The version of record is available at: http://dx.doi.org/10.1561/1500000069

Search and Discovery in Personal Email Collections

Michael Bendersky[1], Xuanhui Wang[2], Marc Najork[3] and Donald Metzler[4]

[1] *Google; bemike@google.com*
[2] *Google; xuanhui@google.com*
[3] *Google; najork@google.com*
[4] *Google; metzler@google.com*

ABSTRACT

Email has been an essential communication medium for many years. As a result, the information accumulated in our mailboxes has become valuable for all of our personal and professional activities. For years, researchers have been developing interfaces, models and algorithms to facilitate search, discovery and organization of email data. In this survey, we attempt to bring together these diverse research directions, and provide both a historical background, as well as a comprehensive overview of the recent advances in the field. In particular, we lay out all the components needed in the design of a privacy-centric email search engine, including search interface, indexing, document and query understanding, retrieval, ranking and evaluation. We also go beyond search, presenting recent work on intelligent task assistance in email. Finally, we discuss some emerging trends and future directions in email search and discovery research.

Michael Bendersky, Xuanhui Wang, Marc Najork and Donald Metzler (2021), "Search and Discovery in Personal Email Collections", Foundations and Trends® in Information Retrieval: Vol. 15, No. 1, pp 1–133. DOI: 10.1561/1500000069.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

1

Introduction

Email has thrived as an electronic communications medium for at least five decades, with the first published email standards dating back to Bhushan *et al.* (1973). While the basic email format — a header containing email metadata and a body containing the message content — remained more or less unchanged through the decades, the types of information shared through email have been continuously evolving.

While email was originally developed with organizational and enterprise communications in mind, the success of web-based services like *Hotmail* and *Yahoo! Mail* in the late 1990's made email a popular consumer communication tool. Over the years, and with the rise of the various messaging applications, there have been reports on a decline in interpersonal email communications, especially among younger users (Tsotsis, 2011). However, consumer email traffic has still consistently kept growing. This discrepancy can be attributed in large part to the rise of machine-generated messages, such as store promotions, newsletters, receipts and bills (Maarek, 2017).

Despite advances in instant communications, email also remains a vital communication tool in the enterprise setting. A recent survey of 1,000 U.S. employees by Naragon (2018) finds that users spend more

The version of record is available at: http://dx.doi.org/10.1561/1500000069

than 3 hours on a weekday checking their work email. Roughly 50% of survey participants check both their personal and work email at least every few hours. Naragon (2018) also reports that in a work setting, email is a more preferred communication medium than either instant messaging (11% preference), or phone (16%), and is tied in popularity with face-to-face communications (31%).

The popularity of email in both our personal lives and in the workplace is in part due to its use for collaborative task management. Collaborative task management involves reminder creation, identification of messages related to the task, synthesis of information from these messages, and interaction with others in order to complete the task. Regardless of its limitations, email is often the preferred medium for these activities (Whittaker, 2005).

As a confluence of these factors, email remains a reliable repository of information about our personal and organizational communications, social networks, activities, financial transactions, travel plans, and work commitments. As our mailboxes grow, so does the need for the development of new effective approaches to information finding in this repository. As researchers repeatedly discover, there is a substantial difference between search in public data (e.g., web search) and private email collections.

First, chronology plays an important role for both email search algorithms and interfaces (Dumais *et al.*, 2003). Second, corpus size of single mailbox is drastically smaller than that of a large web corpus. This often leads to low recall, especially for longer queries, or when there is a vocabulary mismatch between user queries and their mailboxes (Carmel *et al.*, 2015; Li *et al.*, 2019b). Finally, developing effective search algorithms while stringently preserving the privacy of user information is a difficult research challenge (Bendersky *et al.*, 2018).

Therefore, in this survey, we provide an overview of the current state-of-the art techniques that focus on these unique aspects of email management, search and discovery. Since we assume that most of our readers are more familiar with the web search counterparts of these techniques, we contrast and draw comparisons between web and email search, when appropriate.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

1.1 Email Statistics

Before diving into describing the various email use cases, in this section we provide an overview of email usage, including general statistics, the demographic characteristics of its user base, and modes of email access.

The Radicati Group, Inc. (2019) report[1] states that the total number of emails sent and received per day will have exceeded 300 billion in 2020, and that email will be used by 4 billion people, over half of the world's population. Despite email being a mature technology, the report projects steady year-over-year growth of roughly 4% for the next several years. The Radicati Group, Inc. (2015) report also breaks down these statistics by business and consumer users, finding that the number of business emails exceeds the number of consumer emails sent and received, with both numbers projected to grow. The growth in the consumer email traffic is cited to be mainly due to machine-generated email, not interpersonal communication, which is consistent with other reports (Maarek, 2017).

These statistics demonstrate the importance of email in the business setting, and allow to draw a clear distinction between the personal email use case, and the enterprise use case (Narang *et al.*, 2017). This puts business email search and discovery in a clear connection to the existing work on enterprise search (Kruschwitz and Hull, 2017), with the added constraint that the corpora (user mailboxes) are private, rather than shared across the organization.

Narang *et al.* (2017) investigate email usage in a large organization (*Microsoft*) and report on the activities performed by a large sample of close to 300,000 US employees. In particular, they note that as mailbox size increases, people are much more likely to spend time on its organization by deleting, moving or marking email. Search activity also has strong positive correlation with the mailbox size. Activity analysis shows that $20\% - 35\%$ of all email activity involves organization, and $10\% - 20\%$ involves search, with the variation mainly attributed to the mailbox size and email deletion rate.

[1] The Radicati Group is an analyst firm specialized in tracking emerging communication and collaboration technologies, providing quantitative and qualitative market research. In this survey, we are quoting statistics provided in their 2015, 2018 and 2019 executive summaries, which are available online at www.radicati.com.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

For the personal email use-case, Carmel *et al.* (2017b) provide a fascinating peek into the demographics of the *Yahoo! Mail* US user base. Overall, they find that email users are older and more affluent than both the average web searchers, as well as the overall US population. They are also more likely to be female – 58.4% of all email searches come from women, as opposed to 49.7% of web searches.

While in the early days of email desktop clients using POP or IMAP were more prevalent, today many users use webmail or mobile clients to access their email. Both webmail and mobile email clients are usually controlled by a large email provider that also controls a centralized secure storage for all user mailboxes. A recent Litmus Email Analytics (2019) report indicates that only 18% of the email opens today can be attributed to desktop clients. The same report lists *Gmail*, *Yahoo! Mail* and *Outlook.com* as the most used webmail clients. Examples of international webmail providers also include, among others, *QQ Mail* by Tencent, *163/126 Mail* by NetEase, *Mail.Ru*, *Yandex*, *ProtonMail* and *GMX Mail*.

As most people access their email today through one of these large-scale centralized email providers, in the remainder of this survey we shall assume that the mailboxes are centrally and securely stored and managed. This setting provides the opportunity to develop new search and discovery capabilities using a large-scale dataset containing millions of user mailboxes. It also carries the challenge of developing these capabilities while maintaining user trust through audited access, data anonymization, and data erasure compliance.

Indeed, breaking user trust has been shown to have major implications for email providers. This is evidenced by negative public reaction to services like *Google Buzz*, which "automatically searched the user's most emailed contacts and added them as followers, thereby inadvertently exposing potentially sensitive communications" (Nowak, 2010), or Oath (Yahoo mail owner) purportedly selling consumer preferences gleaned from promotional emails to advertisers (Liao, 2010).

Therefore, the tension between the opportunities for novel user experiences and the challenge to preserve user trust is a major recurrent theme that runs throughout the email search and discovery research, and is discussed extensively in this survey. In particular, we dedicate

Chapter 7 to the challenges of privacy-preserving management of user data.

1.2 Email Management and Finding Strategies

In the previous section, we established the scale of email usage, and the importance of mailboxes as personal and organizational information repositories. In this section, we focus on the way that users keep track of and find information in these repositories in real-world settings.

The goal of the majority of email searches is re-finding information in previously seen emails, which relates it to the *known-item search* problem (Craswell *et al.*, 2005), where only one particular, known in advance item can fully satisfy the user information need. It is not surprising, therefore, that emails are frequently revisited, and most of the revisits are information seeking (Alrashed *et al.*, 2018).

Some information types that users seek during email revisits are listed in Table 1.1. Interestingly, finding task-related instructions is the most common reason for email revisit, which is in line with the prevalence of email usage for task management that is noted by other researchers as well (Whittaker, 2005; Lampert *et al.*, 2010).

Table 1.1: Distribution of information types users are looking for in email revisits, as reported in a survey of 395 corporate email users (Alrashed *et al.*, 2018).

Type of Information	Percent
Instructions to perform a certain task	24.1%
A document (e.g., attachment, link)	22.0%
An answer to a question that was previously asked	16.3%
status update	10.2%
A solution to a problem	9.0%
A task request to you	4.9%
A person/customer (e.g., contact information)	2.0%
An appointment/event	2.0%
Machine generated message (e.g., reservation)	0.8%
Other	8.6%

Ai *et al.* (2017) conduct a survey of 324 users to examine what message attributes facilitate searcher recall. They find that, unlike in web search, email searchers tend to remember more details about the

The version of record is available at: http://dx.doi.org/10.1561/1500000069

provenance of the messages they are interested in (e.g., sender and sent date – see Figure 1.1). This reflects greater familiarity with email than web pages, and re-affirms the known-item approach to email search. Ai *et al.* (2017) also find that this good attribute recall is not always reflected in the search query length and structure. Based on a sample of 2 million queries from Outlook email search logs, they report that advanced syntax is used in only 18% of the queries, and most of these advanced queries contain either `from:` or `to:` filters.

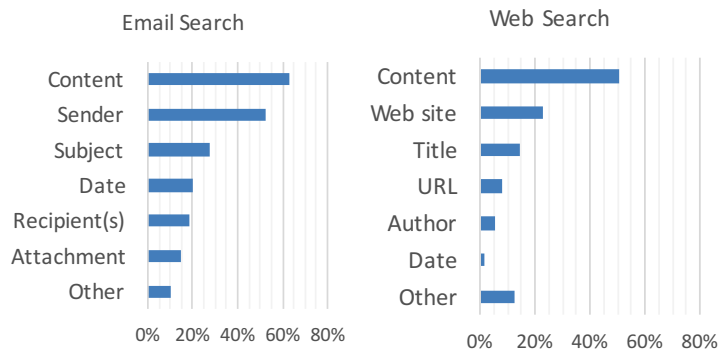


Figure 1.1: Percentage of searchers who remembered certain attributes, compared between email and web searches, based on a survey of 324 regular email users, conducted by Ai *et al.* (2017).

Users may also use other *email discovery* mechanisms beyond search to find the relevant information in their mailboxes. Examples of email discovery mechanisms include content recommendation, classification and information extraction. For instance, some email services can automatically tag emails with labels such as "Travel" or "Finance" (Grbovic *et al.*, 2014) and extract useful information like bill due dates or hotel check-in times (Sheng *et al.*, 2018). This can help with relevant information discovery without the need for conducting an explicit search.

Broadly speaking, most email search and discovery mechanisms discussed in this survey are in the realm of *personal information management* (PIM). PIM studies the organization and maintenance of information items stored for the purpose of completing personal or work-related tasks and activities. In fact, Whittaker *et al.* (2006) argue that email plays a critical role in three key PIM areas, including task management, personal archiving, and contact management.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

One notable exception to viewing email search and discovery as a sub-field of PIM, is access to mailboxes by third-parties who are not the persons to whom the email was addressed. Such access is conducted in cases such as legal e-discovery for litigation or government investigations (Oard and Webber, 2013), historical research (Task Force on Technical Approaches for Email Archives, 2018), or logging by organizational mail auditing tools (Microsoft 365, 2020). As this survey takes a user-centric approach to email search and discovery, most of these cases are outside of our scope. However, some of the described techniques are likely to be helpful in finding relevant information by third-parties as well.

1.3 Survey Scope and Organization

The majority of the research on email search and discovery that this survey draws upon has appeared over the past decade in a broad spectrum of information retrieval and data mining conferences including (but not limited to)

- ACM SIGIR Conference on Research and Development in Information Retrieval – https://dl.acm.org/conference/sigir
- ACM International Conference on Web Search and Data Mining – https://dl.acm.org/conference/wsdm
- ACM SIGKDD Conference on Knowledge Discovery and Data Mining – https://dl.acm.org/conference/kdd
- The Web Conference (formerly known as International World Wide Web Conference, or WWW) – https://dl.acm.org/conference/www
- The Conference on Information and Knowledge Management – https://dl.acm.org/conference/cikm
- Text REtrieval Conerence – https://trec.nist.gov/

We made our best attempt to provide a comprehensive survey of this large body of research, providing some historical perspective,

The version of record is available at: http://dx.doi.org/10.1561/1500000069

organizing it into broad themes, and finally suggesting some directions for future research. We also attempted to provide a perspective – based on the existing research, as well as our own experience – on the unique challenges facing the researchers in this field, contrasting it to the more commonly known web search setting.

Prerequisites This survey assumes minimal prior knowledge and should be relatively self-contained. We keep most of the discussions at a high level of abstraction, and refer the readers to the original research papers for technical details. However, some grasp of standard notation, concepts and techniques in information retrieval and machine learning can be beneficial for getting the most out of this survey. We suggest the following introductory and freely available books as useful accompanying references:

- Introduction to Information Retrieval, by Schütze *et al.* (2008)
- The Elements of Statistical Learning, by Hastie *et al.* (2009)

Target Audience We hope that the following audiences will find this survey useful:

- Search practitioners and engineers who want to be exposed to the scientific fundamentals of email search (or other personal search scenarios)
- Industry and academic researchers and graduate students in the fields of information retrieval, machine learning or natural language processing who are interested in better understanding the state-of-the-art and the emerging trends in email search and discovery.

Outline The remainder of this survey is organized as follows. In Chapter 2 we provide a high-level overview of the architecture of a standard email search engine. As there is no previously published work that summarizes such architecture, we do our best to synthesize multiple

The version of record is available at: http://dx.doi.org/10.1561/1500000069

disparate research avenues, and compare the different design and architecture choices to web search engines, which are likely to be more familiar to our readers.

Chapter 3 is dedicated to the evolution of interfaces for email search and discovery, from manually defined folders and exact search to relevance-based ranking and knowledge panels. In Chapters 4 and 5, we discuss the various aspects of email and query understanding, respectively. In these chapters, due to the heterogeneous topics discussed, we often go beyond the realm of email search and delve into other aspects of email management and discovery, including spam detection, labeling and templatization.

In Chapter 6, we once again broaden our scope beyond search and discuss various assistive applications that allow users to effectively find, manage and create email content. In this chapter, we also often go beyond the boundaries of the mailbox, and discuss how assistance can work across multiple personal content types (e.g., email, calendar entries or personal files).

Chapter 7 is dedicated to management of user data in email search and discovery. We discuss the best practices of privacy-preserving treatment of user data, as well as learning from sparse and biased click data in email search. We speculate on possible future research directions in personal search and discovery in Chapter 8, and conclude the survey in Chapter 9.

Special considerations There are three important considerations that we would like our readers to keep in mind as they make their way through this survey.

First and foremost, data and user privacy is an important *leitmotif* in email search and discovery. Our goal is to elucidate the importance of these topics, and the degree to which they affect how the research in the field is conducted. Therefore, we include a chapter dedicated to privacy-preserving user data management, and return to this topic throughout the survey.

Second, when possible, we try to draw parallels between email and web search. The latter may be a more familiar territory to many of our readers, as it has been one of the focal points of information retrieval

The version of record is available at: http://dx.doi.org/10.1561/1500000069

research for the past two decades. Contrasting email and web search also aids in highlighting the unique aspects of email search and discovery algorithms.

Finally, the readers are likely to notice that while the title of the survey focuses on email search, some chapters broaden their scope beyond email to other types of personal content, and modes of content management and discovery that go beyond search. This is by design, rather than mere lack of focus. We strongly believe that the future of personal content search and discovery lies in integrative approaches that seamlessly combine personal information across various content silos to best assist the users in completing their personal or work tasks.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

2

The Anatomy of an Email Search Engine

In this chapter we provide a high level exposition of the critical components of an email search engine. To better motivate this exposition, it is important to draw some comparisons between the architecture of an email search engine, and that of a generic, non-private search engine. Web search is the most commonly used example of a search system, and one which many of the readers may have some familiarity with. Therefore, in what follows we elaborate on several key aspects that distinguish between email and web search.

Corpus size The users of modern web search engines, like Google, Baidu, and Bing have access to hundreds of billions of web pages (Google Search, 2018). In contrast, in email search, the users only have access to their individual mailboxes, and therefore the number of searchable documents is limited. Thus, the proportion of queries for which no documents are retrieved is likely to be greater in email search than in web search, and increasing *recall* is an important part of email retrieval systems, and is further discussed in Section 2.3.3 and Section 5.3. It is important to note, however, that while the size of each *user's* corpus is small, the *overall* size of a large consumer or enterprise email

The version of record is available at: http://dx.doi.org/10.1561/1500000069

service index (e.g., Gmail or Outlook.com) rivals that of any web search engine. For instance, a recently released email statistics report (The Radicati Group, Inc., 2018) states that: "In 2018, the total number of business and consumer emails sent and received per day will exceed 281 billion".

Links and anchor text Anchor text and link graphs have traditionally been used in multiple components of large-scale web search engines, from crawling (Olston and Najork, 2010) and ranking (Brin and Page, 1998) to test collection generation (Asadi *et al.*, 2011). From the early days of web search, link-based signals like PageRank (Brin and Page, 1998), HITS (Kleinberg, 1999), and SALSA (Lempel and Moran, 2001) have helped these search engines to serve relevant, high-quality results even for very short and ambiguous queries (Najork, 2007). In contrast, in email search anchor text and links are non-existent, as there are no cross-references among user mailboxes. Instead, researchers working on email search often rely on other features that are specific to email structure, such as attachments, email threads and sender and recipient information (Carmel *et al.*, 2015; AbdelRahman *et al.*, 2010) to improve the quality of email search. More details on these structural features are provided in Section 2.4.

Implicit user feedback Implicit user feedback, which is most commonly derived from clicks on links on the search results page, has been successfully leveraged in web search applications for training machine learned ranking models (Agichtein *et al.*, 2006; Joachims, 2002). However, the use of click data in email search ranking has been limited by the lack of cross-user interactions with the same item. To overcome this limitation, Bendersky *et al.* (2017) recently proposed a click aggregation-based approach, which is discussed in more detail in Section 7.3.

Content and query dynamics While document freshness is an important consideration in web search (Dai *et al.*, 2011), web search exhibits a wide spectrum of content and query dynamics. For instance, Kulkarni *et al.* (2011) categorize intent dynamics in web search into three main

The version of record is available at: http://dx.doi.org/10.1561/1500000069

categories: (a) *zoom* – intent zooms in on to the current event, (b) *shift* – intent undergoes a gradual shift over time, and (c) *static* – relatively stable intent over time. For the third intent type, document freshness plays a lesser role than the long-term context of the past user behavior. In contrast, email searchers exhibit a very strong bias towards recent messages (Dumais *et al.*, 2003). This is reflected by the fact that results in many of the popular email search interfaces are presented in descending chronological order (see Section 3.2 and Section 2.3.2 for more discussion on this phenomenon).

Adversarial content and spam While spam, phishing and other adversarial content are prevalent in email, they are generally not addressed as a part of the search architecture. Rather, all suspected spam emails are grouped into a "Spam" folder and are ignored during the search process. This is in contrast to web search, where adversarial approaches like link manipulation, click baiting, and content plagiarism play an important role in search engine design (Spirin and Han, 2012). We discuss the issue of adversarial content filtering later on in this survey, in Section 4.1.1.

Search tasks It is also important to note the differences between the tasks that the users are expecting to achieve via email search as compared to web search. In web search, many search tasks and needs can be categorized as *general*: users have a broad idea of what they are trying to achieve, but do not have a particular web page in mind. In email search, the tasks are *specific* as users are often looking for a particular email or thread, related to their information need. This type of search behavior falls under the category of *known-item search* (Craswell *et al.*, 2005). In a recent survey, Ai *et al.* (2017) found that while in web search 42.9% of the users report that their search tasks are general, the portion of general tasks in email search is only 9.1%. In Section 2.5 we further discuss how the existing test collections and metrics reflect this tendency.

Data privacy Unlike in web search, in email search both the content of the users' documents (emails) and their queries are of a private nature.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

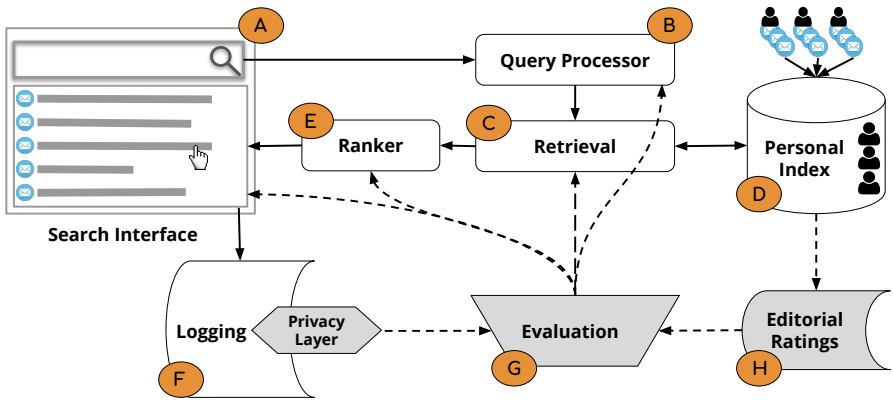


Figure 2.1: An end-to-end email search engine architecture, as described in this chapter. Solid arrows demonstrate the online flow of a search request. Dashed arrows and shaded shapes indicate the offline evaluation process.

This requires researchers to employ privacy-preserving techniques when examining either document (Di Castro *et al.*, 2016b; Sheng *et al.*, 2018) or query (Foley *et al.*, 2018) content. Privacy-preserving data processing is often facilitated by techniques such as data de-identification, k-anonymization or differential privacy. We discuss these technique in detail in Section 7.1.

2.1 Architecture Overview

We now present an end-to-end architecture of an email search engine and discuss its similarities and differences as compared to the standard web search architectures. As shown in Figure 2.1 an email search engine consists of the following modules.

(A) **Search Interface** – user interface used to perform the search activity. It may be a page in a web browser, a desktop client, or a mobile app. In Chapter 3 we provide some examples of email user interfaces, and the principles that guide the development of these interfaces.

(B) **Query Processor** – module that is responsible for rewriting the input user query received from the search interface into its

The version of record is available at: http://dx.doi.org/10.1561/1500000069

internal form, which may include synonym expansion, spelling corrections, stemming, etc. We provide a detailed overview of the most pertinent techniques for query processing for email search in Chapter 5.

(C) **Retrieval** – a system that retrieves all the candidates that match the user query from the index. In general, the retrieval system has to be fast and lightweight, as it considers the contents of the entire personal index. Standard retrieval optimization strategies (Turtle and Flood, 1995) may be used. See Section 2.3 for more details on the retrieval system.

(D) **Personal Index** – real-time and access-controlled indexing system that processes incoming emails. Each search query is matched against only emails belonging to the user who issued the query. We go into more details of this index in Section 2.2.

(E) **Ranker** – re-orders the top results retrieved by the retrieval system, either based on chronological ordering or some relevance criteria. Oftentimes, learning-to-rank techniques are applied at this stage to improve relevance (Carmel *et al.*, 2017b; Zamani *et al.*, 2017). See Section 2.4 for more details.

(F) **Logging** – the logging system records the user's interactions with the search interface including queries, clicks, views, and the features of the emails. It is important to note that since all the logged content is private to the user, the email search logging system often includes a privacy layer that applies data anonymization techniques (e.g., k-anonymity or differential privacy) to facilitate privacy-preserving data access during system development and experimentation (Bendersky *et al.*, 2017; Foley *et al.*, 2018). We discuss various techniques for data anonymization in Section 7.1.

(G) **Evaluation** – as common in search engines, evaluation is required to continuously monitor the performance of the email search engine, as well as for measuring the effect of new changes that are introduced to any of its components. Evaluation can be done either manually, using editorial ratings, or automatically by tracking the

The version of record is available at: http://dx.doi.org/10.1561/1500000069

clicks and session metrics in the logs. We describe both types of evaluation in Section 2.5 in more details.

(H) **Editorial ratings** – unlike in web search, where editorial ratings can be done by third-party raters, in email search it is common to use personal raters who are requested to issue queries and evaluate results from their own mailboxes (Carmel *et al.*, 2015; Dumais *et al.*, 2003). See Section 2.5.1 for more details on how these personal editorial ratings are collected.

2.2 Email Indexing

While the indexing of public web documents is a frequently explored topic in the information retrieval literature, there is much less published work on indexing of private document collections, including email. Therefore, while there is no single canonical publicly available work on best practices for implementing an email message indexing system, in this section we discuss some desiderata for such a system, and how they have been addressed in prior work.

2.2.1 Access Controlled Indexing

Obviously, email messages are private, and should only be accessed and retrieved by the user who owns the messages. This also affects the extent to which researchers and engineers can make use of the email contents to improve search and discovery experiences. For instance, practitioners often use synthetic emails generated via k-anonymization (Di Castro *et al.*, 2016b; Sheng *et al.*, 2018) to perform information extraction, classification, and other learning tasks over email data.

Email services are often hosted by a central provider such as Yandex, Microsoft, or Google. Thus, the most obvious approach to private indexing is maintaining a single secure indexing system that divides the collection into $|U|$ individual indices, where U is the set of all users. The most secure, albeit costly, solution is maintaining a separate document collection, a separate vocabulary, and separate posting lists for each user $u \in U$, such that only user u can access their index. A less costly but more complicated solution is to maintain a cross-user vocabulary

The version of record is available at: http://dx.doi.org/10.1561/1500000069

and postings lists, and instead to rely on the retrieval mechanism to only return the documents of a single user, say by tagging every email with the identity of the recipient and adding that identity tag as a conjunct to each query. This design choice is more space efficient, as longer postings lists have better compression ratios.

Alternatives that do not assume an existence of a single trusted index host, while not widely deployed, do exist in published research. For instance, Bawa *et al.* (2009) propose a methodology for constructing distributed peer-to-peer indices for private or shareable documents with provable privacy guarantees.

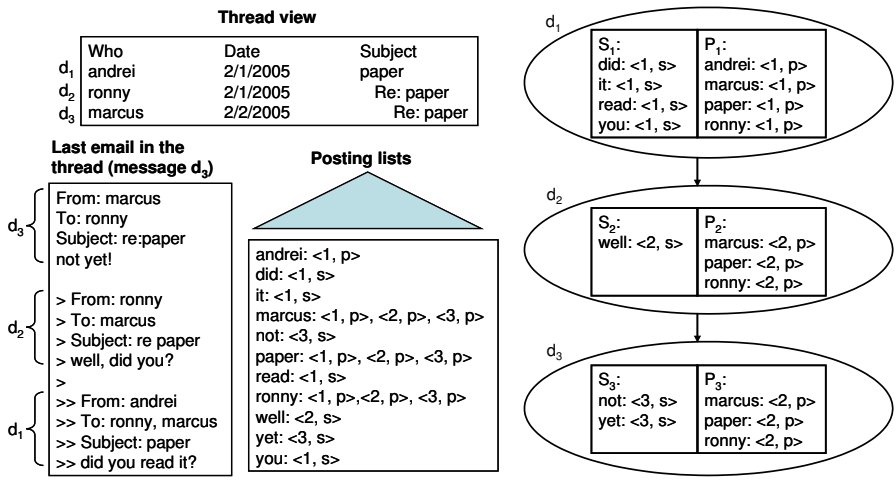
2.2.2 Handling Content Duplication

It is clear from the description in the previous section that a naive implementation of private indexing is costly. For instance, email threads may be shared across users, and would be indexed multiple times for each user. Moreover, as is evident from prior work (Di Castro *et al.*, 2016b; Sheng *et al.*, 2018; Wendt *et al.*, 2016), a large portion of email data consists of machine-generated email, which can be represented as templates for significant savings in indexing capacity.

To address the issue of content duplication across threads, Broder *et al.* (2006) propose indexing email threads in a document tree structure, which allows sharing the content between the nodes of the tree. This requires two simple additions to the standard inverted index structure. First, each posting list entry needs to contain a bit indicating whether it is *shared* (s) or *private* (p). Second, threads are stored in a document tree structure. This is illustrated in Figure 2.2. This structure reduces posting list duplication, as each term is indexed once per thread. As an example, in Figure 2.2, the term "did" has only one posting list entry, despite appearing in both $d1$ and $d2$. This is due to the fact that "did" is shared with the descendants of document $d1$ in the document tree, namely with $d2$ and $d3$. Broder *et al.* (2006) also propose an efficient query evaluation algorithm that makes use of the proposed inverted index structure.

While Broder *et al.* (2006) mainly focus on reducing content duplication due to message threading, another major source for duplication are

The version of record is available at: http://dx.doi.org/10.1561/1500000069



(a) The inverted index (b) The document tree

Figure 2.2: An example of index encoding for a thread $d1 \rightarrow d2 \rightarrow d3$, which eliminates posting list duplication, as each term is indexed only once per thread. (From Broder *et al.* (2006)).

machine generated emails (Maarek, 2017). In these emails, the majority of the content is generated from a single template and repeated across multiple emails (consider, for instance, shopping receipts or credit card statements). For such templated emails, index compression techniques proposed for dealing with versioned document collections (Claude *et al.*, 2011; He *et al.*, 2010) may be readily applied.

2.2.3 Fast Incremental Updates

The amount of new email that users receive on a daily basis creates the need for designing indexing systems that can handle fast addition of messages to the search index. In addition, since users often delete emails to de-clutter their mailboxes, the indexing needs to support fast item deletion. A naive implementation that would simply append or delete a new posting list entry for every update consumes a non-trivial amount of resources. Frequent posting list updates also result in less efficient query processing.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

To combat this, Hawking (2010) proposes a combination of a baseline and update indices which are searched in parallel. The baseline index contains all the historical data and can be optimized for read operations. The update index is much smaller and only includes the most recent updates. The baseline index can be merged with the update index in a batch manner on a pre-defined schedule.

2.3 Retrieval Techniques

In this section we discuss various retrieval techniques employed in email search engines, including search operators, chronological ordering and relevance-based retrieval.

2.3.1 Search Operators

An email message consists of its free-form content as well as structured metadata ("fields"), such as the sender, the recipient, the time received, whether it was opened, etc. It may be convenient for the users to use structured search operators to filter the results by the value of these fields (see Table 2.1). While powerful, these search operators are rarely used explicitly, but are often implicitly present in query intents. For instance, Ai *et al.* (2017) found that searchers often remember certain message attributes like sender, recipients, subject, or date, prior to conducting a search. Therefore, an interesting research direction would be to develop a translation model from implicit intents to explicit operator queries.

In practice, search operators are commonly implemented using field restricts (i.e., only term matches in a specific field are used for retrieval). Thus, the query processor and the index should support complex structured queries. See Lucene as an example of a search system that implements field restricts in its query language.[1]

2.3.2 Chronological Ordering

A common strategy for presenting search results to the users in email web and desktop clients is descending chronological sorting of the

[1] https://lucene.apache.org/core/2_9_4/queryparsersyntax.html#Fields

The version of record is available at: http://dx.doi.org/10.1561/1500000069

Table 2.1: Common examples of search operators used in email search (based on Gmail Help (2018)).

Operator	Description	Example Usage
`to:`	Filter based on email sender	`to:alice@mail.com`
`from:`	Filter based on email recipient	`from:bob@mail.com`
`cc:`	Filter based on a copied recipient	`cc:kate@mail.com`
`subject:`	Filter based on email subject	`subject:dinner`
`has:`	Filter based on an email attribute	`has:attachment`
`is:`	Filter based on the email state	`is:read`
`after:`	Filter based on the email age	`after:2018/01/01`
`label:`	Filter based on a certain label	`label:personal`
`size:`	Filter messages exceeding size	`size:10M`

results. This approach is intuitively appropriate for email search, and is supported by research evidence. For instance, Dumais *et al.* (2003) found that 22% of the items opened in search were first seen within the last week. In addition, Dumais *et al.* (2003) presented an interface that supported multiple sort options such as *Date*, *Author*, and *Rank*. They found that "regardless of which sort order people started with, they issued more queries in which they sorted the results by *Date*". In their experiments, *Rank* was based on a simple Okapi-based algorithm.

In general, the chronological ordering is usually implemented using a straightforward one-pass matching algorithm that requires that all query terms appear in at least one of the message fields, and then sorting the retrieved documents in descending received timestamp order (Carmel *et al.*, 2017b). This approach is efficient and simple to implement, and it scales well to web-based email services that receive millions of search requests. It also guarantees generally high quality results that do not violate users expectations. However, it has an issue with recall, in case of a mismatch between the query terms and the email text.

2.3.3 Relevance-based Retrieval

To combat the issue of recall presented in the previous section, several approaches have been proposed.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

Relaxed Match Carmel *et al.* (2015) propose a simple partial match approach to increase the number of returned results, which requires that *at least one* query term should appear in the message content (or one of its fields). While this approach may surface a lot of irrelevant results as well, especially for longer queries, these results can be demoted by the next stage ranker. Thus, this approach increases the recall of the retrieval stage, while pushing the burden of precision to the ranking stage.

Fielded match Ogilvie and Callan (2005) treat the email search problem as a variant of known-item search in structured documents and propose a hierarchical language modeling approach that models the email as a mixture of its fields. The language model θ_e of an email e is thus estimated using a linear combination of the language models of the different fields

$$P(w|\theta_e) = \sum_f \lambda_f P(w|\theta_{MLE(f)}), \tag{2.1}$$

where f is a field of e and $\theta_{MLE(f)}$ is the Maximum Likelihood Estimate of the multinomial language model of f. Ogilvie and Callan (2005) consider email, subject line, email thread, and the subthread which contains all the replies to the email as fields. The entire collection language model is also used for smoothing. Then, the retrieved results are ordered by: $P(Q|\theta_e) = \prod_{i=1}^{|Q|} P(q_i|\theta_e)$.

Note that this approach is similar to the relaxed match approach, as it allows partial matches, however it provides a more principled and robust relevance ranking of the retrieved messages, as the query term matches are weighted by their language model scores. The scores produced by this model can also be fed as features to the ranking stage, described in the next section.

Query expansion The approaches described above deal with the partial match case, but it is also important to note the importance of query expansion in the retrieval stage as well, as the size of the retrieval corpus is relatively small. Any number of techniques can be applied for this, including log-based, mailbox-based and pseudo-relevance based query

The version of record is available at: http://dx.doi.org/10.1561/1500000069

expansion, which is investigated by Kuzi *et al.* (2017). Additional details on applications of query expansion to email search are covered in more detail in Section 5.3.

Approximate Nearest Neighbors In recent years, the researchers have suggested the use of approximate nearest neighbor algorithms for the retrieval of documents that are semantically related to the query in some dense embedding space (Aumüller *et al.*, 2017). While, to the best of our knowledge, there are no published results on specific applications of approximate nearest neighbors retrieval to email search, many of the developed algorithms and software[2] can be readily adapted to this use case.

2.4 Relevance Ranking

As discussed in the previous section, after the broad matching retrieval stage is employed, we are faced with the challenge of ranking the top results in the best possible manner using all the available information about the retrieved emails. Unlike in other information retrieval scenarios, e.g., web search, where the underlying collection is large, the number of truly relevant documents to the query in email search is typically quite small. As mentioned above, some researchers even model the task as one of known-item search (Craswell *et al.*, 2005; Ogilvie and Callan, 2005) in which case only a single, known in advance item can fully satisfy the user information need.

Thus, solely relying on the retrieval stage to generate the best ordering does not work well in practice (Dumais *et al.*, 2003; Carmel *et al.*, 2015) and can often generate "embarrassing results" (Carmel *et al.* (2015)), i.e., results that are completely irrelevant to the query, but are being pushed to the top of the ranked list simply by the virtue of matching some query terms.

The first published research to employ standard learning-to-rank techniques to this problem is the work by Carmel *et al.* (2015). In general, the learning to rank methods used in email search are fairly

[2]See https://github.com/erikbern/ann-benchmarks for a comprehensive list.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

standard and include RankSVM (Carmel *et al.*, 2015), logistic regression (Ramarao *et al.*, 2016), LambdaMART (Wang *et al.*, 2016a), or feed forward neural networks (Zamani *et al.*, 2017) among others. Therefore for the purpose of this section, the main interest are the features used by the learning-to-rank models, as they differ from the standard learning-to-rank features used in web search. For instance, Table 2.2 provides a detailed breakdown of features used in the seminal work by Carmel *et al.* (2015).

In general, the features used in the ranking stage in email search commonly fall into the following broad categories:

1. **Sender features** depend on some characteristics of the email sender that indicate the affinity between the sender and the searcher (e.g., number of sent or received emails, number of times sender emails were searched for, etc.). In the enterprise setting, affinity based on non-email communications may also be useful, e.g., calendar appointments, shared documents, etc.

2. **Recipient features** reflect the characteristics of the recipient group – individuals and mailing lists in the *to* or *cc* fields of the email – with respect to the searcher.

3. **Message features** depend on the attributes of the email message. These features may include message freshness, the last time it appeared in search results, presence of attachments, and any folders, system-assigned labels (Grbovic *et al.*, 2014; Wendt *et al.*, 2016) or templates (Bendersky *et al.*, 2017) that the email is associated with.

4. **Action features** are based on the actions that the user has explicitly performed on the message (e.g., opens, replies, forwards, stars, spam assignments, etc.).

5. **Query similarity features** measure the topical similarity between the query and the email. These could simply be the scores produced during the retrieval stage, or other common IR similarity measures, e.g., BM25, cosine similarity, language models, query term overlap, and so on. More recently, researchers have used deep

Table 2.2: Commonly used features in learning-to-rank for email search grouped by type as described by Carmel *et al.* (2015).

Type	Sub-Type	Feature (or features set) Name	Description
Message	Freshness	time by days/weeks/months/years	message age by respective resolution
	User Actions	replied	message was replied
		forwarded	message was forwarded
		draft	message is saved as draft
		flagged	message is flagged by star
		seen	message was read
		spam	message was marked as Spam
		ham	message was marked as Ham
	Attachment	has attachment	message has an attachment
		attachment type	set of binary features for attachment type
		attachment size	set of binary features for attachment size range
	Folder	folder type	set of binary features indicating the folder of the message (specific system folder or a personal folder)
	Exchange Type	reply/forward	binary features indicating if message is a reply or a forward
		in thread	message is part of a thread correspondence
Recipient		in To/Cc/Group	set of binary features corresponding to wether recipient is in To/Cc/Group (group is Bcc or mailing list)
Sender	Vertical	sender-user connection	strength of correspondence between sender and user
		self correspondence	sender is the user (binary, message was sent from the user to himself)
	Horizontal (over all users)	sender outbound/inbound traffic	set of binary features for sender's outbound/inbound traffic range
		sender urls	set of binary features for range of urls num in sender's messages
		sender recipients num	set of binary features for range of recipients num in sender's messages
		sender-users actions	ratio of the sender's messages on which a specific action was performed
Query Similarity		BM25f	BM25f similarity between message and query
		tf-idf	tf-idf similarity between message fields and query
		coord	fraction of query terms found in the message

The version of record is available at: http://dx.doi.org/10.1561/1500000069

neural networks to model the text matching problem (Mitra and Craswell, 2018). Various types of neural matching models were also found to be useful in email search (Zamani *et al.*, 2017; Shen *et al.*, 2018; Li *et al.*, 2019b).

6. **Searcher features** enable personalization of email search results, based on what is known about the user performing the search. For instance, Zamani *et al.* (2017) propose using situational context features including time and location to improve search results. In the context of query completion, Carmel *et al.* (2017b) demonstrate that using user search history, as well as their demographic information (age, gender, income level, state of residence) can significantly improve suggestion quality. Similarly, Foley *et al.* (2018) show that the semantics of fine-grained user location, when available, can improve query completion quality by up to 20% for single-character query prefixes.[3]

7. **Click features** that are derived from cross-user interactions with the same item, while highly valuable in web search, are not directly useful in email search. This is due to the fact that each user only interacts with their own mailbox, resulting in a highly sparse click distribution. To combat this data sparsity, Bendersky *et al.* (2017) propose a parameterization approach that enables effective leveraging of historical click data for ranking. In this approach, an email is represented by a set of attributes (e.g., email template, labels, subject n-grams) that generalizes across multiple users. Click data is then aggregated by these attributes and incorporated into the underlying learning-to-rank model. We discuss this click aggregation approach in more detail in Section 7.3.

2.5 Evaluation and Metrics

Evaluation is a central part of any search system. In order to make search quality improvements, one needs to measure success and make

[3] As derived from the Google Maps platform: https://cloud.google.com/maps-platform/places/.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

decisions accordingly. Therefore, in this section, we discuss the evaluation paradigm for email search, which differs significantly from web search.

Email mailboxes are private document collections. It is often hard to argue for "objective" relevance, especially given the short and ambiguous nature of email search queries. Carmel *et al.* (2017b) report that email search queries consist of 1.5 terms on average. Therefore, techniques employed for email search evaluation differ significantly from web search evaluation. First, for editorial ratings, the researchers tend to rely on personal raters who annotate their own data. These annotations are often not reusable (as mailbox contents frequently change) and costly to construct. Thus, the researchers often rely on implicit user feedback from search logs for evaluation as well.

We describe techniques for test collection construction in more detail in Section 2.5.1. Then, in Section 2.5.2 we discuss the success metrics usually employed in the field. Finally, in Section 2.5.3, we provide a brief listing of publicly available email search and discovery test collections.

2.5.1 Test Collections

Synthetic queries AbdelRahman *et al.* (2010), in an attempt to build a reusable test collection, use the publicly available Enron dataset (Cohen, 2015). They sample a small number of previously annotated discussion topics or category labels (Berry *et al.*, 2001) to create a synthetic query set. Some examples of synthetic queries include "attachment", "project progress" and "california energy crisis". Then, up to 100 emails are retrieved by multiple retrieval methods from a corpus consisting of mailboxes of several Enron employees. All the retrieved emails are pooled and manually graded by three judges on a $[0, 1, 2, 3]$ scale. The proposed retrieval methods are then evaluated using the $NDCG@k$ metric.

AbdelRahman *et al.* (2010) provide an interesting example of how to apply the known principles of the Cranfield paradigm to email search evaluation, however their method has several shortcomings. First, it is unclear whether the synthetic queries indeed reflect true user information needs. Second, the relevance is determined by external raters, who may not have a full understanding of the personal search task underlying the

The version of record is available at: http://dx.doi.org/10.1561/1500000069

query. Therefore, in later studies researchers have focused on personal mailbox ratings, as we discuss next.

Personal ratings In the personal rating setting, selected users (e.g., employees of the corporation where the research is conducted) are instructed to search over their own mailboxes and evaluate the retrieved results (Dumais *et al.*, 2003; Carmel *et al.*, 2015). There are several important aspects that differentiate this type of personal rating from the standard editorial ratings over public web data.

First, the raters are generally instructed to issue queries that match the standard query usage. To achieve this, Carmel *et al.* (2015) require that the issued queries match patterns mined from an email search log, e.g., *[<sender name> <body word>]*. This type of requirement ensures that while the information needs themselves are private to the raters, the *types* of their information needs can be generalized to other users.

Second, the results need to be judged with respect to the intent that "the editor had in mind at query time" (Carmel *et al.*, 2015). This goes to demonstrate the importance of the situational context in email search (Zamani *et al.*, 2017). The relevance of an email message is not absolute, and is likely to change over time, as new emails arrive and the intent of the user drifts. This is in contrast to web search, where human raters are provided with general, objective guidelines for their task.

Implicit feedback While personal ratings are an important component of email search evaluation, they are costly to collect, have a limited shelf-life due to the mailbox dynamics, and are often not reusable across search systems. In addition, even with the most rigorous setup, editorial evaluations cannot fully capture the nature of personal search, and do not scale as collections evolve and query intents drift over time. Therefore, much of the more recent work (Bendersky *et al.*, 2017; Wang *et al.*, 2016a; Zamani *et al.*, 2017; Ramarao *et al.*, 2016) resorts to using implicit feedback instead of editorial judgments. It is well known that implicit feedback, such as clicks, is an abundant albeit biased resource (Joachims, 2002). For web search, implicit feedback is rarely used without further validation by ground truth editorial evaluations,

The version of record is available at: http://dx.doi.org/10.1561/1500000069

since it can be maliciously manipulated (Najork, 2009). However, it is more applicable in email search for several reasons.

First, since users interact solely with their own mailboxes, click-bait and click-spam, which are major research challenges in the context of web search (Spirin and Han, 2012), are much less likely in the email search scenario.[4] Second, click data captures well the dynamic nature of email data, and can be used to discover new relevant documents. Third, as discussed in the beginning of the chapter, email search is a type of known-item search. Therefore, users usually know what they are looking for, and their clicks are likely to be informed by this prior knowledge.

Finally, recent studies demonstrate that even though click noise and bias still exist in email search, recent advances in unbiased learning-to-rank (Joachims *et al.*, 2017) can be used to combat them through techniques like inverse propensity weighting (Wang *et al.*, 2016a), regression-based expectation-maximization estimation (Wang *et al.*, 2018), and trust bias modeling (Agarwal *et al.*, 2019a). We discuss all of these advances in more detail in Chapter 7.2.

2.5.2 Success Metrics

Ranking metrics are crucial for continuous progress in search engine quality and are at the heart of all information retrieval research. As there are multiple ranking metrics to consider, the choice of an evaluation metric is heavily dependent on an application. For instance, in the web search setting, researchers consider metrics such as $NDCG@k$ (normalized discounted cumulative gain at rank k) to account for graded ratings and ensure high quality at the top of the list. In contrast, for the TREC newswire collections, where recall plays a larger role, mean average precision (MAP) of the entire ranked list is often used.

As previously discussed, email search is commonly considered to be a special case of a known-item search (Ogilvie and Callan, 2005), where only one relevant email message per query is expected. Thus, researchers often use mean reciprocal rank (MRR) (Carmel *et al.*, 2015;

[4]Note that email search generally skips the Spam folder, which is discussed in more detail in Section 4.1.1.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

Ogilvie and Callan, 2005; Wang *et al.*, 2016a; Bendersky *et al.*, 2017) as the evaluation metric. Formally, MRR over a set of N queries is defined as

$$MRR = \frac{\sum_{i=1}^{N} \frac{1}{rank_i}}{N}, \tag{2.2}$$

where $rank_i$ is the rank of the relevant message for the i-th query. In addition to the MRR metric, Carmel *et al.* (2017b) also suggest *success*@k metric, defined as

$$success@k = \frac{\sum_{i=1}^{N} \mathbb{I}(rank_i \leq k)}{N}, \tag{2.3}$$

which is equivalent to the percentage of queries for which the relevant message was ranked at or above position k.

As discussed in Section 2.5.1, relevance in email search is often derived from implicit feedback, such as user clicks. Since clicks are often biased (Joachims *et al.*, 2017), we might need to weight them for bias correction. Accordingly, Wang *et al.* (2016a) propose the *weighted MRR* (*wMRR*) metric. It is a variant of MRR, where each query i is assigned a weight w_i such that

$$wMRR = \frac{\sum_{i=1}^{N} \frac{w_i}{rank_i}}{\sum_{i=1}^{N} w_i}, \tag{2.4}$$

where weights w_i are estimated from click data. The various estimation methods are discussed in detail in Section 7.2. In general, queries with clicks at higher ranks receive lower weights during the evaluation. This is motivated by the well-known issue of position bias (Joachims *et al.*, 2007), which causes email messages at the top of the ranked list to be clicked more often than those lower in the ranked list irrespective of their relevance.

In addition to purely click-based metrics, Ashkan and Metzler (2019) propose metrics that capture other types of implicit user feedback: abandonment rate

$$ar = \sum_{i=1}^{N} a_i, \tag{2.5}$$

The version of record is available at: http://dx.doi.org/10.1561/1500000069

where a_i is a binary indicator of whether the user abandoned the i-th query without clicking on any results; and time to click

$$ttc = \sum_{i=1}^{N} t_i, \tag{2.6}$$

where t_i is the time to the first click. For both abandonment rate and time-to-click metrics, lower values indicate better search experience.

Ashkan and Metzler (2019) also advocate for making the metrics more *user-centric* by normalizing them with respect to the user's historical behavior. Taking the MRR metric as an example, a user-centric version of the metric can be defined as

$$pMRR = \frac{\sum_{i=1}^{N} \frac{p_i}{rank_i}}{\sum_{i=1}^{N} p_i}, \tag{2.7}$$

where $p_i = \log(\frac{\overline{rank_i}}{rank_i} + 1)$ and $\overline{rank_i}$ denotes the average click rank for the user issuing the i-th query. Intuitively, this type of user-centric metric will reward changes that result in the most effort savings to the user. Ashkan and Metzler (2019) find that user-centric metrics like *pMRR* are generally more discriminative than their standard counterparts in detecting changes in A/B experiments, suggesting that they are a good choice for online evaluation.

It is important to point out that the metrics described in this section are applicable regardless of whether chronological ordering (Section 2.3.2) or relevance-based ranking (Section 2.4) are used, as they measure the user response to the ranked list regardless of how the list was produced. In fact, Carmel *et al.* (2015) demonstrate that relevance-based ranking can boost click-based metrics like *MRR* or *success*@k by large margins over the chronological ordering baseline.

2.5.3 Public Datasets

As discussed in the beginning of this section, the private nature of email corpora creates a substantial barrier for entry for the academic researchers in the field. It is not surprising, therefore, that a large fraction of the research presented in this survey was conducted at technology companies that provide web mail services, like Yahoo, Microsoft, or

The version of record is available at: http://dx.doi.org/10.1561/1500000069

Google. There are, however, two existing public datasets that have been extensively used by the researchers in the field.

The most well known of these datasets is the Enron Email dataset (Cohen, 2015). The Enron dataset contains data from about 150 users, mostly senior management of Enron, organized into folders. The dataset contains a total of about half a million messages. This data was originally made public, and posted to the web, by the Federal Energy Regulatory Commission during its investigation of the Enron corporation. A version of the Enron Email dataset is available for download online.[5] There are multiple research articles published using the Enron dataset on topics including classification (Bekkerman, 2004), search (AbdelRahman *et al.*, 2010), and visualization (Heer, 2005) of email.

It is important to note that while the Enron dataset is currently publicly available for download, it is not officially supported by any data consortium or institution, and the privacy of the email correspondents has not been preserved through any reduction procedure. Therefore, the authors advise that the researchers who decide to use this dataset take extra precautions to ensure that their analysis and algorithms preserve the privacy of Enron correspondents.

A newer publicly available email research dataset is the Avocado Research Email Collection (Oard *et al.*, 2015), which consists of 1.3 million emails taken from 279 accounts of a defunct information technology company referred to as "Avocado". It is similar in structure to Enron Email, albeit larger. It also contains some additional information not available in Enron such as contact information, email attachments, etc. The Avocado dataset has been recently used in research on commitment detection (Azarbonyad *et al.*, 2019), intent modeling (Lin *et al.*, 2018), attachment recommendation (Van Gysel *et al.*, 2017) and action item extraction (Mukherjee *et al.*, 2020). Sayed *et al.* (2020) use the Avocado dataset as the retrieval corpus for creating a test collection containing search topics, as well as email relevance and sensitivity judgments for each topic.

[5] https://www.cs.cmu.edu/~./enron/

The version of record is available at: http://dx.doi.org/10.1561/1500000069

The Avocado dataset can be licensed from the LDC website.[6] Unlike Enron, Avocado `README` specifically discusses privacy reduction methods performed prior to it release, including reducting sensitive email content, attachment content and private correspondent information. Therefore, the authors strongly encourage the use of Avocado dataset over Enron for any new research projects.

[6] https://catalog.ldc.upenn.edu/LDC2015T03

The version of record is available at: http://dx.doi.org/10.1561/1500000069

3

Search Interfaces

In this chapter we discuss the evolution of email management interfaces from filing and organizing to free-form search. We cover a few early systems that heavily rely on foldering as a means of content discovery, and the gradual move to search interfaces in Section 3.1. Section 3.2 discusses attribute-driven search interfaces that allow flexible sorting of search results. Finally, in Section 3.3, we focus on the emergence of relevance-based email search interfaces.

3.1 From Foldering to Finding

Managing email overload through organizing it into thematic folders has a long history. For instance, Cincotta (1983) recommends managing email communications related to different projects or activities by creating appropriate UNIX directories. User defined email folders were also adopted by early web email clients such as RocketMail (later acquired by Yahoo!) or Hotmail (later acquired by Microsoft).

Pachyderm, an experimental email system developed in 1997 (Birrell *et al.*, 1997), presaged many of the features of today's web-based email systems: email was stored by a service and accessed through a web-based client. Users could organize their messages by attaching labels,

The version of record is available at: http://dx.doi.org/10.1561/1500000069

with *inbox*, *unread*, *hidden* and *deleted* being predefined labels. The Pachyderm UI surfaced all defined labels in a list (akin to a folder list). Messages were full-text indexed, and users could retrieve their messages by issuing complex queries on content terms as well as fields and labels. For convenience, users could name a query and thereby add it to the pane of labels; this facility was akin to a smart foldering mechanism.

Gmail by Google, which was announced on April 1st, 2004, in addition to advanced search functionality and large storage, also features *labels*. The main distinctions between folders and labels is that each email can be assigned multiple labels, and that label names can be used in search (see Table 2.1). bluemail by IBM (Tang *et al.*, 2008) further suggests the use of *tag clouds* for email management, inspired by their popularity in contemporary social networking applications.

As the volume of email communication continues to grow, manual email filing and folder, label and tag management is becoming more laborious and time consuming. Moreover, there is no clear indication that careful folder and label curation is actually helpful for email re-finding. For instance, in a study of usage patterns of 345 IBM employees, Whittaker *et al.* (2011) find no correlation between filing behavior and success in finding tasks. Therefore, in recent years, research focus has shifted towards automatic email categorization techniques, which are discussed in more detail in Section 4.1.2, and to interfaces that facilitate more effective email search, which we discuss next.

3.2 Attribute-based Ordering and Filtering

Cutrell *et al.* (2006) argue that web search and email search strategies are unalike; the most important difference being *"that people are familiar with many different characteristics of their information, as well as the context(s) in which they previously encountered them"*. According to Cutrell *et al.* (2006), context plays a critical role in recalling personal information. There are many contextual cues that are used to find the relevant email, including its sender, recipients or the attachments it contains. Perhaps the most important contextual cue of all is time, as searchers are much more likely to look for emails related to recent tasks or events.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

Dumais *et al.* (2003) develop an experimental email search interface SIS (Stuff I've Seen) that recognizes the importance of these contextual cues. Instead of presenting a fixed ranking of messages to the user, SIS provides a flexible interface that allows re-ordering and filtering of email messages based on multiple attributes (see Figure 3.1). In a study of over 8,000 searches conducted by 233 volunteers, Dumais *et al.* (2003) find that – regardless of the initial ranking order – users apply chronological ordering to their search results in over 60% of the cases. This validates the importance of chronological ordering in email search.

Indeed, chronological information has been recognized as a crucial element in effective email management and discovery since the early days of the research in the field (Whittaker and Sidner, 1997). Therefore traditional approaches to email search often consist of the following two stages:

1. Retrieval of all emails that strictly match the search query terms
2. Ordering of the retrieved results in a reverse chronological order (see, e.g., Figure 3.2(a)).

Even today, many of the web email services use this chronological ordering as their default search ranking mechanism. However, chronological ordering has its limitations, and incorporating relevance can lead to significant improvements in email search experience.

3.3 Relevance-Based Search Interfaces

Carmel *et al.* (2015) elucidate two main drawbacks of the chronological ordering in email search. First, it makes the discovery of older messages hard, as (a) recalling the exact attributes of these messages is difficult, and (b) they will be ranked low in the chronological ordering. Second, the chronological ordering imposes strict query match in order "to avoid embarrassing, non-relevant yet recent results from being displayed at the top of the list" (Carmel *et al.*, 2015). This significantly degrades the recall of the email retrieval stage.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

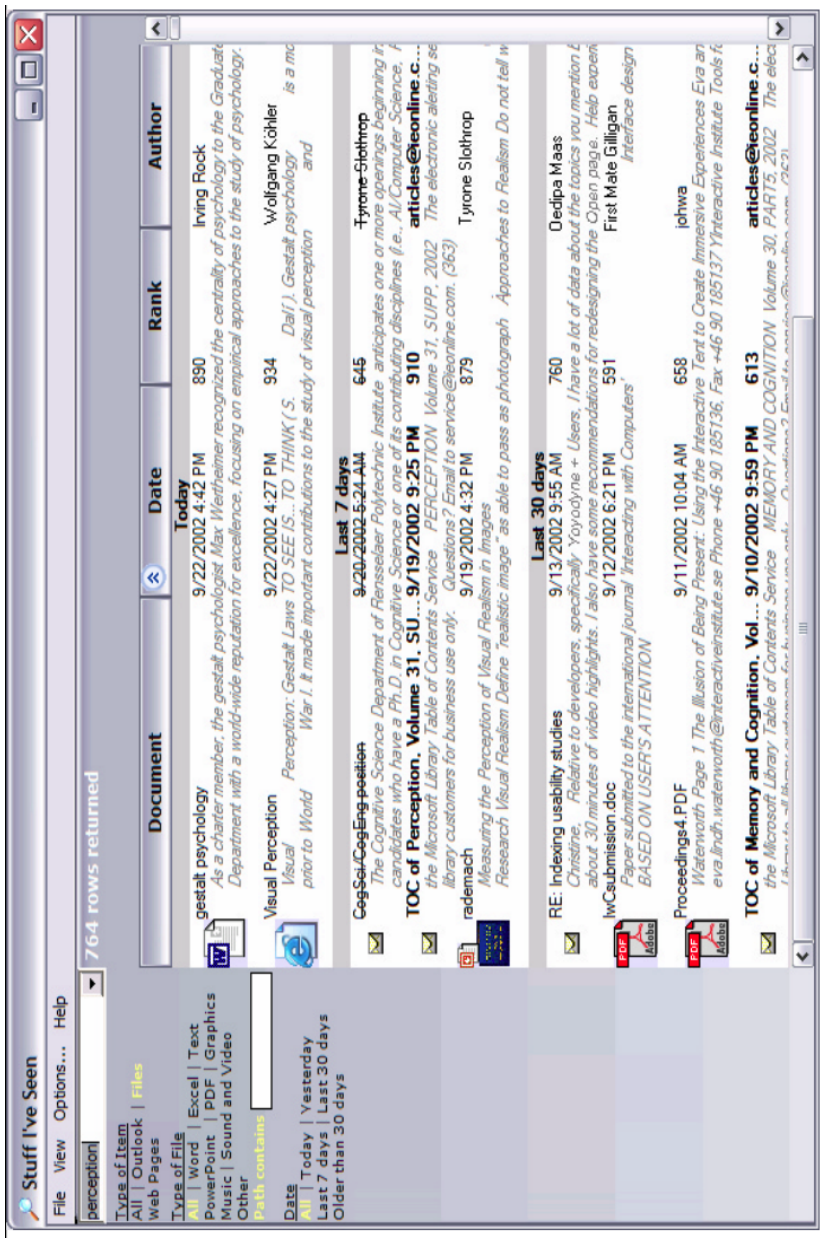


Figure 3.1: SIS (Stuff I've Seen) email search interface, showcasing the usage of different contextual attributes for filtering and ranking (Dumais *et al.*, 2003).

The version of record is available at: http://dx.doi.org/10.1561/1500000069

As an example, consider Figure 3.2 that shows the results retrieved for query *"sigir 2020"* in (a) chronological mode, and (b) relevance mode. Note that in Figure 3.2(a) the messages with the highest likelihood of relevance (calls for papers and tutorials) are not even displayed at the top, as messages about various events in 2020 from the mailing list *[SIG-IRList]* dominate the search results.

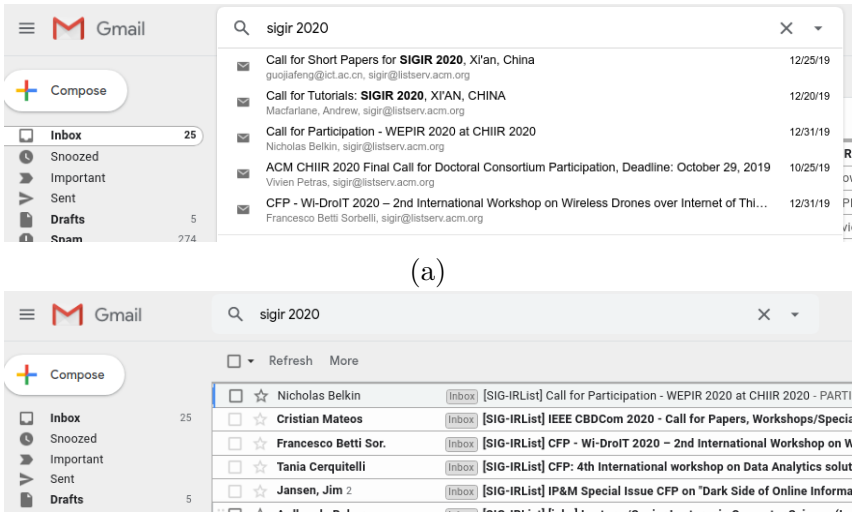


Figure 3.2: Illustration of the results for query *"sigir 2020"* in (a) relevance, and (b) descending chronological modes, as presented in a Gmail webmail client. If the user does not accept any of the relevance results in (a), and presses ENTER, they will be redirected to (b).

Despite its shortcomings, chronological ordering does provide the benefit of predictability. Users often rely on scrolling through the results in a chronological order to improve recall (*"I received the address to Alice's place **after** her invite to the party was sent"*). In a study of 345 users of an email client that supports search, folders and finding tagging, Whittaker *et al.* (2011) find that scrolling still accounts for 62% of all email accesses. Therefore, in real-world applications, both relevance and chronological rankings can provide value.

For instance, Gmail currently provides both the search-as-you-type top relevance results as well as the chronological results, if the user does

not accept any of the relevance-based results (see Figure 3.2). Yahoo Mail provides a Date/Relevance toggle for the displayed search results. Finally, Outlook.com provides a hybrid interface that combines the top relevance results and chronologically sorted results (Figure 3.3). Carmel *et al.* (2017a) refer to these top relevance results as *heroes*, and discuss some potential implementations of this hybrid interface. They report that in live experiments in Yahoo Mail, the hybrid approach results in 12% improvement (as measured by the Mean Reciprocal Rank metric) over chronological ranking. The improvements in Yahoo enterprise email are even higher (18% MRR gain), which validates the importance of relevance email ranking in the organizational setting.

Figure 3.3: An illustrative example of hybrid "heroes" relevance results (*HList*) followed by chronological results (*TList*) displayed on a mobile device, as shown by Carmel *et al.* (2017a).

As email relevance ranking algorithms continue to improve, we are likely to see more innovation in email search interface design, going beyond displaying ranked lists of emails. For instance, as shown in Figure 3.4, search can directly surface structured information relevant to the query through information cards, similar to the knowledge panels that

The version of record is available at: http://dx.doi.org/10.1561/1500000069

are common in web search.[1] Examples of such structured information may include tracking numbers of recently shipped orders, frequent flyer numbers, bill amounts and due dates, upcoming event date and locations, etc. Structured information cards were implemented in the short-lived *Inbox by Gmail* web and mobile clients (Kaushal, 2016), however they are still not commonly seen in the major email clients. We discuss some research challenges that need to be solved to foster wider adoption of such interfaces in Section 8.5 of this survey.

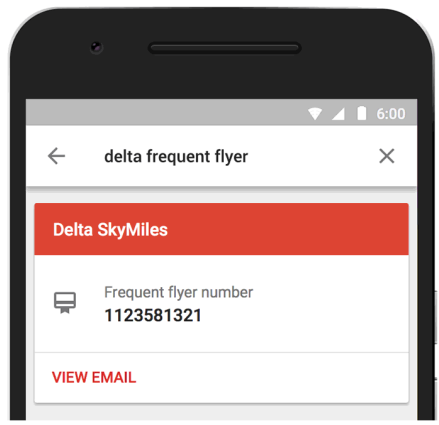


Figure 3.4: Knowledge panel in Inbox by Gmail (a defunct service), which directly surfaces the relevant answer to the query *"delta frequent flyer"*, without the need to read through the email (Kaushal, 2016).

[1] https://support.google.com/knowledgepanel/answer/9163198

The version of record is available at: http://dx.doi.org/10.1561/1500000069

4

Mailbox Understanding

In this chapter, we focus on algorithms and techniques designed specifically for mailbox organization and information extraction from email corpora. While many of these algorithms are related to, and inspired by standard text processing techniques, they were all designed or tailored to address the specific challenges posed by the unique nature of email data.

First, in Section 4.1, we focus on mailbox organization through the lens of various clustering and classification techniques. Effective mailbox organization is a major challenge that many users face on a daily basis both in their personal and professional lives. In addition, since email communication is often abused by spamming and phishing, we also cover techniques for fighting these adversarial activities.

Then, in Section 4.2 we discuss techniques for processing unstructured email content with a focus on various information extraction tasks such as signature extraction, quotation detection, and others.

Finally, in Section 4.3 we discuss template induction and extraction techniques for machine-generated email. This line of research is motivated by the proliferation of bulk-sent machine-generated emails (receipts, bills, and reservations) in user mailboxes (Maarek, 2017).

The version of record is available at: http://dx.doi.org/10.1561/1500000069

4.1 Mailbox Organization

Organizing one's mailbox into manageable groupings that facilitate productivity has been a long-standing challenge that predates the web and even graphical user interfaces. For instance, an early version of the Navy Email Service User Guide (Cincotta, 1983) states that:

> *"Managing ones mail is important. If the number of messages you receive daily encompasses many projects and various activities you would want to organize your mail into various files possibly even various directories".*

The user guide then goes on to suggest UNIX system utilities for email management. While email services have improved immensely since these early days, effective mailbox management still remains a challenge. This challenge is further exacerbated for business email use, as the average worker sends and receives more than 120 emails a day, and this number has been steadily rising over the past several years (The Radicati Group, Inc., 2015).

Accordingly, in this section, we provide an overview of some algorithms and techniques that empower users to effectively organize and manage their mailboxes. We start in Section 4.1.1 by describing the problem of combating adversarial content such as spam and phishing, and continue with foldering (Section 4.1.2) and clustering (Section 4.1.3) mailbox organization techniques.

4.1.1 Adversarial Content Filtering

Spam

Email spam detection and filtering is one of the classical applications of machine learning to text categorization. The research on spam detection goes back for more than twenty years. While the original methods relied on either manual or automatic rule development (Cohen *et al.*, 1996), early research showed that naive Bayes classifiers work surprisingly well for this task (Pantel and Lin, 1998; Sahami *et al.*, 1998). Naive Bayes use for spam filtering was further popularized by Graham (2003) who demonstrated a filtering rate of 99.75% on his own mailbox.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

Modern production-grade spam detection systems rely on features that go beyond email text for spam filtering. Taylor *et al.* (2007) report that sender reputations (the fraction of spam messages reported for the sender) as well as DomainKeys Identified Mail Signatures (DKIM) are important features used by the Gmail service.[1] Taylor *et al.* (2007) also identify the ability to process very large high-dimensional datasets in a distributed fashion as an important requirement for a production-grade spam filtering system.

The *TREC Spam Track* 2005 – 2007 evaluated the spam filtering effectiveness of the competing systems on a chronological sequence of email messages (Cormack, 2007). The systems were evaluated using a combination of a public (Enron) and private email corpora. The track participants had no access to the private corpora, and their system code was sequentially executed on the emails in these corpora by the track coordinators, to provide a realistic simulation of the spam filtering in production systems. The overall effectiveness of the competing systems improved year over year (Cormack, 2007), with Relaxed Online SVMs, which compute an approximate Support Vector Machine solution at greatly reduced expense, significantly outperforming other text classifiers, and achieving a misclassification rate of one in a thousand or better (Sculley and Wachman, 2007).

Phishing

Email phishing is an attack, where a malicious sender impersonates a trusted source to obtain sensitive information (e.g., passwords, bank account information, etc.) from the email recipient. Ramzan (2010) formally identifies a phishing attack as one having all of the following characteristics:

1. *Brand Spoofing* The attacker attempts to convince the user that the email message originates from a trustworthy brand.

2. *Website Involvement* The phishing email contains a link that redirects the user to a malicious site for the purpose of data collection.

[1] https://tools.ietf.org/html/rfc6376

The version of record is available at: http://dx.doi.org/10.1561/1500000069

3. *Sensitive Information Solicitation* The phishing website offers a mechanism to enter personal user information.

There are some standard measures to tackle phishing by preventing email address spoofing using techniques like DKIM, mentioned above. In the data mining and the information retrieval communities, researchers also have focused on machine learning techniques to detect phishing.

For instance, Abu-Nimeh *et al.* (2007) perform a study using various machine learning algorithms and a set of "bag-of-words" features similar to the spam detection work described in the previous section. Fette *et al.* (2007) further demonstrate that using a set of specialized features (e.g., IPs instead of site names in the email links, or the age of linked-to domain names) can improve phishing detection effectiveness when compared to standard spam detection algorithms.

Another research strand focuses on detecting potential phishing sites, independently of the emails that link to them. Phishing site detection is usually performed by measuring the similarity of a suspicious site to either legitimate trusted brand sites (Wenyin *et al.*, 2005) or other known phishing sites (Cui *et al.*, 2017).

4.1.2 Email Categorization

Users are likely to experience email overload as more and more information accretes in their mailboxes (Whittaker and Sidner, 1997), which naturally gives rise to attempts to automatically categorize and label email. Most of the earlier automatic email categorization research focuses on leveraging standard text classification techniques. For instance, Kiritchenko and Matwin (2001) use co-training between email body-based and header-based SVM classifiers. Bekkerman (2004) uses the Enron dataset to do a performance comparison between several popular text classifiers including Maximum Entropy, Naive Bayes, SVMs and Winnow (Littlestone, 1988), demonstrating that a simple-to-implement Winnow classifier is not only efficient, but also as effective as a more complex SVM classifier on this task.

More recent work addresses specific issues that make email categorization inherently different from standard text classification. First, individual foldering and labeling strategies vary significantly, which

The version of record is available at: http://dx.doi.org/10.1561/1500000069

makes developing and surfacing a unified category set to users that can address all their needs intractable due to the large number of possible categories. For instance, Koren *et al.* (2011) manually identify 2,000 English-only labels using 6,000 commonly-used folder names.

Cohen *et al.* (2004) take an alternative approach, and instead of a flat labeling scheme, propose a taxonomy of verbs and nouns (see Figure 4.1) that "jointly describes the *email speech act* intended by the email sender". For instance, an email can be associated with a "Propose Meeting" or a "Request Opinion" speech act. Cohen *et al.* (2004) presciently describe how the speech acts associated with an email can be used by a virtual personal assistant to provide timely reminders of certain commitments users have made.

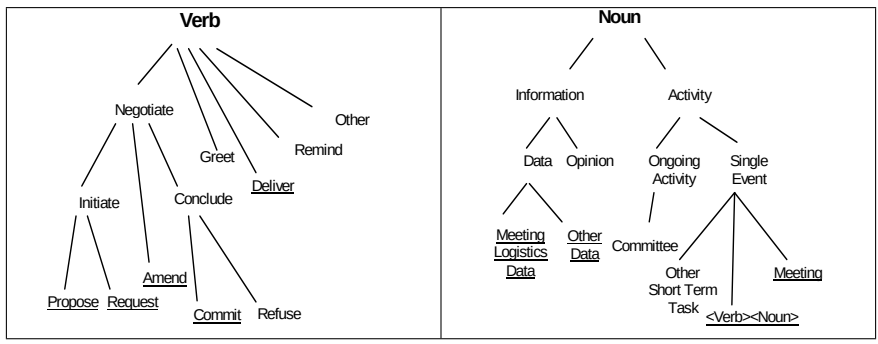


Figure 4.1: An ontology of verbs and nouns that compose the email speech acts as shown by Cohen *et al.* (2004). Underlined nodes indicate intents for which classifiers were trained in the original paper. The *<Verb><Noun>* pair in the figure indicates that the email speech acts may also be defined recursively, e.g., *(remind (deliver data))* is a valid act.

Grbovic *et al.* (2014) entirely move away from personal communications and focus instead on categorizing machine-generated emails, which according to some estimates (Ailon *et al.*, 2013) account for the majority of email traffic. Bootstrapping 100,000 popular human generated folders, Grbovic *et al.* (2014) build LDA topic models with a varying number of topics K using a concatenation of the emails in each of these folders as a single document. The optimal K should ensure that both each individual topic, as well as the overall set of topics achieve significant

The version of record is available at: http://dx.doi.org/10.1561/1500000069

coverage of the popular folders. They find that $K = 6$ best fulfills this requirement, and the resultant set of labeled topics is:

{human, career, shopping, travel, finance, social}.

As machine-generated emails can be associated with structural templates (see more on this in Section 4.3.1), Wendt *et al.* (2016) propose leveraging these templates for improving email categorization using a similar set of email labels: *{receipt, finance, travel}*. Their method is based on two intuitions: first, within a single template all emails should be assigned the same label; second, textually similar templates are likely to have the same label. Thus, Wendt *et al.* (2016) construct a template graph, where edge weights are defined by textual similarity, and each template is associated with a label distribution provided by a seed classifier.

Wendt *et al.* (2016) demonstrate that running label propagation (Ravi and Diao, 2016) over this graph results in a significant coverage increase across the given labels. It can also be used in conjunction with topic modeling to discover new labels in the data such as *politics*, *music*, or *fashion*.

Another interesting property of email labels is time dynamics. Unlike in standard newswire or web text classification, mailbox content is rapidly evolving. In an early work in this area, Segal and Kephart (2000) introduce SwiftFile, an online classifier that predicts the most likely folders the email should be moved into. The classifier is initialized based on the current folders, which are represented by the centroids of the weighted bag-of-words vectors of the messages they contain. When a new email is received, the centroids are updated, and the predictions for the subsequent emails may change.

With the advent of deep neural networks, researchers have explored their application to the email categorization task as well. For instance, Zhang *et al.* (2017) use an LSTM-based classifier to predict the likely category of future emails in a thread. Sun *et al.* (2018) propose a framework for jointly learning embeddings for emails and users, using as input sequences of email templates users both receive and open.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

4.1.3 Mailbox Clustering

While most existing email clients have adopted foldering and labeling organization paradigms, there are other alternative approaches to mailbox organization. As an interesting example, Bar-Yossef *et al.* (2006) propose organizing the mailbox under the assumption that it is *"an egocentric social network, consisting of contacts with whom an individual exchanges email"*. As this egocentric network could be large, producing meaningful groupings or clusters over it may prove difficult. To this end, Bar-Yossef *et al.* (2006) propose a new cluster ranking framework that outputs the maximal clusters in the network, ordered by their strength. For an unweighted network $G = (V, E)$, this can be simply solved by outputting all maximal cliques in G, however just using cliques may be limiting for realistic networks. Instead, the authors propose the notion of *network cohesion*:

$$cohesion(G) = min_{\{S,A,B\}} \frac{|S|}{min\{|A|, |B|\} + |S|}, \tag{4.1}$$

where $S \subseteq V$ is a *vertex separator* that, when removed along with its incident edges from G, separates G into two disconnected components with vertex sets A and B. Intuitively, the more cohesive the network G is, the harder it is to break it into large pieces (A and B) by removing a small number of nodes from the network (S). The network cohesion values will range from 0 for disconnected networks to 1 for cliques. The notion of cohesion can be further generalized for weighted networks, by summing over a sample of possible weight thresholds that can be used to generate a pruned unweighted version of the network. This process is called *integrated cohesion*.

Bar-Yossef *et al.* (2006) propose an efficient algorithm called C-Rank for ranking the clusters in the weighted contact network by their integrated cohesion and discuss its application to the mailbox clustering problem. They include both anecdotal examples (Figure 4.1) as well as an empirical study over the Enron corpus, which includes recall of maximal communities as well as robustness to data changes, showcasing the effectiveness of the C-Rank algorithm.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

Table 4.1: Top ten clusters emerging from running the C-Rank cluster ranking algorithm over one of the co-authors' mailbox contact network (from Bar-Yossef *et al.* (2006)).

Rank	Weight	Size	Member IDs	Description
1	163	2	1,2	grad student + co-advisor
2	41	17	3-19	FOCS program committee
3	39.2	5	20,21,22,23,24	old car pool
4	28.5	6	20,21,22,23,24,25	new car pool
5	28	2	26,27	colleagues
6	28	2	28,29	colleagues
7	25	3	26,30,31	colleagues
8	19	3	32,33,34	department committee
9	15.9	19	35-53	jokes forwarding group
10	15	14	54-67	reading group

4.2 Unstructured Email Processing

Email has long been an interesting data source for information extraction and natural language understanding. While it would not be possible to fully cover all this work in a single section, we attempt to provide a short overview of some of the techniques that are applied to *unstructured email*, i.e., email communications that are not expected to adhere to any particular structure or format. Such communications are most likely to be of personal (and often informal) nature, and the tasks that are studied in the literature are generally applied in this context.

4.2.1 Latent Structure Detection and Summarization

There is a significant amount of research focusing on a correct identification of certain email components. For instance, Carvalho and Cohen (2004) study the problem of *signature block extraction*, wherein they build a machine learning model to identify the set of lines that contain sender signature, including name, phone number, affiliation, etc. This type of analysis can be a building block for applications that automatically construct a detailed user contact list. Carvalho and Cohen (2004) demonstrate that using a set of lexical and syntactic features based on various regular expressions in a sequential classification algorithm such

The version of record is available at: http://dx.doi.org/10.1561/1500000069

as Conditional Random Fields (CRFs) (Sutton and McCallum, 2012) can achieve above 95% precision and recall on this task.

Related to the signature block extraction task, Minkov *et al.* (2005) study an application of CRFs to named entity extraction from emails. Elsayed and Oard (2006) further expand this line of research into modeling identities that tie an informal signature like *Bob*, or an entity mention like *Mr. Bruce*, to a unique email identifier *robert.bruce@enron.com*. Going beyond mentions of individuals, Gao *et al.* (2016) describe an automatic approach for constructing a collection-specific organization knowledge base. This is done via extraction of email domain mentions from an email corpus (e.g., *haas.berkeley*), and linking them to real-world entities (e.g., *Haas School of Business*) via Wikipedia lookup and Google search.

Another common research theme in detecting latent structure in unstructured emails is correct resolution of complex thread structures (Carenini *et al.*, 2007), which also includes removal of noisy text resulting from headers, quotations and signature blocks (Lam, 2002; Rambow *et al.*, 2004), and representation of email communications as a dialogue (Wan and McKeown, 2004; Hu *et al.*, 2009). Thread structure detection enables effective summarization of verbose email threads into short and cohesive narratives (Carenini *et al.*, 2007; Wan and McKeown, 2004; Rambow *et al.*, 2004).

Lampert *et al.* (2009) take a more holistic approach, and argue that in order to successfully process unstructured email it is important to identify all of the *email zones*, each with a different function. They identify nine such zones: *author*, *greeting*, *reply*, *forward*, *signoff*, *signature*, *advertising*, *disclaimer*, and *attachment*. While some of these zones may have a consensus definition, some may be highly dependent on a user or an email client. Therefore, discovering this latent 9-zone structure is not straightforward. Lampert *et al.* (2009) develop an SVM-based classifier based on graphic (layout and presentation), orthographic (distinctive characters or character sequences), and lexical features that achieves over 85% precision in the 9-zone segmentation task, using a sample of 400 annotated emails from the Enron dataset.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

4.2.2 Task Detection and Extraction

There has been a recent resurgence of interest in a variety of NLP applications that aim to leverage the colloquial and subjective nature of the unstructured email medium. Some of these applications include intent detection (Lin *et al.*, 2018; Azarbonyad *et al.*, 2019), sentiment analysis (Hangal *et al.*, 2011), Big Five traits prediction (Shen *et al.*, 2013), understanding communication differences between genders (Mohammad and Yang, 2011), and others.

As email is commonly used for tracking action items and to-do's (Whittaker, 2005), task detection in email is one particular application that has been attracting researcher interest for quite some time. Multiple machine learning approaches have been developed both for email-level task detection (Lampert *et al.*, 2010), as well as for identifying particular email sentences that have a call to action (Bennett and Carbonell, 2007).

In the most recent work on the subject to date, Mukherjee *et al.* (2020) propose a novel abstractive approach to generating action items from emails. First, they apply a classifier to identify emails that contain a commitment to perform a certain action item. For such emails, they first rank the sentences in the email based on their relevance to the aforementioned commitment, and then use a seq2seq model (Sutskever *et al.*, 2014) to generate a coherent action item using the commitment, and the email sentences most relevant to the commitment. Experiments using roughly 10,000 annotated emails from the Avocado corpus demonstrate the superiority of their abstractive approach to a simple extractive baseline (see Figure 4.2 for an example).

From: Kirstin Barnes *To:* Nannie Jacobs *Subject:* Ready for Product Launch
Nannie, I am ready for the product launch. I need to include some of the enhancements in the presentation. I'll submit what is already completed and then do the remaining after the meeting.. Kirstin Barnes Product Engineer AvocadoIT, Inc.
GOLD: Submit presentation with product enhancements.
PRED: Submit the enhancements for product launch.

Figure 4.2: An illustrative email and action item extraction example (from Mukherjee *et al.* (2020)). The identified commitment sentence is highlighted. GOLD is the action item written by the human judge, and PRED is the seq2seq model prediction. The sentences have been paraphrased and names changed due to the data sensitivity of the Avocado dataset.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

4.3 Machine-Generated Email Processing

An important and unique aspect of mailbox content that we focus on in this section is the proliferation of machine-generated content in email communications. Grbovic *et al.* (2014) reported that 90% of inbound non-spam Yahoo! mail traffic originated from bulk senders across a variety of verticals such as Retail, Travel, Social, Finance, etc. These machine-generated messages vary widely in both their importance to the recipients and their intent, and include multiple email types from marketing newsletters to critical information like flight tickets and purchase receipts. However, there is one common theme when considering this wide range of machine-generated email types. Email services, in order to surface timely and important information to their users, have to identify the underlying templates that generate these emails, and must be able to effectively extract information from the various fields in these templates.

To achieve this goal, multiple techniques were developed for understanding structured email content. These techniques can be roughly structured into three main stages that are covered in more detail in the remainder of this section.

First, in Section 4.3.1 we discuss techniques for grouping emails by their underlying templates. Second, in Section 4.3.2 we discuss how these templates can be threaded into cohesive sequences. Finally, in Section 4.3.3 we discuss how various types of information can be extracted from email templates.

4.3.1 Template Induction

Structured email templates were first introduced by Ailon *et al.* (2013). Most abstractly, Ailon *et al.* (2013) make the assumption that given any machine-generated email e, we can efficiently compute its template identifier $\tau(e)$, as well as a list of variable values $var(e)$. We work under this assumption for the remainder of this section.

Templates can be thought of as groupings of semantically identical messages, where some variable fields are replaced. For instance, all purchase receipts from a particular retailer can be grouped into a

The version of record is available at: http://dx.doi.org/10.1561/1500000069

single template and the list of template variables include product prices, delivery dates, etc. Figure 4.3 demonstrates a template example.

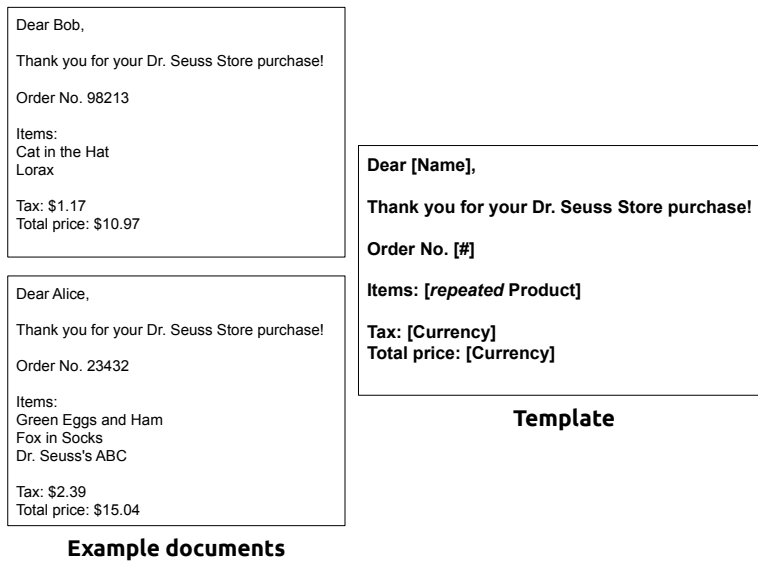


Figure 4.3: An example of template τ with two sample machine-generated emails associated with it. The variable fields *var* are denoted by square brackets.

In particular, Ailon *et al.* (2013) propose a subject template method that works in two stages. First, emails are grouped by senders. Only bulk senders, i.e., senders that send a large volume of email (e.g., `usps.com`, `amazon.com`) are considered. Then, subject lines (e.g., `Your order #1123-222 was received`) are analyzed to derive regular expressions of the form `Your order * was received`.

While many association rule mining algorithms are applicable here, in practice a simple technique that replaces long numbers, proper names, unique identifiers and words with probability below a certain threshold per sender with a wildcard results in reasonable template accuracy (Ailon *et al.*, 2013).

Subject-based templatization, while providing reasonable accuracy in many cases, does have its shortcomings. First, it completely ignores the body of the email. Therefore, two emails e and e' with the same subject will always be grouped into the same subject template (i.e.,

The version of record is available at: http://dx.doi.org/10.1561/1500000069

$\tau(e) = \tau(e')$), even if the bulk sender updates its template generation algorithm between the sending of these two emails. Second, if the subject templatization technique fails to extract a reasonably long regular expression, it may be prone to creating a large default cluster like *, which will not be semantically cohesive and simply group together multiple unrelated messages.

To avoid these shortcomings, Avigdor-Elgrabli *et al.* (2016) propose structural templates that use the email HTML structure. In particular they utilize XPaths – expressions that specify a full path from the document root to some target node in an HTML document. Avigdor-Elgrabli *et al.* (2016) propose two clustering approaches that take as an input <*sender, XPath list*> tuples to derive structural templates. The first approach, referred to as *stripped clustering*, collapses repetitions of sub-structures in the XPath-list into one instantiation of each sub-structure. This ensures that emails with different numbers of items (e.g., itemized receipts) all fall into the same cluster. The second approach further generalizes the stripped clustering approach by grouping together stripped structures having small pairwise edit distance. The authors demonstrate that using stripped XPaths leads to a ten-fold reduction in the number of generated clusters as compared to using non-stripped XPaths, while retaining more than 98% of the extractions. Adding a relaxed matching with an edit distance of 3 increases the extraction coverage of the exact matching approach by 25% (as more structure variations can be captured by the templates), while only reducing the extraction success rate by 2%. This confirms the feasibility of using the HTML email structure for templatization and affirms the importance of approximate – rather than exact – matching in the template clustering phase.

4.3.2 Threading

Unlike in personal communications, where the notion of threads is prevalent, it is not available for machine-generated communication, as it is generally one-sided (e.g., a user is unlikely to respond to a receipt about their purchase). Therefore, Ailon *et al.* (2013) propose the idea of causal threading of sequences of machine-generated emails. Their

approach joins a series of interactions from a bulk sender into a cohesive thread. See Figure 4.4 as an example.

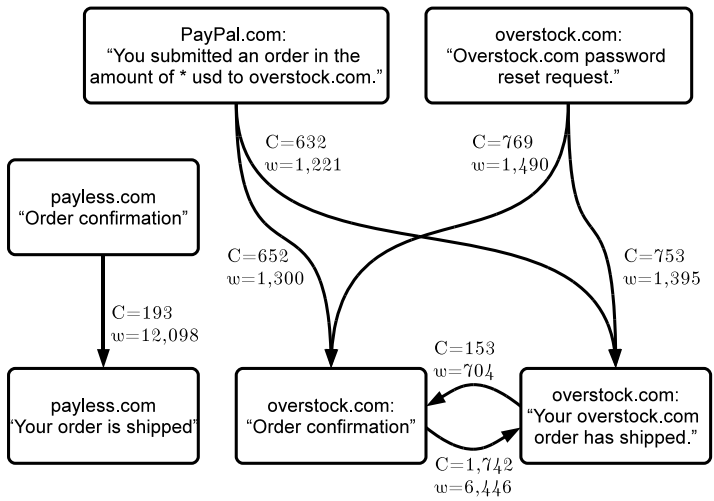


Figure 4.4: A snippet of the learned causal graph from Ailon *et al.* (2013). While these message are not a part of the same thread, the proposed algorithm can correctly infer common sense patterns across vendors, e.g., that order confirmations precede order shipments.

Ailon *et al.* (2013) propose both a causal inference algorithm, as well as a predictive algorithm, to address the problem of unseen threads. The casual inference algorithm constructs a directed causality graph G, where an edge $w(\tau, \tau')$ indicates a causal connection $\tau' \to \tau$ between templates. To infer causality, the prior probability of observing τ in a time window δ is compared to the probability of observing τ in the same time window following an appearance of τ'. Assuming that the number of appearances of a template τ in a time window has a Poisson distribution with a parameter $\lambda(\tau)$, this is formally defined as

$$w(\tau, \tau') = \frac{C(\tau, \tau')/C(\tau)}{1 - \exp(-\lambda(\tau)\delta)},$$

where C are either conditional or unconditional template appearance counts. Only edges with $w(\tau, \tau') > 1$ are retained and the graph G is further pruned by restricting its maximum out-degree.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

The causality graph G can be further augmented by other features defined on the ordered template pairs. Specifically, given two emails e_1, e_2 such that $w(\tau(e_2), \tau'(e_1)) \in G$, Ailon *et al.* (2013) use features derived from the temporal proximity of their templates, matches of $var(e_1)$ and $var(e_2)$, their respective variable lists, and template periodicity features.

Template threading has led to some follow-up work on predicting activity in a certain thread. For instance, Gamzu *et al.* (2015) investigated whether it is possible to predict the arrival of future emails in the same thread. Di Castro *et al.* (2016a) explored what actions users might take on emails in the thread. We cover this work in more detail in Section 6.3.

4.3.3 Information Extraction

Thus far in this section, we only discussed the template structures τ, but not the list of variables $var(e)$ that can be extracted by applying these structures to an individual email e. Such extractions can be used in a variety of search and discovery scenarios. As an example, they can be naturally surfaced in search through knowledge panels that directly present answers extracted from individual emails (see Figure 3.4 for an example), or presented along with the emails to facilitate locating the relevant information. It can also be presented in applications that involve personal assistance (Figure 4.5).

Sheng *et al.* (2018) provide a detailed discussion of a large-scale extraction engine from templated email data, with a focus on three email categories or verticals: *Bills*, *Hotels* and *Offers*. Each of these verticals has different extraction patterns, but the general extraction flow looks as follows:

1. Identify the email template.
2. Identify the vertical of the template (e.g., *Bills*).
3. Identify the relevant fields from the email for extraction, based on the vertical (e.g., due date for *Bills*, or check-in date for *Hotels*).

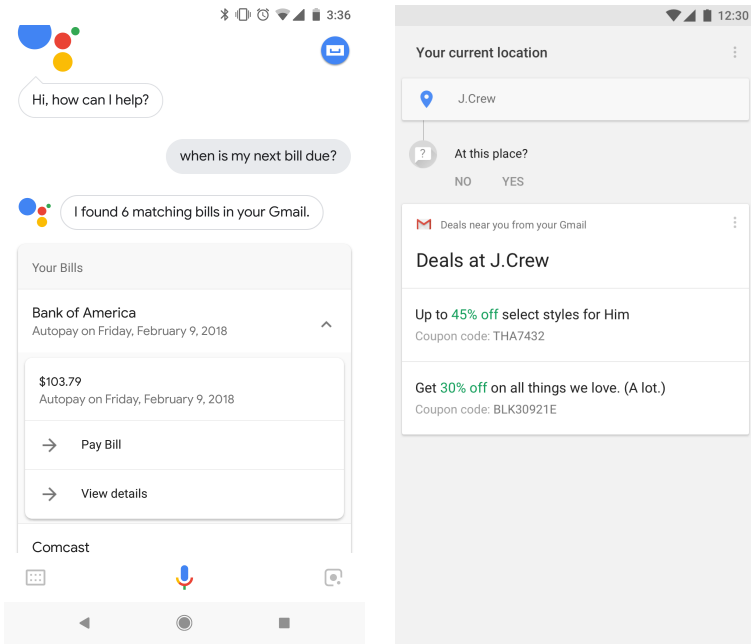


Figure 4.5: Google Assistant responding to a user query for their recent bills, and proactively displaying deals extracted from Gmail when the user enters the relevant store (Sheng *et al.*, 2018)

The first stage of the flow (email template induction) has been discussed in detail in Section 4.3.1. For the last two stages (vertical detection and field extraction), a supervised classification approach is used. The positive examples for the classifiers comes from various sources: (a) microdata – machine-readable semantic annotations in HTML emails,[2] (b) manual, hand-crafted annotations per sender, (c) parsers based on rules and regular expressions. The negative examples are downsampled from the rest of the email corpus. Features for the vertical and field classifiers are presented in Table 4.2.

One important practical consideration made by Sheng *et al.* (2018) is the fact that extractions "require high precision, so the improvement steps usually entail increasing coverage of extractions while maintaining high precision". This is due to the fact that, as shown in Figure 4.5

[2] https://www.w3.org/TR/microdata/

Table 4.2: Features used by Sheng *et al.* (2018) for (a) classification of structured email templates into predefined verticals (e.g., *Bills*), and (b) classification of template fields into predefined type (e.g., *due date*).

Feature Name	Description
subject-text	Words in the subject line
sender-text	Tokens in the sender field
top-text	Top 150 words in the body
strong-text	Text marked header, title, bold etc.
alt-text	Alt-text supplied for image content
footer-text	Last 100 words in the body
html-tag-count	Number of HTML tags in the body
text-token-count	Number of text tokens in the body
link-tag-count	Number of link tags
image-tag-count	Number of image tags
script-tag-count	Number of script tags
table-tag-count	Number of table tags
datetime-count	Number of candidate date-time spans
salient-entities	Top entity IDs

(a) Vertical classifier features

Feature Name	Description
{5/10/20}-w-before	5/10/20 words before the candidate span
{5/10/20}-w-after	5/10/20 words after the candidate span
field-text	Contents of the candidate span
doc-index	Position of the field in the document (0-1)
candidate-index	Positional rank relative to all candidates

(b) Field classifier features

extraction are likely to be surfaced in assistive products, and therefore, there is low tolerance for incorrect information. Therefore, the system developers are likely to start with very conservative extraction rules, and iteratively improve coverage, while maintaining high precision.

Similar to the work by Sheng *et al.* (2018), Di Castro *et al.* (2018) describe a production-ready information extraction system for the *Travel* vertical. Unlike Sheng *et al.* (2018), who rely on a multi-vertical deep learning classifier with a unified set of simple features, Di Castro *et al.* (2018) utilize an automatic rule extraction system tailored towards a specific vertical.

Gupta *et al.* (2019b) describe an information extraction system for structured emails in the *Flight Reservation* vertical. Similar to prior

The version of record is available at: http://dx.doi.org/10.1561/1500000069

work, their system uses a mixture of handcrafted wrappers, rule-based wrapper induction, and machine learning to ensure high precision of the resulting extractions. In addition, Gupta *et al.* (2019b), discuss the adaptation to non-English languages with minimal cost. They consider several alternatives; their best solution translates unlabeled non-English email into English, and uses an English language embedding to represent sequence of words in the original email. Then, a CNN classifier (trained on the English email data) is applied to this translation embedding, achieving both high precision and recall on the source language.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

5

Query Understanding

Query processing and understanding techniques have an extensive track record of success in web search. Search logs, which contain user queries and the associated interactions with the retrieved content (clicks, reformulations, long views, etc.) have long been considered a valuable source of information that significantly improved the efficacy of web search engines in understanding user queries. Our readers may refer to Silvestri (2010) for a good overview of this topic.

In web search engines, features like query auto-completion, query spell-correction, and related query suggestion are all considered an expected norm. In that regard, email search still significantly lags behind web search at the time of this writing. Consider Figure 5.1, which demonstrates how the query *"amazin"* is interpreted by a major web mail service.[1] Statistically, it is likely to be a common misspelling of the online retailer *Amazon*, as any web search engine will helpfully indicate. However, email search treats *"amazin"* as a sub-string match for the word *amazing*, resulting in clearly non-relevant results.

This example indicates an important distinction between email and web search. Unlike in web search, email search requires an understand-

[1] Similar behavior was demonstrated in the other web mail services as well.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

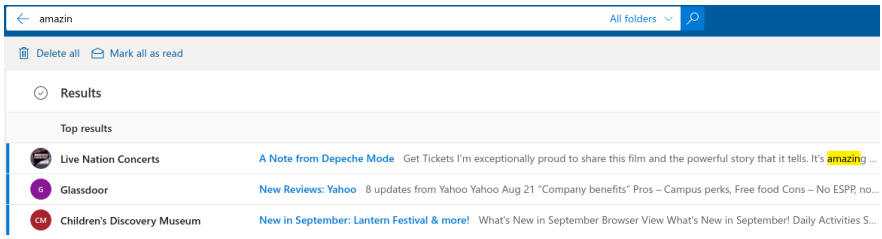


Figure 5.1: Results for query *"amazin"*, as retrieved by a popular mail search engine. Note that results containing "Amazon" are present in the mailbox, but are not retrieved.

ing of the personal mailbox, in addition to the log-based interactions aggregated across a large base of email users. Understanding that (a) many email service users often search for Amazon orders, and (b) the current user is an avid Amazon shopper, would significantly reduce the time spent finding relevant personal content for a potentially misspelled query like *"amazin"*.

Therefore, in this chapter, we provide examples of current research on three common query understanding tasks that can help to improve the email search experience: *query auto-completion*, *query spelling correction* and *query expansion*. To improve these tasks, the researchers may potentially use the following types of email data:

- *Personal mailboxes* – help in identifying any particular misspellings or phrasings that are unique to the user (e.g., informal language, or nicknames).
- *Personal query logs* – email search logs from an individual user, which can help to identify recurring searches, and unique search intents, patterns, and interests.
- *Global query logs* – an anonymized union of all personal query logs, which are useful for leveraging the universal email search trends, similar to web search logs (Silvestri, 2010).

The techniques that we describe in this chapter heavily rely on one (or some) of these data sources to attain better email-specific query understanding.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

5.1 Query Auto-Completion

Unlike in web search, where we can fully rely on the "wisdom of the crowds" to auto-complete queries, email search needs a more nuanced understanding of the individual mailbox to improve query completion. For instance, even if completing *"amaz"* to *"amazon"* is a highly common pattern in the global query logs, it only makes sense to suggest it if the user indeed received messages from Amazon. In addition, for some very personal queries (e.g., sender names and aliases), drawing auto-completions from the global query logs is not a viable option. On the other hand, personal query logs may be sparse for users who do not actively engage in email search.

Therefore, in email search, we need to consider whether query suggestions based on a global search log actually match the personal mailboxes. To this end, Horovitz *et al.* (2017) propose an approach that combines mailbox-based and global log-based completions.

For mailbox-based completion, Horovitz *et al.* (2017) first extract candidates using unigrams and bigrams extracted from the user mailbox. For each candidate, features are extracted based on:

- **tf-idf score**, where *tf* is computed based on the candidate occurrence in the user mailbox, and *idf* is its occurrence across all mailboxes.
- **message-level features**, which are based on the type of messages that the candidates most frequently appear in: read, flagged, forwarded, etc. In addition, candidates that appear mostly in older messages are penalized.
- **field-level features**, which are defined as field-specific (sender, recipient, cc, subject, and attachment) tf-idf candidate scores.

Horovitz *et al.* (2017) train an online variant of SVMRank (Crammer *et al.*, 2009) to learn the feature weights, using prefixes of queries resulting in clicks as a source of training data. They demonstrate that the resulting mailbox-based suggestions are substantially better than those purely based on global search logs, and that a linear combination of mailbox-

The version of record is available at: http://dx.doi.org/10.1561/1500000069

based and global log-based methods yields further improvements. These trends are consistent across prefix lengths and evaluation metrics.

In addition to better mailbox understanding, better user modeling also shows consistent improvements for query auto-completion in email search. As an example, Carmel *et al.* (2017b) show that demographic factors such as age, income, gender and state of residency can all improve log-based query suggestions. More recently, Foley *et al.* (2018) also demonstrate that semantic representations of the fine-grained user location can significantly improve query suggestion over a log-based baseline for very short prefixes. The most substantial gains ($\times$4.51 improvement over the MRR of the baseline method) were observed for *zero*-prefix queries, with the positive effects dissipating for 3+ character prefixes. These findings indicate that user location is highly predictive of query intent in email search, and should be considered as an important feature for mobile email search clients.

Another interesting research direction is personal log-based query suggestion, which was found useful in web search (Mei and Church, 2008), but has not been tackled in the context of email search. It is still unclear whether such log-based personalization can further improve performance if mailbox-based query suggestions have already been deployed, especially since some users do not search their personal email as frequently as the web.

5.2 Query Spelling Correction

Query spelling correction is standard for web search engines today, however, with the exception of Gmail, none of the major web clients support this functionality (Bhole and Udupa, 2015). Similarly to query auto-completion, query spell-correction in email search can be of two flavors. The first flavor is global log-based, and its implementation will be mostly similar to web search (and, in large part, web-based query spelling corrections may be reused in this paradigm).

The second flavor, which we focus on in this section, is based on personal mailboxes, and addresses the personal and context-specific nature of email search. Bhole and Udupa (2015) propose a machine-learning based algorithm that:

The version of record is available at: http://dx.doi.org/10.1561/1500000069

1. generates a set of 1,000 candidates from a user's email data that have a small edit distance from the query and high *idf* score in user's mailbox;

2. scores these candidates using a linear model containing features based on lexical similarity to the query, state of the mailbox and recent user activity.

As contact names and email addresses are often issued as queries in email search (Ai *et al.*, 2017), Ramarao *et al.* (2016) propose a hashing-based people search algorithm to specifically handle misspelled or mis-remembered names. The algorithm learns hash functions that map similar-sounding names to similar binary code words in a language-independent space (e.g., *Lakshmi* and *Laxmi* would map to the same code).

Gupta *et al.* (2019a) further explore the idea of personalized spelling corrections, specifically focusing on the problem of efficiently serving such models with low latency at the very large scale required by the major web mail services. In particular, Gupta *et al.* (2019a) focus on an efficient computation of Levenshtein distance through early termination for candidates that are unlikely to provide good corrections. They also discuss techniques for generating highly compact personalized lexicons with real-time updates, which are generated using email titles and contacts.

Gupta *et al.* (2019a) report that personalized spelling corrections can affect up to 3% of Gmail query traffic and, for the affected queries, result in double-digit improvements across metrics and languages. These results demonstrate the importance of mailbox-based spelling correction in email search, re-affirming the results on mailbox-based query auto-completion reported in Section 5.1.

5.3 Query Expansion

Similarly to the two previous tasks, query expansion techniques in email search fall into two major categories: log-based and mailbox-based. As personal query logs in email are often highly sparse, global and non-personalized log-based techniques are used.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

Email queries are generally very short – 1.5 terms per query as compared to 3 terms per query in web search (Kuzi *et al.*, 2017). Thus, query expansion is important for improving retrieval, especially in terms of recall which constitutes a major problem in email search, as the underlying corpus (a user's mailbox) is relatively small. Moreover, studies show that reusing expansion terms mined from web search logs may be helpful, but not sufficient (Li *et al.*, 2019a).

Kuzi *et al.* (2017) introduce three techniques for query expansion in email search: a translation model based on global query logs, an embedding model based on the user's mailbox, and a pseudo-relevance feedback model, which constructs the expansion terms from the retrieved search results. The experimental results demonstrate that the global translation model tends to be the most effective among the three techniques across multiple baselines and collections. In some cases, a linear interpolation of the three approaches may further improve the results.

Li *et al.* (2019a) further explore the application of global email query logs for the problem of query expansion. In particular, they address the problem of information sparsity through incorporating a multiple-view embedding approach. Unlike the translation model proposed by Kuzi *et al.* (2017) that "translates" query terms into clicked document subjects, Li *et al.* (2019a) consider multiple views of the data including clicks, query sessions and user distribution and incorporate each view into a separate feed forward neural network. Each network learns an embedding based on "similarity" and "context" tasks, which are customized per each view. Table 5.1 summarizes the views and the tasks being used.

At a final stage, the candidate synonyms from each view are filtered through label propagation on a bipartite graph between query and document n-grams with click-weighted edges, and then merged together through a learning-to-rank framework. Offline evaluation suggests significant improvement over a variety of baselines, including individual views, as well as synonym generation based on DESM, a publicly available embedding model based on the Bing query corpus (Mitra *et al.*, 2016). Online experiments using Gmail search also show statistically significant improvements over a system that uses synonyms developed for Google web search. This further validates the importance of employing

The version of record is available at: http://dx.doi.org/10.1561/1500000069

Table 5.1: A summary of the data views used by Li *et al.* (2019a) for learning similarity and context embeddings for generating candidate synonyms for query expansion.

View	Similarity	Context
Click	Subject n-gram is clicked for a query n-gram	Embed a query n-gram with the clicked subject n-grams
Query session	Two query n-grams are in the same search session	Embed a query n-gram with query n-grams in the same session
User distribution	User issued a query n-gram	Embed a query n-gram with the users who issued this query n-gram

specialized data sources and techniques for the query understanding tasks in email search.

Similarly to the case of both query auto-completion and spelling correction, thus far personal query logs have not been used for query expansion in published work. The sparsity of personal query logs and their private nature have limited their utility. A successful incorporation of personal query logs in query understanding for email search is an interesting open research challenge.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

6

Beyond Search: Intelligent Task Assistance

Thus far in this survey, we have discussed the scientific advances that can lead to improvements in search over email and other types of personal content (files, calendar entries, etc.). Search is a major assistive feature that enables easier and faster task completion for millions of users. However, with the recent advances in machine learning, and especially the rise of deep learning, other modes of assistance have become more commonly used in email and other personal content storage systems. Some recent assistive features available in personal content management systems include, among other, file recommendation in Google Drive (Tata *et al.*, 2017), Smart Compose feature in Gmail (Wu, 2018), Suggested Reminders in Cortana (Graus *et al.*, 2016), or Grammarly writing assistance tools.[1]

Therefore, in this chapter, we go beyond search, and focus on other assistance modes that facilitate personal content discovery and creation. In particular, we focus on recent research on personal content recommendation (Section 6.1), cross-platform assistance (Section 6.2), activity prediction (Section 6.3) and assisted composition (Section 6.4). It is important to note that while our survey mainly focuses on email content,

[1] https://www.grammarly.com/

The version of record is available at: http://dx.doi.org/10.1561/1500000069

in this chapter we often go beyond the boundaries of the mailbox, as many of these assistive features aim to boost productivity by bridging between the various types of personal content.

6.1 Personal Content Recommendation

There is a large and growing body of work on content recommendation in domains like e-commerce (Linden *et al.*, 2003), streaming services (Bell and Koren, 2007) and social media (Rogers, 2016). In all of these domains, the recommendations are done over public corpora, such that the recommended items are shared across users. Similarly to the case of search, content recommendation for private content has several important distinctions from content recommendations in these public domains.

First, the documents are private or shared across a few users, so standard collaborative filtering techniques are not applicable. Second, as discussed in previous chapters, private data poses challenges in developing machine learning models while respecting user privacy. Third, since traditional private content storage systems do not incorporate recommendations, designing effective recommendation user interfaces is an important new challenge.

Tata *et al.* (2017) examine these three challenges in the context of Quick Access – a document recommendation system in Google Drive. As Drive documents are access-controlled and are generally not shared widely among all Drive users, the authors eschew the collaborative filtering approaches in favor of a binary classification approach. First, for each user, a candidate *working set* is determined, based on user activity within the last 60 days. Then, each of the documents in the working set is scored using a deep network. The scoring is formulated as a binary classification. For a given scenario, a single positive example is generated (an opened document), as well as a random sample of negative examples selected from the working set. Tata *et al.* (2017) use a deep neural network, incorporating features reflecting various document properties (e.g., mime type), event types (opens, edits, comments) and event client platforms (Windows, Mac, Android, iOS). The features are encoded both as sparse fixed-width vectors, as well as histograms.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

Similarly to research on email search, Tata *et al.* (2017) discuss privacy policy as an important concern while building the system. They discuss code peer-review enforcement, work with anonymized and aggregated summary statistics, data k-anonymization, and debugging the models with data donated by the team members and colleagues as important pre-requisites for successfully building machine learning systems over private data.

Interestingly, Tata *et al.* (2017) also examine the effect of different user interface (UI) variations on the system performance. These variations include integration with zero-state search, pop-up suggestions, thumbnail integration and UI rendering latency. The authors demonstrate that these variations play a crucial role in user experience and live metrics.

Xu *et al.* (2020) further explore the role of *explanations* for personal content recommendation, using the Recommended Document Pane of Microsoft Office 365 as an experimentation platform. They find that the type of provided explanations plays a big role in how users interact with the documents suggested by the platform. For instance, a document recommendation decorated with the explanation "Alice commented on this file" significantly increases the click probability, particularly if the user receiving the recommendation authored that document. As another example, for documents that have not been opened for a while, the explanation "Bob shared this file with you" can help users to better recognize the file and reduce time to file open. On the other hand, some explanations like "You recently opened this" should be avoided, as they convey little information to the users, and do not help in their interactions with the suggested files.

Overall, the research above indicates the importance of user interface design when building personal content recommendation systems. Further exploration of this area is an interesting direction for future research.

6.2 Cross-Platform Assistance

While Quick Access is an example of a system where all the recommendations come from the same source, an important function of private content recommendation is breaking the barriers between the various

The version of record is available at: http://dx.doi.org/10.1561/1500000069

personal content silos. As an example, an average office worker uses the following on a daily basis: email, cloud file storage, calendar application, one (or more) messaging application(s), and a video conferencing system. Search and discovery across these different platforms is an important research challenge.

As an example, Van Gysel *et al.* (2017) address the problem of attachment recommendation – suggesting a relevant document attachment in the context of the current conversation thread. They cast the problem as one of standard information retrieval, by formulating a query from the conversation context, and then retrieving the attachable items in response to this query. In order to formulate a query, Van Gysel *et al.* (2017) first propose a heuristic that generates k candidate terms for each $< request, response >$ pair. The powerset of the k candidate terms is considered as all possible queries that could have retrieved the attachment to the *response* message. The queries are then scored by their ability to retrieve the relevant attachment (as measured by the reciprocal rank). These are then used to generate training data for a convolutional neural network that assigns a weight to each term. Experiments over the public Avocado dataset (see Section 2.5.3) as well as a set of proprietary enterprise emails demonstrate the efficacy of the proposed approach.

In another example, Zhao *et al.* (2018) propose CAPERS: a Calendar-Aware Proactive Email Recommender System. CAPERS proactively suggests emails in response to the context of the user's meeting schedule. As a motivation, in an internal survey, Zhao *et al.* (2018) find that 68.4% of the 592 participants prepare for meetings by accessing email, which suggests that an accurate proactive email discovery may save user time. Similarly to Van Gysel *et al.* (2017), Zhao *et al.* (2018) find that treating this problem as a query formulation process that extracts keywords from the meeting subject and body, followed by a retrieval and a ranking stage attains the best results, achieving over 80% as measured by the *nDCG* metric at the top ranks.

There are numerous other examples that indicate that cross-platform content consumption is an important and practically useful research direction. This is especially pertinent in the era of mobile devices, where limited screen surface makes switching among different personal content

The version of record is available at: http://dx.doi.org/10.1561/1500000069

platforms a hindrance to user productivity. There is a growing body of literature that addresses mobile access to personal content: Chen *et al.* (2019a) demonstrate a method and an interface for making app suggestions in the context of personal messaging; Swaminathan *et al.* (2017) introduce a conversational interface that allows access to email information through a wearable device; Kokkalis *et al.* (2013) convert a mailbox to a stream of tasks that can be easily triaged on a small screen (see Figure 6.1); Graus *et al.* (2016) tackle the problem of automatically scheduling timely reminders for such tasks.

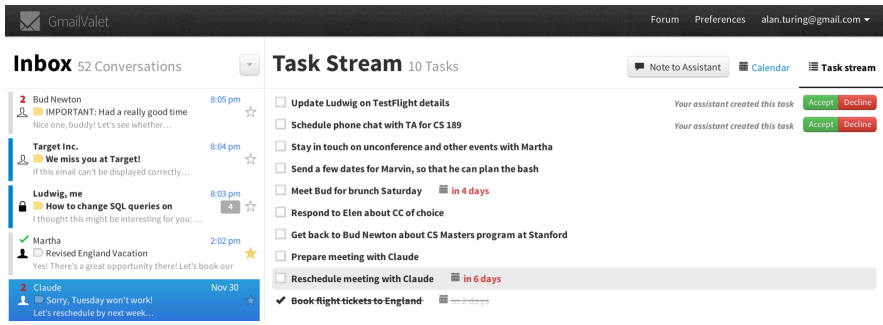


Figure 6.1: Email Valet (Kokkalis *et al.*, 2013) transforms a cluttered mailbox into a succinct, mobile-friendly stream of individual tasks.

6.3 Activity Prediction

As billions of actions are being taken by email users each day, analyzing this aggregated activity provides a unique opportunity for reducing the user information overload. To achieve this goal, researchers have been developing large-scale machine learning algorithms that take as an input activity patterns both from the sender and the recipient standpoints, and attempt to predict future activity.

In one of the earliest works on this subject, Dabbish *et al.* (2005) develop a model that attempts to predict how likely a user is to perceive an email as important or reply to it. The model is based on the survey responses from 124 email users in an academic institution. Dabbish *et al.* (2005) show that emails containing an action, information or scheduling request, status update, or reminder message content were

The version of record is available at: http://dx.doi.org/10.1561/1500000069

more likely to be deemed important. Message importance, in turn, significantly increased the likelihood of email reply. Information requests are additionally found to be highly predictive of email reply regardless of email importance. On the other hand, an increased number of recipients significantly reduces the likelihood of an email reply action.

Aberdeen *et al.* (2010) take a large-scale machine learning approach to predict the priority of an email, i.e., how likely is the user to act on an email. They formulate the prediction problem as

$$p = P(a \in A, t \in (T_{min}, T_{max})|\mathbf{f}, s), \tag{6.1}$$

where A is the set of actions denoting email importance (open, reply, manual corrections), t is the delay between delivery and action a, $\mathbf{f}$ is the feature vector, and s indicates that the user actually saw the email. According to Aberdeen *et al.* (2010) hundreds of features are used from the following main categories:

- *social* - based on the degree of interaction between sender and recipient
- *content* - headers and recent terms that are highly correlated with user action (or lack thereof)
- *thread* - user interactions with the thread thus far
- *label* – user-applied message labels.

For scalability purposes, a linear logistic regression model is used, and two models are trained – a local (i.e., user-specific) model and a global (i.e., at the user population level) model. The final prediction is a sum of the local and the global model, as the global model provides back-off for cases where there is a dearth of data for the local model.

In later research Di Castro *et al.* (2016a) revisit the task of action prediction, with more fine-grained predictions focusing on four action types in response to an email: *read*, *reply*, *delete*, and *delete-without-read*. They leverage the massive scale of a commercial web email service (Yahoo! Mail) for their analysis. Similarly to Aberdeen *et al.* (2010), Di Castro *et al.* (2016a) find that both local and global features are

The version of record is available at: http://dx.doi.org/10.1561/1500000069

useful for this task. In addition, they propose "horizontal" regularization for low-activity users that takes the general form

$$\mathbb{E}_{x,y}[Loss(w, x, y)] + \lambda_n ||w - w'||, \tag{6.2}$$

where w is the local model weights vector, λ is a tunable parameter depending on the number of user actions n, and w' is the weight vector for either an average user or a user belonging to the same latent class (found by k-means clustering). For preserving privacy, the features for each example x are not based on the email content, but only rely on aggregated action counts, either per user or per class. Interestingly, this horizontal regularization approach significantly improves performance for read and reply predictions, but not for delete and delete-without-read predictions, indicating that the latter actions may be less generalizable in nature.

Gamzu *et al.* (2018) examine another user action, that is specific to bulk mail – unsubscription. They find that a logistic regression model that combines personal user activity features with those from the global user population, as well as from relevant demographic groups demonstrates the best performance, reaffirming the importance of "horizontal" or cross-user learning for activity prediction noted by previous work.

Gamzu *et al.* (2018) also argue for the importance of providing the users with a convenient interface for unsubscription from email traffic that may be of low interest to them. In an online experiment, where an unsubscription dialogue (see Figure 6.2) was shown to Yahoo! Mail users, 34.2% of the dialogues were engaged with to unsubscribe. Gamzu *et al.* (2018) report that these numbers are 8 times larger than the number of users who actively performed unsubscriptions by clicking links within email bodies. These high engagement number demonstrate that proactive assistance interfaces are crucial in combating information overload.

In addition to predicting *user activity* in response to a particular email message, researchers have also examined predicting future *sender activity* on an email conversation or their communication with a given recipient. This has been explored in-depth specifically for bulk machine-generated emails (Ailon *et al.*, 2013; Gamzu *et al.*, 2015; Zhang *et al.*, 2017).

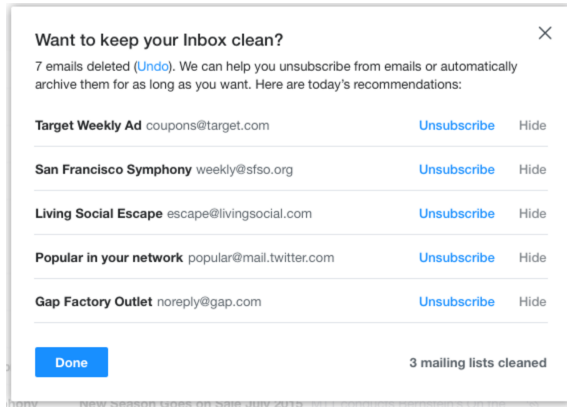


Figure 6.2: Unsubscribe dialog to facilitate easier unsubscription (as shown in (Gamzu *et al.*, 2018)).

As an example, Zhang *et al.* (2017) use an anonymized dataset consisting of 2.5 million machine-generated emails from more than a hundred thousand users to predict one of 17 categorical labels for the next received email (e.g. restaurant reservation, online purchase, job listing, etc.). They examine the performance of Markov chains, multilayer perceptron, and long short-term memory (LSTM) for this task (shown in Figure 6.3). Neural based models that effectively encode the various time-based features (e.g., day of the week, period of the months and the length of the time gap between two consecutive messages) demonstrate superior performance to Markov chains. The high *MRR* values in Figure 6.3, especially for the LSTM model, indicate the feasibility of assistive technologies that take into account the arrival of future emails.

6.4 Assisted Composition

Thus far, we have discussed the assistive technologies that make it easier for users to engage with the existing personal content. In this section, we focus on a case for assistive composition – technologies that facilitate more effective personal content creation. Such assistive technologies generally serve two goals. First, they take care of mundane, repetitive

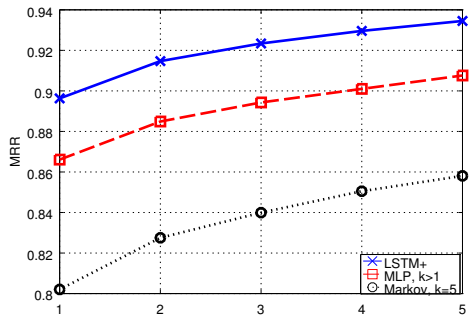


Figure 6.3: The performance gap between the Markov chains and the neural methods (MLP and LSTM) for the task of category prediction for future emails. (Zhang *et al.*, 2017).

content creation tasks such as generating canned email replies (Kannan *et al.*, 2016) or suggesting likely email recipients based on context and past history (Carvalho and Cohen, 2008). Second, they can actually improve the quality and the readability of the generated content by using well-structured grammatical as-you-type auto-completions (Wu, 2018), suggesting relevant email attachments (Van Gysel *et al.*, 2017) or improving the readability of the composed emails through better language clarity or grammar (Grammarly, 2018).

Overall, methods and systems for assisted composition can be broken down into the following categories (in an increasing difficulty order): binary prediction, item ranking, and content generation, each of which we discuss next.

6.4.1 Binary Prediction

The earliest techniques for assisted composition focus on binary classification tasks that help to prevent unintentional message composition errors. As examples, Perronnin (2009) proposes a missing attachment detection system based on the statistical language model of the composed email; Carvalho and Cohen (2007) focus on preventing information leaks by building a classifier for unintended email recipients using both email content and the sender social network information.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

In some cases, binary classification can also be combined with an auxiliary task that focuses on a particular part of the message. For instance, in the context of identifying requests for action, Lampert *et al.* (2010) draw a distinction between *message-level identification* – whether an email contains a request – and *utterance-level identification* – determining where in the message the request is expressed, and how to respond to it. In this case, the former can be formulated as a binary classification, while for the latter a more general information extraction or content generation model needs to be employed.

6.4.2 Item Ranking

Going one step forward beyond binary classification, researchers also proposed ranking models for assistive composition. Some examples include ranking likely email recipients (Carvalho and Cohen, 2008), or recommending which documents should be attached to an email (Van Gysel *et al.*, 2017). Email folder suggestion (Segal and Kephart, 2000) or reply suggestion (Sordoni *et al.*, 2015) where all the suggestions are known in advance, can be modeled as item ranking models.

In general for evaluating these models ranking metrics should be considered, in lieu of classification metrics like precision or recall. In the cases where we expect to have exactly one correct suggestion, *MRR* – mean reciprocal rank of the first relevant system suggestion – could be used as a success criteria. For cases where there are multiple potentially relevant suggestions, *MAP* (mean average precision) or *DCG* (discounted cumulative gain) could be used.

6.4.3 Content Generation

Most recently, commercial systems started to provide more advanced assistive composition features that are directly geared towards content generation. The development of such models is generally afforded by advances in deep learning. These advances enable training models from massive amounts of anonymized user generated content. For instance, Kannan *et al.* (2016) in their work on the Smart Reply feature in Gmail use a training set of 238 million email messages.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

The Smart Reply system suggests brief replies to email messages containing simple information requests. The feature is especially helpful on mobile devices, where tapping on a suggested reply option can significantly reduce time to reply: Kannan *et al.* (2016) indicate that at the time of writing, 10% of mobile replies in Gmail Inbox were "composed with assistance from the Smart Reply system".[2]

As an interesting research problem, Kannan *et al.* (2016) discuss enforcing response diversity to ensure the usefulness of the smart replies. First, the redundant responses are normalized into a single canonical intent (e.g., *Yes, I can!* and *Sure, I can!*), and only one response per intent is shown to the user. Second, in order to provide real choice to the users, if none of the top three responses are negative, the third response is replaced with a negative one.

The Smart Compose system (Wu, 2018; Chen *et al.*, 2019b) takes the content generation techniques one step further by learning to interactively offer sentence completions as the users type. To achieve this, the authors combine a bag-of-words neural language model (Bengio *et al.*, 2003) with a Recurrent Neural Network based language model (RNN-LM) (Mikolov *et al.*, 2010), using the averaged word embeddings of the subject and the previous emails as an input to RNN-LM for the next word prediction.

The creators of the Smart Reply and the Smart Compose features raise several interesting practical considerations for deploying assistive composition at scale. For instance, Wu (2018) reports that using Tensor Processing Units (TPUs) instead of standard CPUs for inference decreases serving latency from hundreds to tens of milliseconds, while also greatly increasing request throughput.

[2] https://en.wikipedia.org/wiki/Inbox_by_Gmail

The version of record is available at: http://dx.doi.org/10.1561/1500000069

7

Managing and Learning from User Data

The ability to effectively leverage user data is crucial to the development of any large scale search or recommendation system. Researchers need such data to empirically quantify and compare the performance of their systems, to perform labeling and annotation, and to use it for training machine learning algorithms. However, unlike in some other textual data processing settings, where human assessors and researchers can access public data (e.g., web pages), in the email setting, the data is in the private domain. The number of publicly available annotated resources is very limited,[1] and not all annotation use cases can be supported by these resources. As a result, when developing mailbox processing algorithms, there is a need for leveraging user data for annotation and training while respecting the privacy constraints of an email corpus.

In addition, when training learning algorithms with user interaction data such as clicks, without direct access to human labeling or email corpus, it is important to be mindful of the inherent biases and sparsity of such interactions. Clicks are not always indicative of relevance or utility, and each user will have access only to their own private email corpus, limiting the generalizability of their interaction data.

[1] See Section 2.5.3 for an overview of the existing public datasets.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

To address these important issues, we dedicate this chapter to the management of users' personal content and interaction data in the context of email search and discovery. We start by discussing best practices for privacy-preserving processing of email content in Section 7.1. In Section 7.2, we describe techniques for click bias correction to ensure optimal training of click-based learning-to-rank algorithm. We conclude in Section 7.3 that discusses techniques for click data aggregation.

7.1 Data Privacy

Data privacy presents multiple model development challenges. For instance, researchers cannot access unredacted email content in order to inspect and debug their models. They also cannot evaluate the performance of a model on an individual user, or a predefined group of users. Finally, models directly trained with unredacted data may leak private information. To address these issues, Kannan *et al.* (2016) propose the following general privacy-preserving principles for data protection and reduction:

- *Encryption* All types of data are stored in an encrypted format, and the unredacted data cannot be directly accessed by the model developers.
- *Aggregation* Model developers can only inspect aggregated model statistics that cannot be associated with a particular user.
- *Frequent Words* If the inspection of a particular text snippet is required, any user identifications are removed and only frequent words (i.e., words that occur across multiple users) are retained.

To comply with these principles, the data processed by email search and discovery algorithms should be *anonymized*, i.e., the data should not be related to an identified or identifiable individual person. The anonymization process should render all personal data anonymous in such a manner that this individual is no longer identifiable.[2]

[2]Based on the General Data Protection Regulation by the European Parliament and Council of the European Union (2016).

The version of record is available at: http://dx.doi.org/10.1561/1500000069

In this section, we discuss three broad classes of data anonymization methods: data de-identification (Section 7.1.1), k-anonymization (Section 7.1.2), and differential privacy (Section 7.1.3), and provide concrete examples of how they can be applied to email search and discovery. In some limited special cases, users may also allow some of their data to be directly annotated with no or limited anonymization. This is discussed in Section 7.1.4.

7.1.1 Data De-identification

The simplest data anonymization technique is *data de-identification*, which attempts to remove any personally identifiable information from the stored content, prior to applying any modeling techniques to it. Data de-identification is often used in practical applications, e.g., in medical data processing to ensure the privacy of patients records (Uzuner *et al.*, 2007).

In particular, HIPAA regulation by the US Department of Health and Human Services (2012) is designed as a standard for de-identification of protected health information, and provides implementation specifications for this standard. The HIPAA regulation suggests the *Safe Harbor* data identification method, which requires removal of the 18 types of personal identifiers listed in Table 7.1 from any released medical record. While the Safe Harbor method was developed primarily for medical record anonymization, it is clear that the personal identifiers it defines are applicable to email (or any other private) content as well.

Data de-identification is an important data pre-processing stage that should be applied to all types of private data. However, it is important to note that it is insuffient to guarantee data privacy on its own. Identifying all the private information in Table 7.1 may be an error prone process (Uzuner *et al.*, 2007), with no provable guarantees for removing all personal identifiers. Therefore, in the next two sections, we discuss two methods that provide additional statistical privacy guarantees: k-anonymity and differential privacy.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

Table 7.1: Personal identifiers that need to be removed in order to de-identify health information. According to the Safe Harbor method, as described by the US Department of Health and Human Services (2012) HIPAA regulation §164.514(b)(2).

(a) Names	
(b) All geographic subdivisions smaller than a state, or the initial three digits of the ZIP code, if the geographic unit formed by combining all ZIP codes with the same three initial digits contains more than 20,000 people	
(c) All elements of dates (except year) for dates that are directly related to an individual	
(d) Telephone numbers	(k) License numbers
(e) Fax numbers	(l) Vehicles identifiers and serial numbers
(f) Email addresses	(m) Device identifiers and serial numbers
(g) Social security numbers	(n) URLs
(h) Medical record numbers	(o) IP addresses
(i) Health plan beneficiary numbers	(p) Biometric identifiers
(j) Account numbers	(q) Full-face photographs
(r) Any other unique identifying number, characteristic, or code	

7.1.2 *k*-Anonymity

The formal principle of k-anonimity states that given a set of protected attributes (e.g., location, age, etc.), the data can be considered k-anonymous only if every combination of values of these attributes can be indistinctly matched to at least k individuals (Samarati and Sweeney, 1998).

Di Castro *et al.* (2016b) apply the principles of k-anonymity to performing annotations over anonymized email data. In particular, these k-anonymized annotations can be used to validate the templatization and extraction algorithms described in Section 4.3.

The email k-anonymization technique consists of three main stages: *grouping*, *masking* and *assignment*.

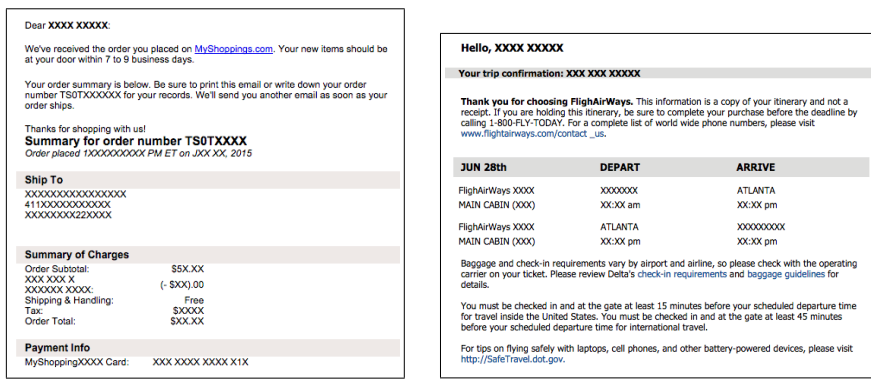
- *Grouping* In this stage, the messages are grouped by a *MailHash* algorithm. First an email is assigned to a unique hash, which is the MD5 hash of its DOM-tree signature. Then, the number of distinct users (each user identified by a unique recipient email address)

associated with each group is counted, such that any group that has less than k unique users associated with it is deleted.

- *Masking* For each of the remaining groups, the text is deleted from each of the DOM-tree entries, up to the point where all messages in the group are identical, and thus k-anonymization is preserved at the group level. The retained masked samples can be considered templates, while the deleted entries represent the template variables (see Section 4.3.1 for a formal email template definition).

- *Assignment* The grouping and masking stages guarantee that each resulting email template is k-anonymized. In the assignment stage, we are also guaranteeing that the assignment stage is k-anonymized by guaranteeing that no samples from a single user are viewed by the same human assessor. For this, the algorithm ensures that each user will be associated with *at most one* template assigned to the same assessor.

Figure 7.1 shows an example of the resulting email samples (receipt and flight itinerary) from this three-stage process.


Dear **XXXX XXXXX**:

We've received the order you placed on MyShoppings.com. Your new items should be at your door within 7 to 9 business days.

Your order summary is below. Be sure to print this email or write down your order number TS0TXXXXXX for your records. We'll send you another email as soon as your order ships.

Thanks for shopping with us!

Summary for order number TS0TXXXX

Order placed 1XXXXXXXXX PM ET on JXX XX, 2015

Ship To

XXXXXXXXXXXXXXXX
411XXXXXXXXXX
XXXXXXXX22XXXX

Summary of Charges

Order Subtotal:	$5X.XX
XXX XXX X	(- $XX).00
XXXXXX XXXX:	
Shipping & Handling:	Free
Tax:	$XXXX
Order Total:	$XX.XX

Payment Info

MyShoppingXXXX Card: XXX XXXX XXXX X1X

Hello, XXXX XXXXX

Your trip confirmation: XXX XXX XXXXX

Thank you for choosing FlighAirWays. This information is a copy of your itinerary and not a receipt. If you are holding this itinerary, be sure to complete your purchase before the deadline by calling 1-800-FLY-TODAY. For a complete list of world wide phone numbers, please visit www.flightairways.com/contact _us.

JUN 28th	DEPART	ARRIVE
FlighAirWays XXXX	XXXXXXX	ATLANTA
MAIN CABIN (XXX)	XX:XX am	XX:XX pm
FlighAirWays XXXX	ATLANTA	XXXXXXXXX
MAIN CABIN (XXX)	XX:XX pm	XX:XX pm

Baggage and check-in requirements vary by airport and airline, so please check with the operating carrier on your ticket. Please review Delta's check-in requirements and baggage guidelines for details.

You must be checked in and at the gate at least 15 minutes before your scheduled departure time for travel inside the United States. You must be checked in and at the gate at least 45 minutes before your scheduled departure time for international travel.

For tips on flying safely with laptops, cell phones, and other battery-powered devices, please visit http://SafeTravel.dot.gov.

Figure 7.1: An example of k-anonymized email sample of a shopping receipt and a flight itinerary (from Di Castro *et al.* (2016b)).

As email search queries are as likely to contain personal information as mailboxes (e.g., searching for a contact name is one of the most

The version of record is available at: http://dx.doi.org/10.1561/1500000069

common email search tasks (Ai *et al.*, 2017)), email query log analysis has similar privacy constraints to mailbox processing. In both cases, it is common to apply k-anonymization to query and email text, retaining only frequent terms co-occurring across multiple users.

As an example of k-anonymized log processing, Li *et al.* (2019b) detail their anonymization technique, wherein only queries and email subject lines are used, and only n-grams that appear in query logs of sufficiently many users are retained. Then – for both subject line and query – a small subset of these frequent n-grams is retained, without order information. Other query processing work discussed in Chapter 5 applies similar anonymization policies (Gupta *et al.*, 2019a; Foley *et al.*, 2018). User identifiers are either reported to be fully removed from the data (Foley *et al.*, 2018), or hashed using a non-invertible function (Carmel *et al.*, 2017b).

Such strict data protection requirements inevitably limit the expressiveness of the resulting models, preventing the usage of sequence learning algorithms such as CRFs, RNNs, LSTMs, or the more recently introduced Transformers (Vaswani *et al.*, 2017). This is due to the fact that in order to meet the privacy threshold, most k-anonymity based approaches only consider term unigrams. As a result, they do not retain term order information for queries, subject lines, or email contents. The applicability of sequence learning algorithms to obfuscated, k-anonymized datasets is an interesting direction for future work, and holds a big promise for improving the state-of-the-art of mailbox and query understanding techniques in email search.

7.1.3 Differential Privacy

The issue with k-anonymity is that it may become difficult on high-dimensional datasets (i.e., if the number of protected attributes is too large) (Aggarwal, 2005). In addition, in the case of datasets where the data has user associations, even if each individual attribute is k-anonymized, it can still be compromised by joining the data with external sources of information (Naranyanan and Shmatikov, 2008).

To address these issues, *differential privacy* was proposed (Dwork, 2008). Formally, a randomized algorithm $\mathcal{A}$ has an ϵ-differential privacy

The version of record is available at: http://dx.doi.org/10.1561/1500000069

guarantee if

$$\frac{P(\mathcal{A}(D) \in S)}{P(\mathcal{A}(D') \in S)} \leq \exp(\epsilon), \tag{7.1}$$

where D and D' are any two datasets differing in at most one element, and S is a possible outcome of the algorithm $\mathcal{A}$. Simply put, this definition provably ensures that adding or removing a single database item does not substantially change the output distribution of the algorithm $\mathcal{A}$.[3] In practice, differential privacy is usually implemented by adding small amounts of random noise to either the dataset D or the algorithm A, to ensure that single item changes will be masked by this noise.

As an example for a differential privacy application for email corpora, consider the case of Smart Compose (Chen *et al.*, 2019b) and Smart Reply (Kannan *et al.*, 2016), two examples of large language models trained on a text corpus comprising of the personal emails of millions of users. Smart Compose and Smart Reply are available to Gmail users, providing automatic sentence completions and reply suggestions, respectively.[4] Obviously, even though the training data for these language models may contain sensitive information about individual users, the models should never emit such data as suggestions. One simple solution is applying k-anonymization to the data, however, as discussed at the end of Section 7.1.2, this will severely limit their expressiveness and effectiveness due to the loss of word order information.

Alternatively, differential privacy does not require k-anonymization, and instead focuses on injecting noise into either the data itself, or the model training process, such that we will have provable guarantees that the model will not inadvertently memorize rare-but-sensitive items (e.g., personal identifiers). In this context, Carlini *et al.* (2019) demonstrate that applying differential privacy minimizes the risk of such *unintended memorization* that occurs when trained neural networks may expose the presence of secret or private data in the model (a.k.a. *canaries*). They propose a formal definition of these canaries and define an *exposure* evaluation metric

$$exposure_\theta(s[r]) = \log_2 |\mathcal{R}| - \log_2 rank_\theta(s[r]), \tag{7.2}$$

[3] For strong privacy guarantees, $\epsilon \in [0, 0.01]$ is recommended, although it can potentially be as large as $\ln(2)$, if there is more risk tolerance (Dwork, 2008).

[4] See Section 6.4 for more details on the implementation of these models.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

where $s[r]$ is a canary sequence chosen through randomness r, from some randomness space $\mathcal{R}$, and *rank* is the index of canary $s[r]$ in the list of all possibly-instantiated canaries, ordered by the empirical perplexity of the evaluated model θ.

Intuitively, $exposure_\theta(s[r])$ measures how much more likely a particular canary $s[r]$ is to be guessed by a model θ than any other random sequence. For instance, consider inserting a specific canary $s[r] =$*"Social security #523452345"* into a dataset, and training a language model over it. Then, one can compute the perplexity of this canary, and rank it with respect to all other possible canaries. The value of the *exposure* will range from $\log_2 |\mathcal{R}|$ (canary is ranked first) to 0 (its rank is indistinguishable from that of any other random sequence). Higher values of the *exposure* metric indicate model propensity to memorization and a higher risk of unintentional private data leakage.

Given a task of predicting the next character given a prior sequence of characters (akin to the Smart Compose use case), Carlini *et al.* (2019) discuss several standard techniques that can ameliorate the private data leakage risk, including weight regularization and data de-identification. However, they find that none of these techniques provide provable guarantees against memorization and data exposure. Instead, they turn to differential privacy, and empirically demonstrate that the differentially-private stochastic gradient descent algorithm (DP-SGD) by Abadi *et al.* (2016) is an effective defense that can almost completely prevent data exposure, albeit with some performance losses for the next character prediction task. An interesting finding is that even for large values of ϵ – which introduce vanishingly small amounts of noise to DP-SGD, and do not have a significant impact on task performance – the measured exposure becomes negligible.

7.1.4 Transparent Data Access

In some cases, the researchers may still need access to the full mailbox corpus without any anonymization applied, e.g., for the purposes of code debugging or better understanding of the model behavior. In such a case, the users should be aware of such access, understand what data is being shared and with whom, and voluntarily opt-in for being a

The version of record is available at: http://dx.doi.org/10.1561/1500000069

part of such a research experiment. For instance Carmel *et al.* (2015) describe accessing a sample of real user queries issued on Yahoo Web mail service from a set of opted-in users.

Researchers do find that in some instances users may be comfortable in sharing some of their mailbox data with researchers or annotators, at least for the purpose of limited study. For instance, Kokkalis *et al.* (2013) propose a *Valet* approach to email data, where users knowingly share data with crowd-source workers for a limited duration of time.

Kokkalis *et al.* (2013) find that a majority of the study participants (18 of 28) were initially uncomfortable with data sharing. However, "*over half of those with concerns (10 of 18) ... reported that they felt more comfortable with the service over time, while no one reported a decrease in comfort*". This indicates that given a perceived value of annotations, users may be willing to provide limited access to their mailbox as long as:

(a) the users can clearly specify the time limits for workers access;

(b) the users have full control over what data is being shared; and

(c) the users perceive the services enabled by data sharing valuable.

This finding validates the feasibility of conducting personal search studies with small groups of volunteers who provide informed consent for data access for research purposes for a limited time period, subject to the review of the proper institutional review boards. Developing formal guidelines for such review boards, as well as ethics training programs for researchers are important investments to enable further research advances in the field (Gibney, 2017).

7.2 Data Bias

Despite the privacy limitations, it is important to be able to leverage some user information in order to improve the efficacy of email search and discovery algorithms. One such source of information is user interaction data, especially clicks. Therefore, in this section, we turn our attention to a class of techniques that aims to optimize email search ranking purely

The version of record is available at: http://dx.doi.org/10.1561/1500000069

from click data, without access to any human relevance judgments. As prior work demonstrates, such techniques can be applied to any document corpus with search logs (Joachims, 2002); in this section we focus on their specific applications to email search, while providing a brief survey of relevant prior work on utilizing click data.

Click data from search logs has been extensively explored for search ranking optimization, starting with the seminal early work by Joachims (2002). While click data is attractive due to its abundance, its main limitation is bias – user click behavior may be biased by factors not directly related to document relevance – e.g., position bias (Radlinski *et al.*, 2008), presentation bias (Yue *et al.*, 2010), or freshness bias (Zhang *et al.*, 2011).

Multiple click models that attempt to simulate user behavior in the presence of such biases have been developed to mitigate their effect (see Chuklin *et al.* (2015) for an overview). However, it is important to note that click models, which learn click probabilities from large quantities of clicks for individual *<query, document>* pairs, cannot be directly utilized in personal search applications like email search, where documents are not shared across users (Wang *et al.*, 2016a). It is thus often impractical to accumulate a large number of clicks for each *<query, document>* pair.

Instead, a class of techniques called *unbiased learning-to-rank* has been recently proposed, based on the counterfactual learning framework (Joachims *et al.*, 2017). An attractive property of unbiased learning-to-rank is that it does not require queries to repeat in the click logs, which makes it a fitting choice for email search applications. In the remainder of this section, we first provide a basic primer on the topic of position bias correction via unbiased learning-to-rank in Section 7.2.1. We then discuss specific ways to combat position bias in email search in Section 7.2.2.

7.2.1 Position Bias Correction

Formal Definition

Position bias can be modeled via a simple yet effective generative click model (Chuklin *et al.*, 2015). This model assumes that the *observed* click

The version of record is available at: http://dx.doi.org/10.1561/1500000069

Bernoulli variable C depends on two *other* hidden Bernoulli variables E and R. E represents the event whether a user examines a document at a certain position k. R represents the event whether a document d is relevant to a query q. Specifically,

$$P(C = 1|q, d, k) = P(E = 1|k) \cdot P(R = 1|q, d), \tag{7.3}$$

where $P(C = 1|q, d, k)$ is the probability of clicking document d that is shown at position k given query q, $P(E = 1|k)$ is the probability that position k is examined, and $P(R = 1|q, d)$ is the probability that document d is relevant to query q. For succinctness, we use the following shorthand in the remainder of this discussion:

$$\theta_k = P(E = 1|k) \tag{7.4}$$

$$\gamma_{q,d} = P(R = 1|q, d) \tag{7.5}$$

$$\xi_{q,d,k} = \theta_k \gamma_{q,d}. \tag{7.6}$$

Intuitively, the click model $\xi_{q,d,k}$ decouples the concepts of examination and relevance. It assumes that the examination only depends on the position, while relevance only depends on the query and document. Therefore, this model accounts for click position bias, which is often observed in real world search engines. While document relevance remains constant, regardless of where it is displayed, its probability of examination θ_k varies by its position. In traditional search interfaces, where the results are presented in a vertical order, and are examined in a top down fashion, this leads to rarer examination of documents at the bottom ranks (Joachims *et al.*, 2007).

With this click model at hand, we next focus on various way to estimate the hidden examination variable θ_k. Note that given the definition of the observed click probability $\xi_{q,d,k}$ in Equation 7.6, we need a way to either estimate the hidden relevance variable $\gamma_{q,d}$, or eliminate it from the equation, which can be done either with or without randomizing the search results.

Position Bias Estimation with Randomization

A simple way to estimate $\xi_{q,d,k}$ is randomizing the order of the search results shown to the users. It is easy to see that given sufficient search

The version of record is available at: http://dx.doi.org/10.1561/1500000069

traffic, result randomization effectively integrates out the relevance component $\gamma_{q,d}$, as we will observe a large sample of all possible orderings. There are several ways to perform result randomization, and we describe two commonly used methods in this section.

Wang *et al.* (2018) prove that given a randomized data set $\mathcal{R}$ where the top N documents are randomly shuffled before showing them to users, the probability of examination θ_k is proportional to the number of clicks in $\mathcal{R}_k$. While simple to implement and effective for precise estimation of the position bias *Randomize TopN* can significantly lower the relevance of the top results, and thus create an undesired degradation in user experience.

A relatively milder intervention is to randomize pairs. Joachims *et al.*, 2017 show that an experiment that swaps results at rank 1 and rank k at random, gives a good estimation of $\frac{\theta_1}{\theta_k}$ for $k = 1 \dots N$. Joachims *et al.*, 2017 further demonstrate that it is sufficient to know the ratio between the different θ_k's for unbiased learning algorithms, not their absolute values.

This intervention can still be relatively disruptive, as the most relevant result could move to the k-th position at the bottom of the list. Instead, Wang *et al.* (2018) propose *Randomize Pair*, which swaps only adjacent pairs at position $k-1$ and k. k's are varied, and search logs for each k are collected separately. Such an intervention gives us a good estimation of $\frac{\theta_k}{\theta_{k-1}}$. A chain rule can be then applied to estimate the relative ratio between θ_1 and θ_k:

$$\frac{\theta_k}{\theta_1} = \frac{\theta_k}{\theta_{k-1}} \cdot \frac{\theta_{k-1}}{\theta_{k-2}} \cdots \frac{\theta_2}{\theta_1}.$$

Position Bias Estimation without Randomization

To be effective, result randomization has to be done on real search traffic, and that can be harmful to the user experience due to the shuffling of the search results. For instance, Wang *et al.* (2018) show that *Randomize TopN* and *Randomize Pair* can lead to 14% and 7% respective drops in the mean reciprocal click position metric for email search. Therefore, estimating position bias without result randomization is a challenging, but important research topic. Next, we briefly review two representative techniques that address this problem.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

Wang *et al.* (2018) propose a regression-based EM algorithm that does not require randomization to learn the position bias. Given a regular click log $\mathcal{L} = \{(c_i, q_i, d_i, k_i)\}_{i=1}^{N}$, the log likelihood of generating this data is

$$\log P(\mathcal{L}) = \sum_{(c,q,d,k)\in\mathcal{L}} c \log \theta_k \gamma_{q,d} + (1-c)\log(1-\theta_k\gamma_{q,d}).$$

The EM (Expectation-Maximization) algorithm can now be used to find the parameters that maximize the log likelihood of the whole data.

The regression-based EM algorithm only modifies the Maximization step in the standard EM algorithm, Using the exact identifiers is particularly challenging in the case of email search, due to click sparsity caused by each user having a separate corpus of documents. Instead of directly working with (q, d) identifiers, Wang *et al.* (2018) assume there is a feature vector $\mathbf{x}_{q,d}$ representing them, and use a function to compute the relevance $\gamma_{q,d} = f(\mathbf{x}_{q,d})$. The Maximization step is then to find a regression function $f(\mathbf{x})$ to maximize the likelihood of the data given the estimation from the Expectation step.

Intervention Harvesting (Agarwal *et al.*, 2019b) is another technique to learn position bias without randomization. It is motivated by the *Randomize Pair* approach and generalizes the *Randomize Pair* approach for any pair of positions k and k'. It views a *Randomize Pair* intervention as imposing two ranking functions: one is the original ranking, and the other is the one that only differs from the previous one by swapping documents at k and k'.

Instead of running intervention experiments, the key idea behind Intervention Harvesting is that natural interventions are readily available in virtually any operational system – namely that there is more than one ranking function employed in a real-world search engine. In the *Randomize Pair* approach, both ranking functions require the same amount of traffic. This condition is not required in the Intervention Harvesting approach, which instead computes the normalized clicks $c_k^{k,k'}$ and $c_{k'}^{k,k'}$. The click count normalization accounts for non-uniform assignment probabilities to positions k and k' by the two ranking functions. Based on these normalized counts of all pairs of positions k and k', Agarwal *et al.* (2019b) develop an *AllPairs* estimator for position bias, and empirically demonstrate its robustness.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

Unbiased Metrics

The position bias estimation techniques described above can be used to *unbias* metrics defined over biased click data. Recall that we introduced a weighted variant of the Mean Reciprocal Rank (*wMRR*) metric in Equation 2.4 in Section 2.5. Wang *et al.* (2016a) show that weighting the i'th query based on the inverse propensity of the position of its click, $w_i = \frac{1}{\theta_{k_i}}$, provides a principled way to remove position bias from the *MRR* metric. Intuitively, this unbiased *wMRR* metric will assign a higher weight to queries with clicks at *lower* position, since they are likely to be underrepresented in the data due to smaller probability of examination.

In a multi-click setting, an unbiased version of the Discounted Cumulative Gain metric (*DCG*), can be similarly defined, by weighting the gain from each clicked document by its inverse propensity (Joachims *et al.*, 2017). Such an unbiased metric can be readily optimized by learning-to-rank algorithms such as LambdaRank (Burges, 2010).

7.2.2 Position Bias Correction in Email Search

In the previous section, we have discussed the theoretical grounding of some of the commonly used position bias estimation techniques in the unbiased learning-to-rank framework. In this section, we turn our attention to the empirical evaluation of unbiased learning-to-rank in the context of email search.

Wang *et al.* (2016a) examine the question of how pronounced position bias in email search is, and whether user clicks are indeed affected by their position in the ranked list. They run a *Randomize TopN* experiment for a small fraction of users of an email search service (Gmail), as well as a cloud file search service (Google Drive). Figure 7.2 demonstrates the propensity scores θ_k obtained by this experiment. As we can see in Figure 7.2, email search users are more than twice as likely to click on the first position compared to the fourth position, regardless of the result relevance (as email ranks in this experiment are randomized). The position bias still exists, but is much less pronounced in the cloud file search setting. This demonstrates the importance of separately estimating propensity scores for each individual search service, as they

may be affected by multiple factors. Moreover, they should be re-estimated periodically, as they are informed by users' past experience with the search service.

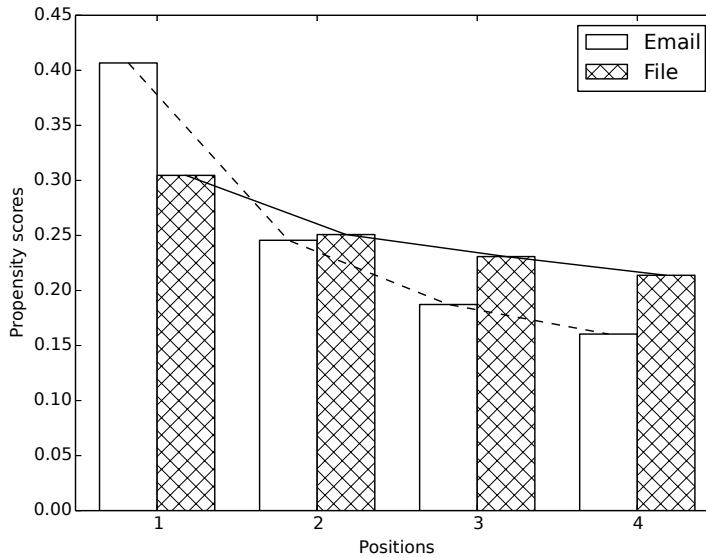


Figure 7.2: Propensity scores obtained by a *Randomize TopN* experiment conducted in email and cloud file search services (Wang *et al.*, 2016a).

However running result randomization to estimate the probability of examination at each position is likely to have a severe negative impact on search quality, as shown in Table 7.2. This can be somewhat ameliorated by applying the *Randomize Pair* technique, however some significant losses in performances can still be observed, especially for swaps at the top positions. This further motivates the use of non-randomization propensity techniques described in Section 7.2.1.

Finally, several publications confirm the importance of accurate position bias estimates for the purpose of tangible search quality improvements. For instance, Wang *et al.* (2016a) apply *Randomize TopN* inverse propensity weighting when training an unbiased learning-to-rank model using email search click data. They find that such weighting can improve the click-through rate by up to 0.7% using an experiment on live traffic.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

Table 7.2: The effects of *Randomize TopN (RandTopN)* and *Randomize Pair (RandPair)*, as measured by the relative change of MRR against the production system (Wang *et al.*, 2018). * indicates statistically significant differences.

	Email (N=3)	File Storage (N=5)
RandTopN	-13.94%*	-31.04%*
$RandPair(1,2)$	-6.80%*	-12.44%*
$RandPair(2,3)$	-0.56%	+3.75%
$RandPair(3,4)$	+0.20%	+1.09%
$RandPair(4,5)$	+0.38%	+0.36%

Wang *et al.* (2018) demonstrate that similar improvements can be obtained when the regression-based EM algorithm is used for propensity estimation. Such improvements, while small in terms of percentages, are highly significant for systems that serve millions of users daily, as they can significantly increase consumer satisfaction and the productivity of enterprise workers.

The click bias model presented in Equation 7.3 assumes that clicks indicate that a document was both examined by the user and relevant. However, in realistic settings, non-relevant documents may also be clicked as a result of user judgment error. Agarwal *et al.* (2019a) postulate that this can be modeled by position-dependent *trust bias*. Such trust bias results from the fact that the perceived relevance of the document may be influenced not only by its actual relevance, but also by its position in the ranked list. Agarwal *et al.* (2019a) use ϵ_k^+ and ϵ_k^- to respectively denote perceived relevance, or lack thereof. Thus, the position bias model (Equation 7.3) can be generalized to model trust bias as well:

$$P(C = 1|q, d, k) = \theta_k(\epsilon_k^+ \gamma_{k,d} + \epsilon_k^-(1 - \gamma_{k,d})). \tag{7.7}$$

Agarwal *et al.* (2019a) extend the regression-based EM algorithm by Wang *et al.* (2018) to estimate trust bias parameters ϵ_k^+ and ϵ_k^-. They demonstrate that the obtained inverse propensity weights can further improve click-through rates in a live experiment in email search, when compared to the version that does not account for trust bias.

Some follow-up work extends the position bias model to account for query context as well. For instance, position bias may differ between

The version of record is available at: http://dx.doi.org/10.1561/1500000069

queries where the user is most likely to click the first result (e.g., the latest hotel booking), and queries where the user may be more likely to examine the ranked list in depth (e.g., an old email from a friend). Wang *et al.* (2016a) propose a *generalized bias model* for email search that explicitly attempts to predict position bias for a query given its features such as query length or the categories of the clicked document. Fang *et al.* (2019) extend the Intervention Harvesting approach discussed in Section 7.2.1 to take into account query context such as its length, its position in the search session, and the number of returned results.

Overall, unbiased learning-to-rank has been proven to be an influential technique in email search and beyond. It remains an active research area, as evidenced by recent tutorials[5], open-source code releases[6], and surveys (Ai *et al.*, 2021). One interesting direction for future work is applying the most recent work in the area — including a dual learning algorithm for propensity and relevance estimation (Ai *et al.*, 2018) and a unification of unbiased learning-to-rank with online learning (Oosterhuis and Rijke, 2021) — in the context of email search.

7.3 Data Aggregation

Thus far in this chapter, we tackled the issues of data privacy and data bias in personal search applications such as email search. Both of these issues stem from *data sparsity*: in email search, each user has access only to their own personal corpus (e.g., emails, documents or multimedia files). Data sparsity has an additional important limitation. Cross-user interactions with the same item, which are common in web search (i.e., millions of users visiting the same web page) are non-existent in personal search.

As an example, consider the email search example in Figure 7.3. In this case the user skipped the first two results (even though they might have more terms in common with the query *book order number*) and clicked on the last result. It would be impossible to directly leverage this specific interaction to learn a model for other users given the private

[5] https://ultr.aiqingyao.org/
[6] https://github.com/ULTR-Community/ULTRA

nature of the interaction (since no other user received an email with the exact same order number).

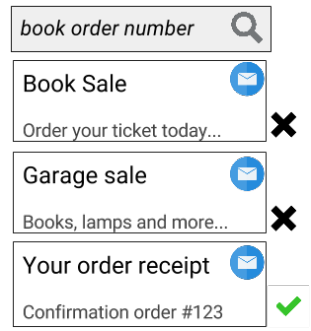


Figure 7.3: Illustrative example of email search results for query *[book order number]* as shown by Bendersky *et al.* (2017). The first two results are skipped, and the last one is clicked.

However, by *aggregating* non-private query and document attributes (i.e., those that exclude any personal information such as order number) across a large number of user interactions, it is possible to identify privacy-preserving query-document *associations* that can be leveraged to improve search quality across all users. For instance, by using term associations, we can learn that emails with the frequent term *receipt* in the subject are likely to be relevant to queries containing the frequent n-gram *order number*. As another example, using the structural templates described in Section 4.3.1, we can learn that emails from an online bookstore *AliceBookseller.com* that correspond to a subject template *Your order receipt ** are more likely to be relevant to queries containing the frequent n-gram *book order*.

Bendersky *et al.* (2017) propose to address the issue of click sparsity through aggregating cross-user interactions via attribute modeling of email messages. This approach is schematically described in Figure 7.4. Both documents and queries are projected into an aggregated attribute space, and the matching is done through that intermediate representation, rather than directly. Since we assume that the attributes are semantically meaningful, we expect that similar personal documents and queries will share many of the same aggregate attributes, making the attribute level matches a useful feature in a learning-to-rank model. Some

examples of privacy-preserving query-document associations that could be learned by aggregating across a large number of private user interactions include email labels or categories, machine-generated structural email templates, and frequent n-grams appearing in the content of user queries and clicked email subjects. Bendersky *et al.* (2017) demonstrate that, when incorporated into an unbiased learning-to-rank framework, these aggregated attributes contribute to significant email search quality improvements.

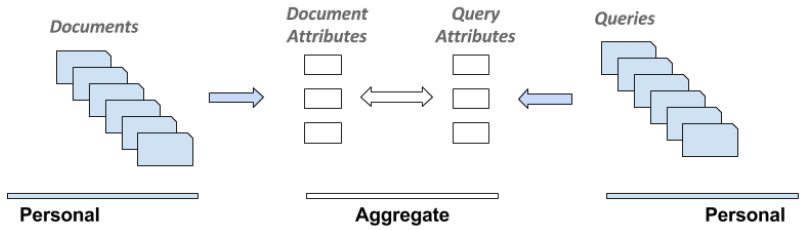


Figure 7.4: Document and query attribute aggregation and matching as shown by Bendersky *et al.* (2017). Instead of directly matching attributes at the personal (email) level, they can be matched via aggregating cross-user interactions, to overcome data sparsity.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

8

Open Research Challenges

Thus far in the survey we have discussed the extensive research on the various aspects of email search and discovery, including search engine design, email and query understanding, intelligent assistance, and privacy-preserving user data management. However, despite the breadth of this research, many open challenges still remain. Recent fundamental research breakthroughs at the intersection of the fields of deep learning, differential privacy (Abadi *et al.*, 2016), unbiased learning (Joachims *et al.*, 2017), multi-modal content understanding (Zhuang and Liu, 2019), multi-task learning (Caruana, 1997), domain adaptation (Ganin *et al.*, 2016), and federated learning (Konečný *et al.*, 2016) open up fascinating directions for future advancements in the field of email search and discovery.

In this chapter, we discuss some of these potential directions in detail. Some of them are not specific to email, and have also been identified as key challenges during the latest Strategic Workshop on Information Retrieval (Culpepper *et al.*, 2018): multi-modality (Section 8.1), conversational information seeking (Section 8.3), and fairness (Section 8.5). Some are specific to email search, including domain-specific models (Section 8.2) and user privacy (Section 8.6). Some

The version of record is available at: http://dx.doi.org/10.1561/1500000069

combine general search problems with the distinctive characteristics of access to personal content (Section 8.4).

Finally, it is important to note that much of the visionary research cited in this chapter comes from academic institutions rather than industrial research labs. This is despite the fact that there are only few publicly available email datasets. Some of the work circumvents the steep entry barrier into this research field by extensively utilizing crowd-sourcing for data annotation (Liang *et al.*, 2019), surveys of email search behavior (Swaminathan *et al.*, 2017), and even accessing redacted email content with user consent (Swaminathan *et al.*, 2017; Kokkalis *et al.*, 2013). Other researchers propose retrieval systems for experiments over private collections with content obfuscation (Shao *et al.*, 2019), and logging with differential privacy (Feild *et al.*, 2011). Such techniques can be instrumented within a research lab or a company to facilitate privacy-preserving data collection or in-situ studies.

We hope that the research presented in this chapter will inspire our readers to deeply reflect on their assumptions about search and discovery in personal communication archives, and eventually advance the field beyond the existing paradigm of email search, as reflected in current production systems. While email and other personal content management systems have a storied past and a well-established present, there is still much that remains to be done to improve search and discovery in these systems.

8.1 Multi-modal Search

Thus far in this survey we have mostly focused on textual information retrieval tasks, as email is inherently a written communication medium. However, in addition to email text, there is a lot of valuable content contained in email attachments: PDFs, images, links to sites containing multimedia (e.g., YouTube) and so on. How can unlocking the content of these attachments improve the effectiveness of an email search system?

Some recent advances in search over personal media suggest an answer to this question. For instance, Jiang *et al.* (2017) conduct a study of the Flickr personal media repository. They propose a Visual

The version of record is available at: http://dx.doi.org/10.1561/1500000069

Query Embedding method that maps query terms to related visual concepts, using click data as an indicator of relevance. They learn the mapping using deep architectures. In particular, they find that a multi-layer perceptron with a max-pooling layer is highly effective, significantly outperforming a variety of baselines, including exact match, WordNet-based matching and pre-trained word2vec embeddings. This indicates the importance of leveraging user interactions (clicks, in this case) for better multi-modal representations.

Jiang *et al.* (2017) note that a majority of personal media queries are relatively short, averaging at 1.5 words (stopwords excluded). They also focus on the visual rather than topical content (e.g., "*lake*" rather than "*trip to Tahoe*"). How to map topical queries into their visual counterparts remains an interesting research question. For instance, one can imagine that a query "*trip to Tahoe*" over one's mailbox may potentially retrieve emails with attachments of photos of the lake or the ski slopes taken during the trip.

As a step towards this goal of answering complex information needs over private multi-modal corpora, Liang *et al.* (2019) propose MemexQA: a question answering system for personal media collections. Given a sequence of user photos, the MemexQA system can answer factoid questions like *"What did we eat for Aldo's birthday?"* accompanied by a group of photos that can help the user to verify the correctness of the answer (see Figure 8.1).

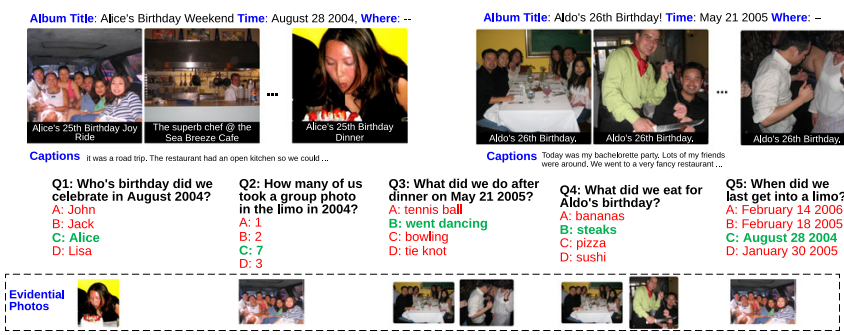


Figure 8.1: Questions, multiple-choice answers and supporting evidence photos in the MemexQA system (the correct answers are marked in green) (Liang *et al.*, 2019).

The version of record is available at: http://dx.doi.org/10.1561/1500000069

Anguera *et al.* (2008) consider the case for multi-modal search on mobile devices. They postulate that in mobile search, spoken rather than typed queries are more likely, and therefore propose a multi-modal indexing and retrieval system that allows the user to associate audio tags with an image at indexing time for easier retrievability. A fully automatic end-to-end personal content search system that bridges the semantic gap between the various input and output modalities (speech, image, video, text) is an exciting direction for future research.

Overall, the research described in this section demonstrates that there is a value that can be unlocked by modeling emails holistically. This would include any image or video attachments, and can help in answering topical questions and voice requests through related visual concepts.

8.2 Domain-specific Search and Domain Adaptation

Email search is an important discovery tool for the enterprise. The Radicati Group, Inc. (2015) report indicates that the number of enterprise emails currently surpasses the number of consumer (not machine-generated) emails; the same report indicates that an average enterprise user receives close to 130 emails on a daily basis. Worker mailboxes contain large amounts of enterprise knowledge, and workers may spend as much as 19% of their work time on search and information gathering activities (Chui *et al.*, 2012). Given these statistics, it is not an exaggeration to state that improving email search in the enterprise can directly improve worker productivity, and positively impact the world economy.

In the enterprise setting, in addition to the personal mailbox, it is interesting to consider the unique properties of each individual business or company. Therefore, applying a single monolithic search solution for all the businesses may lead to sub-optimal result quality for each individual enterprise. Alternatively, training an individual ranking model for each enterprise is untenable, due to the limited amounts of interaction data available for training purposes, especially for smaller businesses.

Tran *et al.* (2019) investigate the use of domain adaptation to address this problem. In domain adaptation we assume that a model trained on data from a large *source domain* can be adapted to a small *target domain*,

The version of record is available at: http://dx.doi.org/10.1561/1500000069

where training data is absent or limited. Most adaptation strategies attempt to bring the source and the target feature representations to be as semantically close as possible to each other, in order to enable effective knowledge transfer across domains (Ganin *et al.*, 2016). To this end, Tran *et al.* (2019) propose a Maximum Mean Discrepancy (MMD) loss regularization. Formally, for a given domain a loss is given as

$$\mathcal{L} = \mathcal{L}_p + \lambda_{MMD}\mathcal{L}_{MMD}, \tag{8.1}$$

where $\mathcal{L}_p$ is a ranking loss in the domain, and the MMD loss $\mathcal{L}_{MMD}$ is given by the L_2 norm of the difference between the feature distributions of the source and the target domains. Here, the source domain is a large sample of users from the overall search traffic, while the target domain is a specific enterprise. λ_{MMD} is a tunable parameter.

Tran *et al.* (2019) demonstrate that MMD regularization significantly improves the quality of search in a given domain, compared to several baselines, while being robust to changes in the setting of the λ_{MMD} parameter. Experiments over four small enterprises demonstrate $3\% - 8\%$ improvements when compared to simply training the model over the domain data alone. These results clearly indicate the importance of thinking about enterprise email search as a mixture of specific enterprise needs with generilizable user behaviors, which opens up several interesting research directions.

8.3 Question Answering Systems for Personal Content

Intelligent assistants like Alexa, Google Assistant and Siri can handle simple questions about user personal content today, such as "*What's on my calendar tomorrow?*".[1] However, further research is still required into more complex questions, including questions that cannot be answered by a single email or calendar event, and require multi-document synthesis.

Balog and Kenter (2019) present a compelling vision towards the end goal of complex question answering over personal data. They propose the concept of a *personal knowledge graph* (PKG) – a structure that captures the entities, their attributes and their relations for a particular user.

[1] https://gsuiteupdates.googleblog.com/2019/11/use-google-assistant-with-your-g-suite.html

The version of record is available at: http://dx.doi.org/10.1561/1500000069

Balog and Kenter (2019) postulate that PKGs can help in a variety of applications from entity disambiguation ("*my mom*") and query understanding ("*bob's recent emails*") to a personalized conversational agent that can help complete complex tasks ("*schedule an appointment with a dentist, which was recently recommended by Alice*").

8.4 Search on Mobile and Wearable Devices

Search in general, and personal content search in particular, are rapidly moving to mobile environments. Search characteristics, both in terms of query distributions and their topics, as well as user behavior in response to the queries, shift significantly as users transition from desktop to mobile devices (Kamvar *et al.*, 2009). These changes may be especially significant in the era of intelligent assistants, such as Siri, Cortana or Google Assistant, that enable users to interact with their devices using natural language voice queries.

Mobile devices afford access to information that is not available on a desktop. This information may be highly valuable for improving user search experience. To take location as an example, Foley *et al.* (2018) analyze an anonymized search log of a mobile email application and find that "*a large fraction of queries include some term that is also part of the name or title of their location*". E.g., users entering a Target store, may search for their recent "*Target coupons*". One can imagine that such user searches may be better served by proactive assistance – e.g., automatically surfacing a recently received email promotion whenever a user enters a store where the discount is available. More generally, as discussed in Section 2.4, Zamani *et al.* (2017) show that situational context like country, query language, weekday and time of the query can all improve the quality of email search ranking in a desktop setting. It is plausible that an even finer grained situational context (e.g., GPS coordinates) will play an even larger role in a mobile search setting.

To examine the prevalence of mobile email search, Swaminathan *et al.* (2017) conduct a user study that finds that over 30% of email search queries are issued in on-the-go settings. Example information needs include order confirmations, coupon codes, links, event information, and contact information of retailers or individuals. Swaminathan *et al.*

(2017) argue that many of these information needs can be fulfilled more effectively by a question answering system, rather than a ranked list of results. This highlights the importance of question answering systems, as discussed in Section 8.3, for mobile email search. Figure 8.2 provides some illustrative examples of how Swaminathan *et al.* (2017) envision such question answering system operating across a range of on-the-go settings and devices: intelligent assistants, smart home appliances, wearable devices, etc.

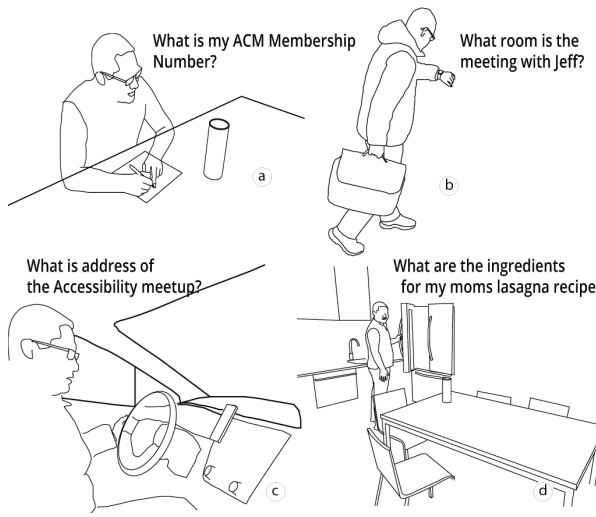


Figure 8.2: Illustrative examples of email search in various on-the-go settings (Swaminathan *et al.*, 2017).

Triaging, organizing and composing emails and documents on mobile devices is often uncomfortable and time-consuming. However, there is a trend of increasing mobile email usage. For instance, according to a recent survey by Watson Marketing (2018), half of all emails are read today on mobile devices (with some geographical variations). Therefore assistive features like the ones discussed in Section 6.4 have an ample opportunity to boost user productivity on mobile devices. For instance, Tata *et al.* (2017) report that accurate instant file suggestions can save up to 50% time in accessing a file on mobile, as compared to standard search or navigation. Therefore, assistive features that specifically target

The version of record is available at: http://dx.doi.org/10.1561/1500000069

mobile productivity is an important future research direction.

8.5 Beyond Relevance Ranking

While relevance ranking (a.k.a. "the ten blue links") is the bedrock of modern web search engines, most of these search engines have moved beyond purely relevance-based strategies, and consider many other factors in constructing the search results page. In what follows, we discuss some of these factors, and their applicability to email and other personal content search scenarios.

For instance, most of the existing work in personal search assumes the existence of a fixed user interface, where the search results are ordered top-down in descending relevance or chronological order. Such user interface — as repeatedly shown in prior work (Joachims, 2002; Wang *et al.*, 2016a) — is heavily prone to position bias, and the top positions receive the majority of the clicks. More advanced presentation layouts (Oosterhuis and de Rijke, 2018), better snippet design (Yue *et al.*, 2010) and online learning-to-rank that continuously adapts to user feedback (Kveton *et al.*, 2015) have all been proposed for correcting position biases in prior work, and are promising exploration avenues in the context of email search.

Another direction that has not seen much published work is merging multiple personal corpora (emails, files, calendar events) on a single results page, akin to the *federated search* setting that has been previously explored in the web (Arguello *et al.*, 2011) and enterprise (Arya *et al.*, 2015) search settings. Merging multiple result types may require considering the overall presentation, diversity and relevance of the presented search results in order to optimize the whole-page search utility, which has been explored in depth in web search (Santos *et al.*, 2015; Wang *et al.*, 2016b), and will undoubtedly lead to valuable user experience improvements in the personal search setting as well.

Fairness is another emerging research topic in ranking models (Singh and Joachims, 2018). As of this writing, there is no published work on fairness in the context of personal content search. Even the simple definition of ranking fairness in a setting like email search remains an open problem.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

However, fairness definitely plays an important role in the setting of personal communications. For instance, let us consider the case of email as a two-sided market in the form of senders and recipients. Contact recency and frequency have been shown to be important relevance features in email search (Carmel *et al.*, 2015), which may create and reinforce an unfair advantage to information surfaced from a close-knit group of contacts, and potentially exclude some less frequent contacts. A fair email search engine will need to provide some guarantees that all legitimate non-spam non-phishing communications have a chance to be shown in search results, while still ensuring high relevance and utility of search results. As email communications are at the center of many of our personal and work-related activities, ensuring fairness of information access in email search will have far-reaching implications.

8.6 Federated Learning

As discussed in Section 7.1, proper user data protection and privacy are of utmost importance and should be at the top of researchers' minds, when working with email and other personal content. A most visible expression of public concerns regarding the use of private data for research and development purposes is demonstrated by the recent regulations by the European Parliament and Council of the European Union (2016). These regulations stipulate large fines for violations of user data protection and privacy policies. However, the paradigm of a central server that has access to all the user data (e.g., search logs) — a paradigm we take for granted for most of the research discussed in this survey – is vulnerable to either unintentional or malicious breaches of this regulation, even if server side encryption and anonymization are applied.

One possible way to address these concerns is the *federated learning* paradigm that has gained traction in recent years. In this paradigm, the training data remains distributed over a large number of clients, and is never seen by the central server in its entirety. Instead, each client independently computes model updates based on its local data, and communicates these updates to a server, which aggregates the updates to compute a new global model. Konečný *et al.* (2016) provide

The version of record is available at: http://dx.doi.org/10.1561/1500000069

a good overview of federated learning, and propose approaches for improving client-server communication efficiency via structured and sketched updates. The former reduce communication costs by requiring them to have a certain pre-specified structure, while the latter applies lossy compression to the updates. Client updates may also be further anonymized via hashing and sub-samping (Apple, 2017).

Federated learning is an important future research direction for enhancing privacy in email search and discovery. Even if we set government regulations aside, consumers and enterprise users may still be hesitant to share the entirety of their private and personal data with a single third-party, regardless of the provision of certain privacy guarantees. In addition, local data storage will make server-side data attacks much less harmful to user privacy.

Wider adoption of federated learning may also open the doors for search and discovery experiences that work across multiple personal content management silos, especially if they are operated by different central third-parties. In addition, major tech companies are already providing federated learning in some of their products (McMahan and Ramage, 2017; Apple, 2017).

Najork (2018) lays out the important research questions that need to be addressed to adapt the ideas from federated learning to personal search, ranking and retrieval. Some of these questions are already starting to be answered by researchers. For instance, Shao *et al.* (2019) discuss a privacy-aware neural ranking method that replaces exact term matches with soft matching using obfuscated kernel values and term closures. Such a technique will allow the server to answer user queries without leaking exact word frequencies and occurrences. Zhang *et al.* (2016) discuss differential privacy applications for query log anonymization.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

9

Conclusions

In this survey, we have attempted to describe the unique challenges of building systems for search and discovery in personal email collections, which are inherently built for private content. Since our readers may be more familiar with information retrieval systems for public corpora, such as the web, our hope is that this survey will provide a fresh perspective on some unique aspects of private content search. In Chapters 2 – 5 we provide an in-depth discussion of the major parts of the email search engine, including its overall architecture (Chapter 2), search interface (Chapter 3), mail corpus organization (Chapter 4), and query understanding (Chapter 5).

In the latter parts of this survey, we go beyond search and discuss *assistive features*. We cover both existing assistance features such as content recommendation, activity prediction, and assisted composition (Chapter 6), as well as emerging ones such as personal assistants, and email access on mobile and wearable devices (Chapter 8). While these chapters do not directly deal with ad hoc relevance search, they provide some valuable lessons on how to improve the productivity of both consumers and enterprise email users. We hope that these chapters will showcase that – at least in some settings – the effort of explicit query

The version of record is available at: http://dx.doi.org/10.1561/1500000069

formulation can be avoided by proactive and timely assistance. We also highlight the fact that email is often tightly interconnected with other types of personal content such as task lists, calendars, and personal documents, and demonstrate how these connections make email search and discovery more effective.

In Chapter 7 we discuss *data privacy* and *data quality* considerations, which are of utmost importance in managing user data in email search and discovery systems, and in designing models that learn from user data. We discuss multiple privacy-preserving approaches including de-identification, k-anonymity, differential privacy, and transparent user controls to both mailboxes and search queries. As reusable test collections annotated by objective assessors are difficult to obtain and maintain, and private test collections are prohibitively costly to develop, user interactions (such as clicks) are often used for training email search ranking models. Therefore, we also discuss ways to correct bias and aggregate sparse user interactions to improve the quality of user interaction data used in email search and discovery research.

We hope that this survey helps to ignite interest in personal email search and discovery in the research community. While the email communication format has been with us for more than five decades now, there is still much to be done to simplify and improve email access and management in our everyday lives!

The version of record is available at: http://dx.doi.org/10.1561/1500000069

Acknowledgements

Many people made the existence of this survey possible. In particular, we would like to thank Maarten de Rijke for his patience throughout the manuscript preparation process, and for his early guidance that significantly influenced the direction of this work. We thank Corinna Cortes and Andrew Tomkins for their insightful comments on an early version of the manuscript. We thank the anonymous FnTIR reviewers for their detailed and constructive feedback, which helped us to further improve the survey and make our claims more precise, and the coverage more complete. Last, but not least, we would like to thank our families for their constant encouragement and support.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

References

Abadi, M., A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. (2016). "Deep Learning with Differential Privacy". In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM. 308–318.

AbdelRahman, S., B. Hassan, and R. Bahgat. (2010). "A New Email Retrieval Ranking Approach". *International Journal of Computer Science and Information Technology*. 2(5): 44–63. DOI: 10.5121/ijcsit.2010.2504.

Aberdeen, D., O. Pacovsky, and A. Slater. (2010). "The Learning Behind Gmail Priority Inbox". In: *LCCC: NIPS 2010 Workshop on Learning on Cores, Clusters and Clouds*.

Abu-Nimeh, S., D. Nappa, X. Wang, and S. Nair. (2007). "A Comparison of Machine Learning Techniques for Phishing Detection". In: *Proceedings of the Anti-Phishing Working Groups 2nd Annual eCrime Researchers Summit*. ACM. Pittsburgh, PA, USA. 60–69. DOI: 10.1145/1299015.1299021.

Agarwal, A., X. Wang, C. Li, M. Bendersky, and M. Najork. (2019a). "Addressing Trust Bias for Unbiased Learning-to-Rank". In: *Proceedings of the World Wide Web Conference. WWW '19*. San Francisco, CA, USA: Association for Computing Machinery. 4–14. DOI: 10.1145/3308558.3313697.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

Agarwal, A., I. Zaitsev, X. Wang, C. Li, M. Najork, and T. Joachims. (2019b). "Estimating Position Bias without Intrusive Interventions". In: *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining. WSDM '19*. 474–482.

Aggarwal, C. C. (2005). "On k-anonymity and the Curse of Dimensionality". In: *Proceedings of the 31st International Conference on Very Large Data Bases (VLDB 2005)*. VLDB Endowment. 901–909.

Agichtein, E., E. Brill, and S. Dumais. (2006). "Improving Web Search Ranking by Incorporating User Behavior Information". In: *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR '06*. Seattle, Washington, USA: ACM. 19–26. DOI: 10.1145/1148170.1148177.

Ai, Q., K. Bi, C. Luo, J. Guo, and W. B. Croft. (2018). "Unbiased Learning to Rank with Unbiased Propensity Estimation". In: *The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR '18*. Ann Arbor, MI, USA: Association for Computing Machinery. 385–394. DOI: 10.1145/3209978.3209986.

Ai, Q., S. T. Dumais, N. Craswell, and D. J. Liebling. (2017). "Characterizing Email Search using Large-scale Behavioral Logs and Surveys". In: *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*. 1511–1520. DOI: 10.1145/3038912.3052615.

Ai, Q., T. Yang, H. Wang, and J. Mao. (2021). "Unbiased Learning to Rank: Online or Offline?" *ACM Transactions on Information Systems (TOIS)*. 39(2): 1–29.

Ailon, N., Z. S. Karnin, E. Liberty, and Y. Maarek. (2013). "Threading Machine Generated Email". In: *Sixth ACM International Conference on Web Search and Data Mining, WSDM 2013, Rome, Italy, February 4-8, 2013*. 405–414. DOI: 10.1145/2433396.2433447.

Alrashed, T., A. H. Awadallah, and S. Dumais. (2018). "The Lifetime of Email Messages: A Large-Scale Analysis of Email Revisitation". In: *Proceedings of the 2018 Conference on Human Information Interaction & Retrieval. CHIIR '18*. New Brunswick, NJ, USA: ACM. 120–129. DOI: 10.1145/3176349.3176398.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

Anguera, X., J. Xu, and N. Oliver. (2008). "Multimodal Photo Annotation and Retrieval on a Mobile Phone". In: *Proceedings of the 1st ACM International Conference on Multimedia Information Retrieval*. ACM. 188–194.

Apple. (2017). "Learning with Privacy at Scale". URL: https://machinelearning.apple.com/docs/learning-with-privacy-at-scale/appledifferentialprivacysystem.pdf.

Arguello, J., F. Diaz, and J. Callan. (2011). "Learning to Aggregate Vertical Results into Web Search Results". In: *Proceedings of the 20th ACM International Conference on Information and Knowledge Management. CIKM '11*. Glasgow, Scotland, UK: ACM. 201–210. DOI: 10.1145/2063576.2063611.

Arya, D., V. Ha-Thuc, and S. Sinha. (2015). "Personalized Federated Search at LinkedIn". In: *Proceedings of the 24th ACM International Conference on Information and Knowledge Management*. ACM. 1699–1702.

Asadi, N., D. Metzler, T. Elsayed, and J. Lin. (2011). "Pseudo Test Collections for Learning Web Search Ranking Functions". In: *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR '11*. Beijing, China: ACM. 1073–1082. DOI: 10.1145/2009916.2010058.

Ashkan, A. and D. Metzler. (2019). "Revisiting Online Personal Search Metrics with the User in Mind". In: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2019)*. Paris, France: Association for Computing Machinery. 625–634. DOI: 10.1145/3331184.3331266.

Aumüller, M., E. Bernhardsson, and A. Faithfull. (2017). "ANN-benchmarks: A benchmarking tool for approximate nearest neighbor algorithms". In: *International Conference on Similarity Search and Applications*. Springer. 34–49.

Avigdor-Elgrabli, N., M. Cwalinski, D. Di Castro, I. Gamzu, I. Grabovitch-Zuyev, L. Lewin-Eytan, and Y. Maarek. (2016). "Structural Clustering of Machine-Generated Mail". In: *Proceedings of the 25th ACM International Conference on Information and Knowledge Management. CIKM '16*. Indianapolis, Indiana, USA: ACM. 217–226. DOI: 10.1145/2983323.2983350.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

Azarbonyad, H., R. Sim, and R. W. White. (2019). "Domain Adaptation for Commitment Detection in Email". In: *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. ACM. 672–680.

Balog, K. and T. Kenter. (2019). "Personal Knowledge Graphs: A Research Agenda". In: *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval*. ACM. 217–220.

Bar-Yossef, Z., I. Guy, R. Lempel, Y. S. Maarek, and V. Soroka. (2006). "Cluster Ranking with an Application to Mining Mailbox Networks". In: *Sixth International Conference on Data Mining (ICDM'06)*. 63–74. DOI: 10.1109/ICDM.2006.35.

Bawa, M., R. J. Bayardo Jr, R. Agrawal, and J. Vaidya. (2009). "Privacy-preserving Indexing of Documents on the Network". *The VLDB Journal—The International Journal on Very Large Data Bases*. 18(4): 837–856.

Bekkerman, R. (2004). "Automatic Categorization of Email into Folders: Benchmark Experiments on Enron and SRI Corpora". *Tech. rep.* No. 218. Computer Science Department Faculty Publication Series, University of Massachusetts Amherst.

Bell, R. M. and Y. Koren. (2007). "Lessons from the Netflix Prize Challenge." *SiGKDD Explorations*. 9(2): 75–79.

Bendersky, M., X. Wang, D. Metzler, and M. Najork. (2017). "Learning from User Interactions in Personal Search via Attribute Parameterization". In: *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM. 791–799.

Bendersky, M., X. Wang, M. Najork, and D. Metzler. (2018). "Learning with Sparse and Biased Feedback for Personal Search". In: *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI 2018)*. Stockholm, Sweden. 5219–5223. DOI: 10.24963/ijcai.2018/725.

Bengio, Y., R. Ducharme, P. Vincent, and C. Jauvin. (2003). "A Neural Probabilistic Language Model". *Journal of Machine Learning Research*. 3(Feb): 1137–1155.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

Bennett, P. N. and J. G. Carbonell. (2007). "Combining Probability-Based Rankers for Action-Item Detection". In: *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*. Rochester, New York: Association for Computational Linguistics. 324–331. URL: https://www.aclweb.org/anthology/N07-1041.

Berry, M. W., M. Browne, and B. Signer. (2001). "Topic annotated Enron email data set". *Philadelphia: Linguistic Data Consortium*.

Bhole, A. and R. Udupa. (2015). "On Correcting Misspelled Queries in Email Search." In: *Proceedings of the 29th AAAI Conference on Artificial Intelligence*.

Bhushan, A., K. Pogran, R. Tomlinson, and J. White. (1973). "Standardizing Network Mail Headers". URL: https://tools.ietf.org/html/rfc561.

Birrell, A. D., E. P. Wobber, and M. D. Schroeder. (1997). *Pachyderm*. URL: http://birrell.org/andrew/pachywww/.

Brin, S. and L. Page. (1998). "The Anatomy of a Large-scale Hypertextual Web Search Engine". *Computer Networks and ISDN Systems*. 30(1-7): 107–117.

Broder, A. Z., N. Eiron, M. Fontoura, M. Herscovici, R. Lempel, J. McPherson, R. Qi, and E. Shekita. (2006). "Indexing Shared Content in Information Retrieval Systems". In: *Advances in Database Technology - EDBT 2006*. Ed. by Y. Ioannidis, M. H. Scholl, J. W. Schmidt, F. Matthes, M. Hatzopoulos, K. Boehm, A. Kemper, T. Grust, and C. Boehm. Berlin, Heidelberg: Springer Berlin Heidelberg. 313–330.

Burges, C. J. (2010). "From RankNet to LambdaRank to LambdaMART: An overview". *Learning*. 11(23–581): 81.

Carenini, G., R. T. Ng, and X. Zhou. (2007). "Summarizing Email Conversations with Clue Words". In: *Proceedings of the 16th International Conference on World Wide Web (WWW 2007)*. Banff, Alberta, Canada: Association for Computing Machinery. 91–100. DOI: 10.1145/1242572.1242586.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

Carlini, N., C. Liu, Ú. Erlingsson, J. Kos, and D. Song. (2019). "The Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks". In: *Proceedings of the 28th USENIX Conference on Security Symposium (SEC 2019)*. 267–284.

Carmel, D., G. Halawi, L. Lewin-Eytan, Y. Maarek, and A. Raviv. (2015). "Rank by Time or by Relevance?: Revisiting Email Search". In: *Proceedings of the 24th ACM International Conference on Information and Knowledge Management*. ACM. 283–292.

Carmel, D., L. Lewin-Eytan, A. Libov, Y. Maarek, and A. Raviv. (2017a). "Promoting Relevant Results in Time-Ranked Mail Search". In: *Proceedings of the 26th International Conference on World Wide Web. WWW '17*. Perth, Australia: International World Wide Web Conferences Steering Committee. 1551–1559. DOI: 10.1145/3038912.3052659.

Carmel, D., L. Lewin-Eytan, A. Libov, Y. Maarek, and A. Raviv. (2017b). "The Demographics of Mail Search and Their Application to Query Suggestion". In: *Proceedings of the 26th International Conference on World Wide Web (WWW 2017)*. International World Wide Web Conferences Steering Committee. Perth, Australia: Association for Computing Machinery. 1541–1549. DOI: 10.1145/3038912.3052658.

Caruana, R. (1997). "Multitask Learning". *Machine Learning*. 28(1): 41–75.

Carvalho, V. R. and W. W. Cohen. (2004). "Learning to Extract Signature and Reply Lines from Email". In: *Proceedings of the Conference on Email and Anti-Spam*. Vol. 2004.

Carvalho, V. R. and W. W. Cohen. (2007). "Preventing Information Leaks in Email". In: *Proceedings of the 2007 SIAM International Conference on Data Mining*. SIAM. 68–77.

Carvalho, V. R. and W. W. Cohen. (2008). "Ranking Users for Intelligent Message Addressing". In: *European Conference on Information Retrieval*. Springer. 321–333.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

Chen, F., K. Xia, K. Dhabalia, and J. I. Hong. (2019a). "MessageOnTap: A Suggestive Interface to Facilitate Messaging-related Tasks". In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI 2019). CHI '19*. Glasgow, Scotland UK: Association for Computing Machinery. 575:1–575:14. DOI: 10.1145/3290605.3300805.

Chen, M. X., B. N. Lee, G. Bansal, Y. Cao, S. Zhang, J. Lu, J. Tsay, Y. Wang, A. M. Dai, Z. Chen, T. Sohn, and Y. Wu. (2019b). "Gmail Smart Compose: Real-Time Assisted Writing". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. KDD '19*. Anchorage, AK, USA: Association for Computing Machinery. 2287–2295. DOI: 10.1145/3292500.3330723.

Chui, M., J. Manyika, J. Bughin, R. Dobbs, C. Roxburgh, H. Sarrazin, G. Sands, and M. Westergren. (2012). "The Social Economy: Unlocking Value and Productivity Through Social Technologies". *Tech. rep.* McKinsey Global Institute. URL: https://www.mckinsey.com/industries/technology-media-and-telecommunications/our-insights/the-social-economy.

Chuklin, A., I. Markov, and M. de Rijke. (2015). *Click Models for Web Search*. Morgan & Claypool. DOI: 10.2200/S00654ED1V01Y201507ICR043.

Cincotta, A. V. (1983). "Navy Electronic Mail Service User's Guide." *Tech. rep.* No. CMLD-83-22. Bethesa, MD 20084: David W. Taylor Naval Ship Research and Development Center.

Claude, F., A. Fariña, M. A. Martínez-Prieto, and G. Navarro. (2011). "Indexes for Highly Repetitive Document Collections". In: *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*. ACM. 463–468.

Cohen, W. W. *et al.* (1996). "Learning Rules That Classify E-mail". In: *AAAI Spring Symposium on Machine Learning in Information Access*. Vol. 18. Stanford, CA. 25.

Cohen, W. W., V. R. Carvalho, and T. M. Mitchell. (2004). "Learning to Classify Email into "Speech Acts"". In: *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*. 309–316.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

Cohen, W. W. (2015). "Enron Email Dataset". URL: http://www.cs.cmu.edu/~enron/.

Cormack, G. V. (2007). "TREC 2007 Spam Track Overview". In: *Proceedings of the Sixteenth Text REtrieval Conference (TREC 2007)*.

Crammer, K., A. Kulesza, and M. Dredze. (2009). "Adaptive Regularization of Weight Vectors". In: *Advances in neural information processing systems*. 414–422.

Craswell, N., A. P. de Vries, and I. Soboroff. (2005). "Overview of the TREC 2005 Enterprise Track". In: *Proceedings of the Fourteenth Text REtrieval Conference (TREC 2005)*. Vol. 5. 199–205.

Cui, Q., G.-V. Jourdan, G. V. Bochmann, R. Couturier, and I.-V. Onut. (2017). "Tracking Phishing Attacks Over Time". In: *Proceedings of the 26th International Conference on World Wide Web. WWW '17*. Perth, Australia: International World Wide Web Conferences Steering Committee. 667–676. DOI: 10.1145/3038912.3052654.

Culpepper, J. S., F. Diaz, and M. D. Smucker. (2018). "Report from the Third Strategic Workshop on Information Retrieval (SWIRL)".

Cutrell, E., S. T. Dumais, and J. Teevan. (2006). "Searching to Eliminate Personal Information Management". *Commun. ACM*. 49(1): 58–64.

Dabbish, L. A., R. E. Kraut, S. Fussell, and S. Kiesler. (2005). "Understanding Email Use: Predicting Action on a Message". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. 691–700.

Dai, N., M. Shokouhi, and B. D. Davison. (2011). "Learning to Rank for Freshness and Relevance". In: *Proceedings of the 34th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. URL: https://www.microsoft.com/en-us/research/publication/learning-to-rank-for-freshness-and-relevance/.

Di Castro, D., I. Gamzu, I. Grabovitch-Zuyev, L. Lewin-Eytan, A. Pundir, N. R. Sahoo, and M. Viderman. (2018). "Automated Extractions for Machine Generated Mail". In: *Companion Proceedings of the The Web Conference 2018. WWW '18*. Lyon, France: International World Wide Web Conferences Steering Committee. 655–662. DOI: 10.1145/3184558.3186582.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

Di Castro, D., Z. Karnin, L. Lewin-Eytan, and Y. Maarek. (2016a). "You've got Mail, and Here is What you Could do With It!: Analyzing and Predicting Actions on Email Messages". In: *Proceedings of the 9th ACM International Conference on Web Search and Data Mining (WSDM 2016)*. San Francisco, CA, USA: ACM. 307–316. DOI: 10.1145/2835776.2835811.

Di Castro, D., L. Lewin-Eytan, Y. Maarek, R. Wolff, and E. Zohar. (2016b). "Enforcing k-anonymity in Web Mail Auditing". In: *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*. ACM. 327–336.

Dumais, S., E. Cutrell, 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: *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM. 72–79.

Dwork, C. (2008). "Differential Privacy: A Survey of Results". In: *Proceedings of the 5th International Conference on Theory and Applications of Models of Computation (TAMC 2008)*. Vol. 4978. *Lecture Notes in Computer Science*. Xi'an, China: Springer Verlag. 1–19. URL: https://www.microsoft.com/en-us/research/publication/differential-privacy-a-survey-of-results/.

Elsayed, T. and D. W. Oard. (2006). "Modeling Identity in Archival Collections of Email: A Preliminary Study." In: *Proceedings of the 3rd Conference on Email and Anti-Spam (CEAS 2006)*. 95–103.

European Parliament and Council of the European Union. (2016). "General Data Protection Regulation". URL: http://data.europa.eu/eli/reg/2016/679/oj.

Fang, Z., A. Agarwal, and T. Joachims. (2019). "Intervention Harvesting for Context-Dependent Examination-Bias Estimation". In: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2019)*. Paris, France: Association for Computing Machinery. 825–834. DOI: 10.1145/3331184.3331238.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

Feild, H. A., J. Allan, and J. Glatt. (2011). "CrowdLogging: distributed, private, and anonymous search logging". In: *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. 375–384.

Fette, I., N. Sadeh, and A. Tomasic. (2007). "Learning to Detect Phishing Emails". In: *Proceedings of the 16th International Conference on World Wide Web (WWW 2007)*. Banff, Alberta, Canada: ACM. 649–656. DOI: 10.1145/1242572.1242660.

Foley, J., M. Zhang, M. Bendersky, and M. Najork. (2018). "Semantic Location in Email Query Suggestion". In: *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*. 977–980. DOI: 10.1145/3209978.3210116.

Gamzu, I., Z. S. Karnin, Y. Maarek, and D. Wajc. (2015). "You Will Get Mail! Predicting the Arrival of Future Email". In: *Proceedings of the 24th International Conference on World Wide Web Companion, WWW 2015, Florence, Italy, May 18-22, 2015 - Companion Volume*. 1327–1332. DOI: 10.1145/2740908.2741694.

Gamzu, I., L. Lewin-Eytan, and N. Silberstein. (2018). "Unsubscription: A Simple Way to Ease Overload in Email". In: *Proceedings of the 11th ACM International Conference on Web Search and Data Mining (WSDM 2018)*. Marina Del Rey, CA, USA: ACM. 189–197. DOI: 10.1145/3159652.3159698.

Ganin, Y., E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. (2016). "Domain-Adversarial Training of Neural Networks". *The Journal of Machine Learning Research*. 17(1): 2096–2030.

Gao, N., M. Dredze, and D. W. Oard. (2016). "Knowledge base population for organization mentions in email". In: *Proceedings of the 5th Workshop on Automated Knowledge Base Construction*. 24–28.

Gibney, E. (2017). "Ethics of Internet Research Trigger Scrutiny". *Nature News*. 550(7674): 16.

Gmail Help. (2018). "Search operators you can use with Gmail". URL: https://support.google.com/mail/answer/7190?hl=en.

Google Search. (2018). "How Search Organizes Information". URL: https://www.google.com/search/howsearchworks/crawling-indexing/.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

Graham, P. (2003). "Better Bayesian Filtering". URL: http://www.paulgraham.com/better.html.

Grammarly. (2018). "5 Ways Grammarly Helps You Learn While You Write". URL: https://www.grammarly.com/blog/grammarly-helps-you-learn/.

Graus, D., P. N. Bennett, R. W. White, and E. Horvitz. (2016). "Analyzing and Predicting Task Reminders". In: *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization. UMAP '16*. Halifax, Nova Scotia, Canada: ACM. 7–15. DOI: 10.1145/2930238.2930239.

Grbovic, M., G. Halawi, Z. S. Karnin, and Y. Maarek. (2014). "How Many Folders Do You Really Need?: Classifying Email into a Handful of Categories". In: *Proceedings of the 23rd ACM International Conference on Information and Knowledge Management, CIKM 2014, Shanghai, China, November 3-7, 2014*. 869–878. DOI: 10.1145/2661829.2662018.

Gupta, J., Z. Qin, M. Bendersky, and D. Metzler. (2019a). "Personalized Online Spell Correction for Personal Search". In: *The World Wide Web Conference. WWW '19*. San Francisco, CA, USA: ACM. 2785–2791. DOI: 10.1145/3308558.3313706.

Gupta, R., R. Kondapally, and S. Guha. (2019b). "Large-Scale Information Extraction from Emails with Data Constraints". In: *International Conference on Big Data Analytics*. Springer. 124–139.

Hangal, S., M. S. Lam, and J. Heer. (2011). "MUSE: Reviving Memories Using Email Archives". In: *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology. UIST '11*. Santa Barbara, California, USA: ACM. 75–84. DOI: 10.1145/2047196.2047206.

Hastie, T., R. Tibshirani, and J. Friedman. (2009). *The Elements of Statistical Learning*. Springer. DOI: 10.1007/978-0-387-84858-7.

Hawking, D. (2010). "Enterprise Search". In: *Modern Information Retrieval, 2nd Edition*. Ed. by R. Baeza-Yates and B. Ribeiro-Neto. Addison-Wesley. 645–686. URL: http://david-hawking.net/pubs/ModernIR2_Hawking_chapter.pdf.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

He, J., J. Zeng, and T. Suel. (2010). "Improved Index Compression Techniques for Versioned Document Collections". In: *Proceedings of the 19th ACM International Conference on Information and Knowledge Management. CIKM '10*. Toronto, ON, Canada: ACM. 1239–1248. DOI: 10.1145/1871437.1871594.

Heer, J. (2005). "Exploring Enron: A Sketch of Visual Data Mining of Email". In: *Email Archive Visualization Workshop, University of Maryland*.

Horovitz, M., L. Lewin-Eytan, A. Libov, Y. Maarek, and A. Raviv. (2017). "Mailbox-Based vs. Log-Based Query Completion for Mail Search". In: *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM. 937–940.

Hu, J., R. J. Passonneau, and O. Rambow. (2009). "Contrasting the interaction structure of an email and a telephone corpus: A machine learning approach to annotation of dialogue function units". In: *Proceedings of the SIGDIAL 2009 Conference*. 357–366.

Jiang, L., Y. Kalantidis, L. Cao, S. Farfade, J. Tang, and A. G. Hauptmann. (2017). "Delving Deep into Personal Photo and Video Search". In: *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM. 801–810.

Joachims, T. (2002). "Optimizing Search Engines Using Clickthrough Data". In: *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM. 133–142.

Joachims, T., L. Granka, B. Pan, H. Hembrooke, F. Radlinski, and G. Gay. (2007). "Evaluating the Accuracy of Implicit Feedback from Clicks and Query Reformulations in Web Search". *ACM Trans. Inf. Syst.* 25(2). DOI: 10.1145/1229179.1229181.

Joachims, T., A. Swaminathan, and T. Schnabel. (2017). "Unbiased Learning-to-rank With Biased Feedback". In: *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM. 781–789.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

Kamvar, M., M. Kellar, R. Patel, and Y. Xu. (2009). "Computers and iPhones and Mobile Phones, Oh My!: A Logs-based Comparison of Search Users on Different Devices". In: *Proceedings of the 18th international conference on World wide web*. 801–810.

Kannan, A., K. Kurach, S. Ravi, T. Kaufmann, A. Tomkins, B. Miklos, G. Corrado, L. Lukács, M. Ganea, P. Young, and V. Ramavajjala. (2016). "Smart Reply: Automated Response Suggestion for Email". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*. 955–964. DOI: 10.1145/2939672.2939801.

Kaushal, G. (2016). "Inbox by Gmail: Find Answers Even Faster". URL: https://www.blog.google/products/gmail/inbox-by-gmail-find-answers-even-faster/.

Kiritchenko, S. and S. Matwin. (2001). "Email Classification with Co-training". In: *Proceedings of the 2001 Conference of the Center for Advanced Studies on Collaborative Research (CASCON 2001)*. Toronto, Ontario, Canada: IBM Press. 301–312. URL: http://dl.acm.org/citation.cfm?id=782096.782104.

Kleinberg, J. M. (1999). "Authoritative Sources in a Hyperlinked Environment". *J. ACM*. 46(5): 604–632. DOI: 10.1145/324133.324140.

Kokkalis, N., T. Köhn, C. Pfeiffer, D. Chornyi, M. S. Bernstein, and S. R. Klemmer. (2013). "EmailValet: Managing Email Overload Through Private, Accountable Crowdsourcing". In: *Proceedings of the 2013 conference on Computer supported cooperative work*. ACM. 1291–1300.

Konečný, J., H. B. McMahan, D. Ramage, and P. Richtarik. (2016). "Federated Optimization: Distributed Machine Learning for On-Device Intelligence". arXiv: 1610.02527 [cs.LG].

Koren, Y., E. Liberty, Y. Maarek, and R. Sandler. (2011). "Automatically Tagging Email by Leveraging Other Users' Folders". In: *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '11*. San Diego, California, USA: ACM. 913–921. DOI: 10.1145/2020408.2020560.

Kruschwitz, U. and C. Hull. (2017). "Searching the Enterprise". *Foundations and Trends® in Machine Learning*. 11(1): 1–142. DOI: 10.1561/1500000053.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

Kulkarni, A., J. Teevan, K. M. Svore, and S. T. Dumais. (2011). "Understanding Temporal Query Dynamics". In: *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining. WSDM '11.* Hong Kong, China: ACM. 167–176. DOI: 10.1145/1935826.1935862.

Kuzi, S., D. Carmel, A. Libov, and A. Raviv. (2017). "Query Expansion for Email Search". In: *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2017).* ACM. Shinjuku, Tokyo, Japan. 849–852. DOI: 10.1145/3077136.3080660.

Kveton, B., C. Szepesvari, Z. Wen, and A. Ashkan. (2015). "Cascading Bandits: Learning to Rank in the Cascade Model". In: *International Conference on Machine Learning.* 767–776.

Lam, D. S. (2002). "Exploiting E-mail Structure to Improve Summarization". *PhD thesis.* Massachusetts Institute of Technology.

Lampert, A., R. Dale, and C. Paris. (2009). "Segmenting Email Message Text into Zones". In: *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing.* Singapore: Association for Computational Linguistics. 919–928. URL: https://www.aclweb.org/anthology/D09-1096.

Lampert, A., R. Dale, and C. Paris. (2010). "Detecting Emails Containing Requests for Action". In: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics.* Los Angeles, California: Association for Computational Linguistics. 984–992. URL: https://www.aclweb.org/anthology/N10-1142.

Lempel, R. and S. Moran. (2001). "SALSA: The Stochastic Approach for Link-Structure Analysis". *ACM Trans. Inf. Syst.* 19(2): 131–160. DOI: 10.1145/382979.383041.

Li, C., M. Zhang, M. Bendersky, H. Deng, D. Metzler, and M. Najork. (2019a). "Multi-view Embedding-based Synonyms for Personal Search". In: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2019).* Paris, France: ACM. 575–584. DOI: 10.1145/3331184.3331250.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

Li, P., Z. Qin, X. Wang, and D. Metzler. (2019b). "Combining Decision Trees and Neural Networks for Learning-to-Rank in Personal Search". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '19*. Anchorage, AK, USA: Association for Computing Machinery. 2032–2040. DOI: 10.1145/3292500.3330676.

Liang, J., L. Jiang, L. Cao, Y. Kalantidis, L. Li, and A. G. Hauptmann. (2019). "Focal Visual-Text Attention for Memex Question Answering". *IEEE Transactions on Pattern Analysis and Machine Intelligence*: 1–1. DOI: 10.1109/TPAMI.2018.2890628.

Liao, S. (2010). "Yahoo Mail is still scanning your emails for data to sell to advertisers". URL: https://www.theverge.com/2018/8/28/17792522/yahoo-mail-email-scan-data-advertisers-opt-out.

Lin, C.-C., D. Kang, M. Gamon, and P. Pantel. (2018). "Actionable Email Intent Modeling with Reparametrized RNNs". In: *Thirty-Second AAAI Conference on Artificial Intelligence*.

Linden, G., B. Smith, and J. York. (2003). "Amazon.com Recommendations: Item-to-item Collaborative Filtering". *IEEE Internet Computing*. 7(1): 76–80.

Litmus Email Analytics. (2019). "The 2019 Email Client Market Share". URL: https://litmus.com/blog/infographic-the-2019-email-client-market-share.

Littlestone, N. (1988). "Learning Quickly when Irrelevant Attributes Abound: A New Linear-threshold Algorithm". *Machine Learning*. 2(4): 285–318.

Maarek, Y. (2017). "Web Mail is not Dead!: It's Just Not Human Anymore". In: *Proceedings of the 26th International Conference on World Wide Web (WWW 2017)*. International World Wide Web Conferences Steering Committee. Perth, Australia: Association for Computing Machinery. 5. DOI: 10.1145/3038912.3050916.

McMahan, B. and D. Ramage. (2017). "Federated Learning: Collaborative Machine Learning without Centralized Training Data". URL: https://ai.googleblog.com/2017/04/federated-learning-collaborative.html.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

Mei, Q. and K. Church. (2008). "Entropy of Search Logs: How Hard is Search? With Personalization? With Backoff?" In: *Proceedings of the 2008 International Conference on Web Search and Data Mining (WSDM 2008)*. Palo Alto, California, USA: Association for Computing Machinery. 45–54. DOI: 10.1145/1341531.1341540.

Microsoft 365. (2020). *Manage mailbox auditing*. URL: https://docs.microsoft.com/en-us/microsoft-365/compliance/enable-mailbox-auditing.

Mikolov, T., M. Karafiát, L. Burget, J. Černocký, and S. Khudanpur. (2010). "Recurrent Neural Network Based Language Model". In: *11th Annual Conference of the International Speech Communication Association (INTERSPEECH 2010)*. Makuhari, Chiba, Japan.

Minkov, E., R. C. Wang, and W. W. Cohen. (2005). "Extracting Personal Names from Email: Applying Named Entity Recognition to Informal Text". In: *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. 443–450.

Mitra, B. and N. Craswell. (2018). "An Introduction to Neural Information Retrieval". *Foundations and Trends® in Information Retrieval*. 13(1): 1–126.

Mitra, B., E. Nalisnick, N. Craswell, and R. Caruana. (2016). "A Dual Embedding Space Model for Document Ranking". arXiv: 1602.01137 [cs.IR].

Mohammad, S. M. and T. Yang. (2011). "Tracking Sentiment in Mail: How Genders Differ on Emotional Axes". In: *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis. WASSA '11*. Portland, Oregon, USA: Association for Computational Linguistics. 70–79. URL: http://dl.acm.org/citation.cfm?id=2107653.2107662.

Mukherjee, S., S. (Mukherjee, M. Hasegawa, A. Hassan Awadallah, and R. W. White. (2020). "Smart To-Do : Automatic Generation of To-Do List from Emails". In: *Annual Conference of the Association for Computational Linguistics (ACL 2020)*.

Najork, M. (2009). "Web Spam Detection." *Encyclopedia of Database Systems*. 1: 3520–3523.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

Najork, M. (2018). "Training On-Device Ranking Models from Cross-User Interactions in a Privacy-Preserving Fashion". In: *Proceedings of the First Biennial Conference on Design of Experimental Search & Information Retrieval Systems, Bertinoro, Italy, August 28-31, 2018*. 108. URL: http://ceur-ws.org/Vol-2167/short11.pdf.

Najork, M. A. (2007). "Comparing the Effectiveness of Hits and Salsa". In: *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management. CIKM '07*. Lisbon, Portugal: ACM. 157–164. DOI: 10.1145/1321440.1321465.

Naragon, K. (2018). "We Still Love Email, But We're Spreading the Love with Other Channels". URL: https://theblog.adobe.com/love-email-but-spreading-the-love-other-channels/.

Narang, K., S. T. Dumais, N. Craswell, D. Liebling, and Q. Ai. (2017). "Large-Scale Analysis of Email Search and Organizational Strategies". In: *Proceedings of the 2017 Conference on Computer Human Information Interaction and Retrieval. CHIIR '17*. Oslo, Norway: ACM. 215–223. DOI: 10.1145/3020165.3020175.

Naranyanan, A. and V. Shmatikov. (2008). "Robust De-anonymization of Large Datasets". In: *Proceedings of the 2008 IEEE Symposium on Security and Privacy, May 2008*.

Nowak, P. (2010). "Privacy commissioner reviewing Google Buzz". URL: https://www.cbc.ca/news/technology/privacy-commissioner-reviewing-google-buzz-1.899267.

Oard, D., W. Webber, D. Kirsch, and S. Golitsynskiy. (2015). "Avocado Research Email Collection". *Philadelphia: Linguistic Data Consortium*.

Oard, D. W. and W. Webber. (2013). "Information retrieval for e-discovery". *Foundations and Trends in Information Retrieval*. 7(2-3): 99–237.

Ogilvie, P. and J. Callan. (2005). "Experiments with Language Models for Known-Item Finding of E-mail Messages". In: *Proceedings of the Fourteenth Text REtrieval Conference (TREC 2005)*. National Institute of Standards and Technology. Gaithersburg, Maryland, USA.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

Olston, C. and M. Najork. (2010). "Web Crawling". *Foundations and Trends® in Machine Learning*. 4(3): 175–246. DOI: 10.1561/1500000017.

Oosterhuis, H. and M. de Rijke. (2018). "Ranking for Relevance and Display Preferences in Complex Presentation Layouts". In: *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM. 845–854.

Oosterhuis, H. and M. de Rijke. (2021). "Unifying Online and Counterfactual Learning to Rank: A Novel Counterfactual Estimator that Effectively Utilizes Online Interventions". In: *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 463–471.

Pantel, P. and D. Lin. (1998). "Spamcop: A Spam Classification & Organization Program". In: *Proceedings of AAAI-98 Workshop on Learning for Text Categorization*. 95–98.

Perronnin, F. C. (2009). "Statistical Language-model Based System for Detection of Missing Attachments".

Radlinski, F., M. Kurup, and T. Joachims. (2008). "How Does Clickthrough Data Reflect Retrieval Quality?" In: *Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM 2008)*. Napa Valley, California, USA: ACM. 43–52. DOI: 10.1145/1458082.1458092.

Ramarao, P., S. Iyengar, P. Chitnis, R. Udupa, and B. Ashok. (2016). "InLook: Revisiting Email Search Experience". In: *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR '16*. Pisa, Italy: ACM. 1117–1120. DOI: 10.1145/2911451.2911458.

Rambow, O., L. Shrestha, J. Chen, and C. Lauridsen. (2004). "Summarizing Email Threads". In: *Proceedings of HLT-NAACL 2004*. USA. 105–108.

Ramzan, Z. (2010). "Phishing Attacks and Countermeasures". In: *Handbook of Information and Communication Security*. Springer. 433–448.

Ravi, S. and Q. Diao. (2016). "Large Scale Distributed Semi-supervised Learning Using Streaming Approximation". In: *Artificial Intelligence and Statistics*. 519–528.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

Rogers, S. K. (2016). "Item-to-item Recommendations at Pinterest". In: *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM. 393–393.

Sahami, M., S. Dumais, D. Heckerman, and E. Horvitz. (1998). "A Bayesian Approach to Filtering Junk E-mail". In: *Proceedings of AAAI-98 Workshop on Learning for Text Categorization*. Vol. 62. Madison, Wisconsin. 98–105.

Samarati, P. and L. Sweeney. (1998). "Protecting Privacy When Disclosing Information: k-anonymity and its Enforcement Through Generalization and Suppression". *Tech. rep.* technical report, SRI International.

Santos, R. L. T., C. Macdonald, and I. Ounis. (2015). "Search Result Diversification". *Foundations and Trends® in Information Retrieval*. 9(1): 1–90. DOI: 10.1561/1500000040.

Sayed, M. F., W. Cox, J. L. Rivera, C. Christian-Lamb, M. Iqbal, D. W. Oard, and K. Shilton. (2020). "A Test Collection for Relevance and Sensitivity". In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1605–1608.

Schütze, H., C. D. Manning, and P. Raghavan. (2008). *Introduction to Information Retrieval*. Vol. 39. Cambridge University Press.

Sculley, D. and G. Wachman. (2007). "Relaxed Online SVMs in the TREC Spam Filtering Track." In: *Proceedings of the Sixteenth Text REtrieval Conference (TREC 2007)*.

Segal, R. B. and J. O. Kephart. (2000). "Swiftfile: An Intelligent Assistant for Organizing E-mail". In: *AAAI 2000 Spring Symposium on Adaptive User Interfaces*. Palo Alto, CA, USA: AAAI. 107–112.

Shao, J., S. Ji, and T. Yang. (2019). "Privacy-aware Document Ranking with Neural Signals". In: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2019)*. Paris, France: Association for Computing Machinery. 305–314. DOI: 10.1145/3331184.3331189.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

Shen, J., M. Karimzadehgan, M. Bendersky, Z. Qin, and D. Metzler. (2018). "Multi-task Learning for Email Search Ranking with Auxiliary Query Clustering". In: *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM. 2127–2135.

Shen, J., O. Brdiczka, and J. Liu. (2013). "Understanding Email Writers: Personality Prediction from Email Messages". In: *Proceedings of the 21st International Conference on User Modeling, Adaptation, and Personalization (UMAP 2013)*. Ed. by S. Carberry, S. Weibelzahl, A. Micarelli, and G. Semeraro. Vol. 7899. *Lecture Notes in Computer Science*. Rome, Italy: Springer. 318–330. DOI: https://doi.org/10.1007/978-3-642-38844-6_29.

Sheng, Y., S. Tata, J. B. Wendt, J. Xie, Q. Zhao, and M. Najork. (2018). "Anatomy of a Privacy-Safe Large-Scale Information Extraction System Over Email". In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD 2018)*. London, United Kingdom: ACM. 734–743. DOI: 10.1145/3219819.3219901.

Silvestri, F. (2010). "Mining Query Logs: Turning Search Usage Data into Knowledge". *Foundations and Trends® in Machine Learning*. 4(1 & 2): 1–174. DOI: 10.1561/1500000013.

Singh, A. and T. Joachims. (2018). "Fairness of Exposure in Rankings". In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM. 2219–2228.

Sordoni, A., M. Galley, M. Auli, C. Brockett, Y. