InLook: Revisiting Email Search Experience

Pranav Ramarao, Suresh Iyengar, Pushkar Chitnis[*], Raghavendra Udupa, B.Ashok
Microsoft Research, India
Bangalore, India
{t-prrama,supartha,pushkarc,raghavu,bash}@microsoft.com

ABSTRACT

Emails continue to remain the most important and widely used mode of online communication despite having its origins in the middle of last century and being threatened by a variety of online communication innovations. While several studies have predicted the continuous growth of volume of email communication, there is little innovation on improving the search in emails, an imperative part of the user experience. In this work, we present a lightweight email application codenamed **InLook**, that intends to provide a productive search experience.

Keywords

Email, search, productivity

1. INTRODUCTION

Email is the oldest modes of online communication and is used by more than 2.5 billion users today. Studies have predicted its continuous growth over the next few years [1]. Email **search** is an integral part of the user's experience. Studies also report that users search their mail data frequently and spend significant amount of time searching and re-searching [15].

Despite being an important activity, Email Search has not seen significant innovation in comparison to, say, Web Search. Web Search has evolved over the past decade from ten blue links to a more rich and task-oriented experience. Though email search has similar objectives to Web Search, there are some important distinctions:

- *Personalized*: Email Search is more personalized in nature as compared to web search.
- *Strong Intent*: As the document collection against which the user is querying is the user's own mail data, search queries generally have a strong intent.
- *Privacy Concerns*: As mail data is mostly private, so are queries that users issue against that data. Users are generally reluctant to share any data that is needed to build richer search experiences.
- *Absence of query logs*: One major impediment to email search is the absence of large volumes of query logs.

In this work, we present InLook, a light-weight email search tool, which is an amalgamation of different bodies of research related to email search. The underlying objective is to enhance the search experience of Email users.

The rest of the paper is structured as follows: Section 2 summarizes the various related work in the area of email search. Sections 3.1 through 3.3 show the varied features in InLook search experience. Section 4 describes the ranking algorithm used in InLook. We describe InLook application design in Section 5. Section 6 reports preliminary results of our user study of InLook. Section 7 briefly compares InLook with other email applications.

2. RELATED WORK

Email search can be tedious and frustrating experience, especially for older mails [13]. Researchers have tried various approaches to alleviate the problems related to email search. These can be broadly bucketed into following categories:

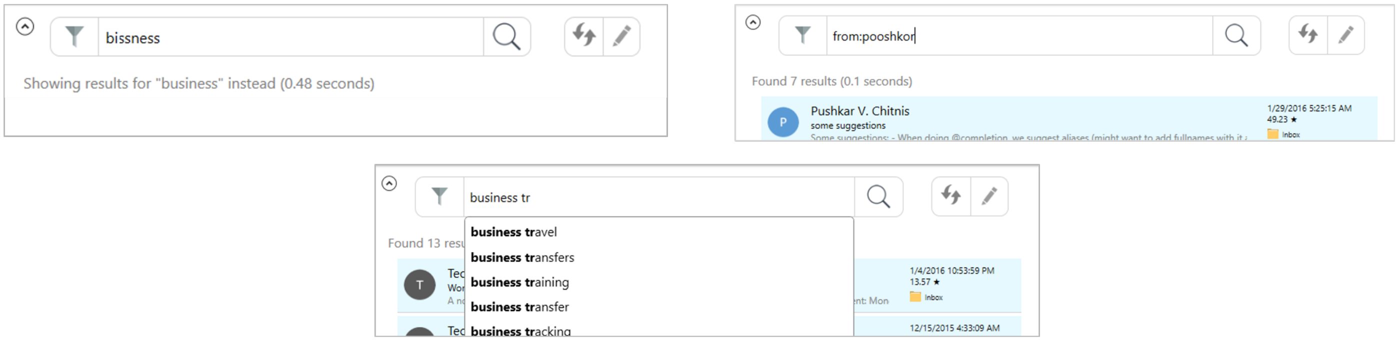
Organizing emails: Contrary to the popular belief, organizing emails into multiple folders might offer little help in search [20]. An alternative approach tries to organize emails into fewer categories like Social, Travel and Promotions [14] [3].

Email actions: Aberdeen *et al* [4] [8] predict the importance of an email based on the probability that it will be acted upon. A recent work by Dotan Castro *et al* [12] tries to predict actions on emails based on past user behavior. The actions considered in this work are: read, reply, delete and delete-without-read.

Order by relevance: To the best of our knowledge, there are only a handful of published research works in the area of email relevance. AbdelRahman *et al* [7] employ different features like email content, subject and sender to assign a score to an email. Ogilvie and Callan [18] use a language model based approach to rank emails. A recent work by Carmel *et al* [11] discuss the area of email relevance in great detail. The authors employ a wide range of features to rank emails, in a

[*]Currently working at Microsoft Research, Redmond, WA, USA

SIGIR '16, July 17-21, 2016, Pisa, Italy
© 2016 ACM. ISBN 978-1-4503-4069-4/16/07...$15.00
DOI: http://dx.doi.org/10.1145/2911451.2911458



bissness

Showing results for "business" instead (0.48 seconds)

from:pooshkor

Found 7 results (0.1 seconds)

Pushkar V. Chitnis
some suggestions
Some suggestions: - When doing @completion, we suggest aliases (might want to add fullnames with it
1/29/2016 5:25:15 AM
49.23 ★
Inbox

business tr

Found 13 resu

business travel
business transfers
business training
business transfer
business tracking

1/4/2016 10:53:59 PM
13.57 ★
Inbox

12/15/2015 4:33:09 AM

Figure 1: Guided Search: Spell-Correct, Fuzzy person search, Auto-complete

learning to rank framework. The features include text similarity, folder information, attachments and sender behavior. Popular email applications like Google Inbox [4] and Thunderbird [6] display search results by relevance. But, these offerings have not divulged any related algorithmic details.

Other research areas in the context of emails search include generating suggestions [9] and spell-corrections [10] in the absence of query logs.

3. INLOOK SEARCH EXPERIENCE

This section details the email search experience offered by InLook. Some of the ideas below have been tried out before; InLook stitches them together in one application.

3.1 Guided search in InLook

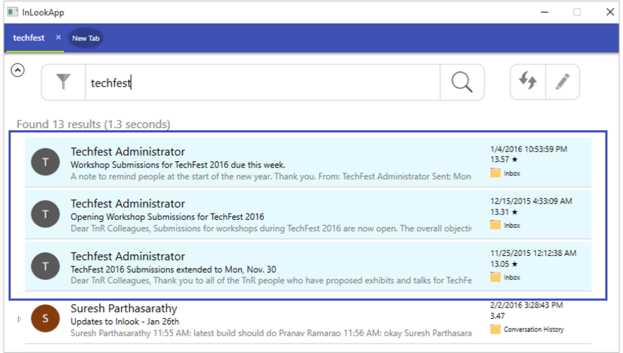
One of the main contributors to the success of Web Search is guided search. In InLook we provide guided search for email users by means of the following search features:

1. *Auto-Completion and Spelling correction*: We employ the algorithm similar to the one described in [10] for auto-completion and spelling correction of both contacts and general search queries. Some of the features used in this algorithm to the rank the candidates are:
   - SubjectMatch: The maximum of fraction of candidate tokens appearing in any single subject line.
   - ContactMatch : The fraction of tokens in candidate which are present in at least one user contact.
   - Language Model Feature: Bigram language model score of the candidate. The raw bigram counts are discounted using Kneser-Ney technique.
2. *Fuzzy Search of People names*: We employ hashing-based people search algorithms developed in [17][19]. This approach learns hash functions that map similar names to similar binary code words in a language independent space. For example, "Chifa" ∼ "Qifa", "Lakshmi" ∼ "Laxmi", "Posansky" ∼ "Poznansky".

Figure 1 shows guided search instances in InLook.

