → Search	0.58	350
Search → END	0.30	55

6 CONCLUSIONS

Unlike some other information retrieval applications such as web search where it is common to build a test collection using third party judges, email search occurs over a private corpus, so it is useful to develop new methods of analysis and evaluation. Here we employed an in-situ survey that is tightly coupled with user activity logs, to get both explicit and implicit data on search success for the same individuals.

This allows us to analyze general characteristics of email search success, which could be reused as a comparison point in other studies of email. In the survey, 20% of queries had an operator, with the most common operator being *from:*. Use of search operators was associated with a higher success rate, although we have no evidence of a causal link. Easy successful sessions were also associated with finding more recent messages, with fewer shorter queries, and fewer opened messages. High-effort successful sessions, while rare, had more queries and opened messages, and the opened messages were older, suggesting the need for better information retrieval support for such cases. Failed sessions were in between, perhaps indicating that the user gave up because the search was not all that important or they used other means for finding the relevant message.

At the message level, having a higher dwell time and being the last-selected message were both good success indicators, agreeing with findings from web search [12]. After joining with activity logs, a wide variety of activity types were found to be associated with session success. Specifically the following actions were all associated with search success: Reply, Move, Forward, Delete, Mark, ReadingLong and Flag. Short dwell selection (ReadingShort) was the only message-level negative indicator for overall success. Considering actions not tied to a new search result, sending a message was the most positive, and Search and Create were negative. The negative events may indicate that the user is still trying to find the information they need, via another search or by creating an email or meeting request, perhaps to ask someone else for some information that they failed to retrieve. These event-level insights could be used directly in future studies, rewarding and penalizing activities observed during an email search experiment. The insights could also be compared against findings from other email search interfaces and analysis methods.

Once the predictive models of search success have been learned, they can be applied to new log data without any new survey data. For example, MarkovChain(0) can be applied using only the data in Table 11 in equation (1). Future work could consider how to automatically optimize an email search system to yield these positive events and avoid the negative events. One approach would be to treat our predictive model as a utility function, then use a factored design as in [20].

As a future work, the proposed approach can be extended in many ways. More complex learning models can potentially improve the quality of prediction. New search experiences such as voice can be evaluated using this hybrid approach of label and log data.

REFERENCES

[1] Qingyao Ai, Susan T. Dumais, Nick Craswell, and Dan Liebling. 2017. Characterizing Email Search using Large-scale Behavioral Logs and Surveys. In *Proceedings of WWW'2017*. ACM.

[2] Ahmed Hassan Awadallah. 2012. A semi-supervised approach to modeling web search satisfaction. In *Proceedings of SIGIR 2012*. ACM.

[3] Ahmed Hassan Awadallah, Yang Song, and Li-wei He. 2011. A task level metric for measuring web search satisfaction and its application on improving relevance estimation. In *Proceedings CIKM 2011*. 125–134. DOI: http://dx.doi.org/10.1145/2063576.2063599

[4] Andrew P. Bradley. 1997. The Use of the Area Under the ROC Curve in the Evaluation of Machine Learning Algorithms. *Pattern Recogn.* 30, 7 (July 1997), 1145–1159. DOI: http://dx.doi.org/10.1016/S0031-3203(96)00142-2

[5] Leo Breiman. 2001. Random Forests. *Mach. Learn.* 45, 1 (Oct. 2001), 5–32. DOI: http://dx.doi.org/10.1023/A:1010933404324

[6] David Carmel, Guy Halawi, Liane Lewin-Eytan, Yoelle Maarek, and Ariel Raviv. 2015. Rank by Time or by Relevance?: Revisiting Email Search. In *Proceedings of the 24th CIKM*. ACM, 283–292.

[7] Marta E Cecchinato, Abigail Sellen, Milad Shokouhi, and Gavin Smyth. 2016. Finding email in a multi-account, multi-device world. In *CHI Conference on Human Factors in Computing Systems*. ACM, 1200–1210.

[8] Olivier Chapelle, Thorsten Joachims, Filip Radlinski, and Yisong Yue. 2012. Large-scale validation and analysis of interleaved search evaluation. *ACM Transactions on Information Systems (TOIS)* 30, 1 (2012), 6.

[9] Gordon V Cormack and Thomas R Lynam. 2005. Spam Corpus Creation for TREC.. In *CEAS*.

[10] Susan Dumais, Edward Cutrell, Jonathan J Cadiz, Gavin Jancke, Raman Sarin, and Daniel C Robbins. 2003. Stuff I've Seen: a system for personal information retrieval and re-use. In *Proceedings of the 26th SIGIR*. ACM, 72–79.

[11] Susan T Dumais and Nicholas J Belkin. 2005. The TREC interactive tracks: Putting the user into search. *TREC: Experiment and evaluation in information retrieval* (2005), 123–153.

[12] Steve Fox, Kuldeep Karnawat, Mark Mydland, Susan Dumais, and Thomas White. 2005. Evaluating implicit measures to improve web search. *ACM Transactions on Information Systems (TOIS)* 23, 2 (2005), 147–168.

[13] Ahmed Hassan, Rosie Jones, and Kristina Lisa Klinkner. 2010. Beyond DCG: user behavior as a predictor of a successful search. In *Proceedings of the WSDM 2010*. ACM, 221–230.

[14] Jiepu Jiang, Ahmed Hassan Awadallah, Xiaolin Shi, and Ryen W White. 2015. Understanding and predicting graded search satisfaction. In *Proceedings of the WSDM 2015*. ACM, 57–66.

[15] Youngho Kim, Ahmed Hassan, Ryen W. White, and Imed Zitouni. 2014. Modeling Dwell Time to Predict Click-level Satisfaction. In *Proceedings of the WSDM 2014 (WSDM '14)*. ACM, New York, NY, USA, 193–202. DOI: http://dx.doi.org/10.1145/2556195.2556220

[16] Wendy E Mackay. 1988. Diversity in the use of electronic mail: A preliminary inquiry. *ACM TOIS* 6, 4 (1988), 380–397.

[17] Dmitry Lagun Mikhail Ageev, Qi Guo and Eugene Agichtein.

[18] Kanika Narang, Susan T. Dumais, Nick Craswell, and Dan Liebling. 2017. Analysis of Email Search and Org Strategies. In *Proceedings of CHIIR'2017*. ACM.

[19] Ellen M Voorhees and Donna K Harman. 2005. *TREC: Experiment and evaluation in information retrieval*. Vol. 1. MIT press Cambridge.

[20] Yue Wang, Dawei Yin, Luo Jie, Pengyuan Wang, Makoto Yamada, Yi Chang, and Qiaozhu Mei. 2016. Beyond ranking: Optimizing whole-page presentation. In *WSDM*. ACM, 103–112.

[21] Steve Whittaker and Candace Sidner. 1996. Email overload: exploring personal information management of email. In *Proceedings of the SIGCHI*. ACM, 276–283.