3.2 InLook intent pane

In InLook, we present a mix of relevance and recency results for search queries. Showing results with either of the sort orders solely might not suffice for all queries. InLook introduces a notion of "Intent pane" which displays the best three emails based on relevance for a particular query. Rest of the mails in the result are shown in a traditional date order, we call this as the "Recency pane". Intent pane can be thought of as analogous to an ”entity card” in Web Search. Section 4 describes the ranking model used. Additionally, this pane is not restricted to email results. It can also show relevant attachments, contact cards or links. Figure 2 shows Intent pane in action in InLook.



techfest

Found 13 results (1.3 seconds)

Techfest Administrator
Workshop Submissions for TechFest 2016 due this week.
A note to remind people at the start of the new year. Thank you. From: TechFest Administrator Sent: Mon
1/4/2016 10:53:59 PM
13.57 ★
Inbox

Techfest Administrator
Opening Workshop Submissions for TechFest 2016
Dear TnR Colleagues, Submissions for workshops during TechFest 2016 are now open. The overall objecti
12/15/2015 4:33:09 AM
13.31 ★
Inbox

Techfest Administrator
TechFest 2016 Submissions extended to Mon, Nov. 30
Dear TnR Colleagues, Thank you to all of the TnR people who have proposed exhibits and talks for TechFe
11/25/2015 12:12:38 AM
13.05 ★
Inbox

Suresh Parthasarathy
Updates to Inlook - Jan 26th
Suresh Parthasarathy 11:55 AM: latest build should do Pranav Ramarao 11:56 AM: okay Suresh Parthasara
2/2/2016 3:28:43 PM
3.47
Conversation History

Figure 2: Intent Pane

3.3 Tabbed searching and commands

We take inspiration from web browsers and introduce the notion of tabs in InLook. This allows users to make multiple searches at the same time. We also provide a mechanism to pin a query to a tab. This becomes like an active channel or a standing query and the tab gets refreshed for this query periodically. Also, one can tear and dock tabs just like browsers and put them in multiple workspaces. This provides a very intuitive notion of segregating and searching through emails. Figure 3 shows InLook tabs in action. We plan to perform a detailed study on usage of tabs in the context of email search.

We introduce a notion of command box in InLook with an objective of faster task completion. The functionality of search box is over-ridden so that it morphs into a powerful command box. Users can think of this to be a shell on top of their email data. All these commands are powered by autocomplete to make it very easy to use. Some of the commands that can be executed are "*#compose* @person mailbody", "*#chat* @person" and "*#template* parameters".

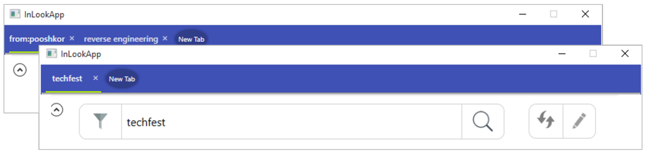


Figure 3: Tabbed searches

4. RANKING OF EMAILS

Emails in the intent pane are ordered by relevance. Collecting training data for training email ranking models is a difficult task. It is tedious for a user to mark relevant mails for each query. Instead, we rely on a user's click log to give us partial relevance judgments and use it for ranking [16]. Also, it is much more plausible to infer that a particular mail is better than the other, instead of an absolute judgement. The learning to rank algorithm we discuss below is similar to the ranking discussed in [11].

For instance, let's say a user clicks on mails 1, 3 and 5. Given that the mail snippets are sufficiently informative, this gives some indication of the user's preferences. In this case, we may infer that the email at position 5 is more relevant than mails at positions 2 and 4. Likewise, we may infer that mail at position 3 is more relevant than the mail at position 2. We call such pairs of mails as discordant pairs, represented as $mail_j <_r mail_i$, which implies $mail_i$ is more relevant to user than $mail_j$. The sequence of clicks doesn't impose any ordering on mails 1, 3 and 5. We aggregate click logs from all users and train our ranking model using this data. For a given user, the logs consist of set of queries and for each query, the click sequence of mails. We log only the feature values as the mails can contain confidential information. We use a model based on logistic regression for our training. The objective function looks like:

$$Obj = min\left(\lambda * \frac{|w|^2}{2} + \frac{1}{n} * \sum_{(i,j)} Loss(mail_i, mail_j)\right) \quad (1)$$

where n is the number of discordant mail pairs (i,j) and $mail_i$ and $mail_j$ are the corresponding mails. w represents the weight vector we want to learn. This loss equation consists of a regularizer and sums up the loss over all the discordant pairs. $Loss(mail_i, mail_j)$ can be written as

$$\frac{1}{1 + exp(-(s_i - s_j))} \quad (2)$$

The score for i^{th} mail s_i is $\sum w_i * f_i$ where f_i is the feature value and w_i is the weight we want to learn. The optimal weights for the above optimization problem can be estimated using an algorithm like LBFGS [5]. Table 1 lists some of the features we use in ranking.

The above ranking model which we refer to as the global model is trained on click logs of multiple users. As the user interacts with the system, we build a local ranking model which makes it more personalized.

5. INLOOK DESIGN

InLook [1] is engineered as a C# WPF desktop application using a MVC framework. Figure 5 depicts the design of InLook. All the searches get routed by the mail manager to the appropriate stores. Caches are built-in at each stores to speed-up the retrieval. The index is built using Lucene.Net. Our design allows pluggable algorithms for Autocomplete, Spell correct and Ranking.

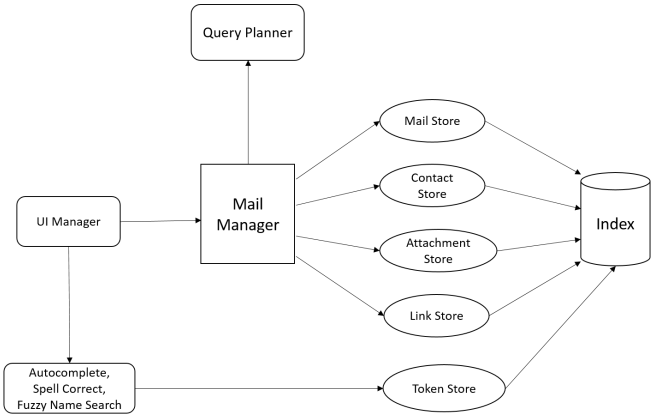


Figure 4: InLook Design

In addition to providing a fast and productive email experience, InLook also provides a test bed for new algorithms and experiences related to emails. InLook's modular design helps us experiment with other algorithms ranging from subject line auto-completion to varied UX experiences using the same code base. InLook provides us with an easy way to conduct user studies with emails and get their feedback.

6. PRELIMINARY RESULTS

We instrument InLook and log various metrics like query response time, auto-complete usage and click positions. These metrics were collected from 50 users over a period of one month.

Average response time	**792 ms**
Auto-complete benefit (% key strokes saved)	**50.8%**
Intent pane had one mail $\notin$ recency pane	**65%**
Clicks on mail in intent pane $\notin$ recency pane	**58%**

While these numbers are encouraging, the scale of experiment is small. We plan to dogfood with a larger audience. We are exploring different UX layouts to remove any unconditional click bias.

7. COMPARISON WITH OTHER APPLICATIONS

We compare InLook with following commercial offerings: Gmail [2], Inbox by Google [4] and Thunderbird by Mozilla [6]. None of these provide all the features we described in one place.

Gmail provides guided search but does not provide Fuzzy name search and relevance results. Google Inbox provides relevance pane (though not for all queries) but provides minimal guided search. Thunderbird provides relevance results and minimal guided search. Thunderbird client makes uses

[1] InLook can be downloaded from http://aka.ms/InLookApp

Feature Name	Feature Description
Body-Phrase-Match	Tokens (from Query) matched a phrase in the body of a mail
Subject-Phrase-Match	Tokens matched a phrase in the subject of a mail
URL-Match	Token matched a link in a mail
Filename-Prefix	Token matched prefix of a file name in an attachment
MailFrom-Match	Token matched (includes Fuzzy) a contact in "From" field of a mail
MailTo-Normalized	Token matched (includes Fuzzy) a contact in "To" field of a mail, normalized w.r.t. #recipients
Content-Match	Token matched a word in one of the attachments of a mail
Mail-Freshness	Decay feature which penalizes older mails

Table 1: Some of the features used in ranking

of tabs to display search results and emails. But, the algorithms employed in these offerings are not disclosed publicly making it hard to compare with quantitatively.

8. CONCLUSION

In this work, we presented InLook: a lightweight and productive email search experience. We have incorporated varied facets of email search into one single application. Specifically, we have incorporated guided search and have introduced an Intent pane for showing decorated/relevant results. Initial results suggest that a mix of relevant and time ordered mails is a good way to show search results. As future work, we plan to release InLook to a broader audience, to understand user behavior in detail. Exploring alternate UI layouts for displaying relevant and recent mails is an interesting future direction we want to pursue.

9. ACKNOWLEDGMENTS

We thank all our colleagues who helped dogfood InLook and gave us value feedback.

10. REFERENCES

[1] Email statistics. http://www.radicati.com/wp/wp-content/uploads/2015/02/Email-Statistics-Report-2015-2019-Executive-Summary.pdf.

[2] Gmail. https://mail.google.com.

[3] Gmail tabs. https://support.google.com/mail/answer/3055016?hl=en.

[4] Google's inbox. https://en.wikipedia.org/wiki/Inbox_by_Gmail.

[5] Lbfgs. https://en.wikipedia.org/wiki/Limited-memory_BFGS.

[6] Thunderbird. https://www.mozilla.org/en-US/thunderbird/.

[7] Samir AbdelRahman, Basma Hassan, and Reem Bahgat. A new email retrieval ranking approach. *International Journal of Computer Science and Information Technology*, 2010.

[8] Douglas Aberdeen, Ondrej Pacovsky, and Andrew Slater. The learning behind gmail priority inbox. In *in NIPS 2010 Workshop on Learning on Cores, Clusters and Clouds*, 2010.

[9] Sumit Bhatia, Debapriyo Majumdar, and Prasenjit Mitra. Query suggestions in the absence of query logs. In *SIGIR 2011, Beijing, China*.

[10] Abhijit Bhole and Raghavendra Udupa. On correcting misspelled queries in email search. In *AAAI, 2015, Austin, Texas, USA*.

[11] David Carmel, Guy Halawi, Liane Lewin-Eytan, Yoelle Maarek, and Ariel Raviv. Rank by time or by relevance?: Revisiting email search. In *CIKM 2015, Melbourne, VIC, Australia*.

[12] Dotan Di Castro, Zohar Shay Karnin, Liane Lewin-Eytan, and Yoelle Maarek. You've got mail, and here is what you could do with it!: Analyzing and predicting actions on email messages. In *WSDM, San Francisco, CA, USA, February 22-25, 2016*.

[13] David Elsweiler, Morgan Harvey, and Martin Hacker. Understanding re-finding behavior in naturalistic email interaction logs. In *SIGIR 2011, Beijing, China*.

[14] Mihajlo Grbovic, Guy Halawi, Zohar Shay Karnin, and Yoelle Maarek. How many folders do you really need?: Classifying email into a handful of categories. In *CIKM 2014, Shanghai, China*.

[15] Morgan Harvey and David Elsweiler. Exploring query patterns in email search. In *(ECIR), Barcelona, Spain, April 1-5, 2012*.

[16] Thorsten Joachims. Optimizing search engines using clickthrough data. In *ACM SIGKDD, 2002, Edmonton, Alberta, Canada*.

[17] Shaishav Kumar and Raghavendra Udupa. Multilingual people search. In *SIGIR 2010, Geneva, Switzerland*.

[18] Paul Ogilvie and Jamie Callan. Experiments with language models for known-item finding of e-mail messages. In *TREC 2005, Gaithersburg, Maryland, USA*, 2005.

[19] Raghavendra Udupa and Shaishav Kumar. Hashing-based approaches to spelling correction of personal names. In *EMNLP 2010, MIT Stata Center, Massachusetts, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*.

[20] Steve Whittaker, Tara Matthews, Julian A. Cerruti, Hernan Badenes, and John C. Tang. Am I wasting my time organizing email?: a study of email refinding. In *CHI 2011, Vancouver, BC, Canada*.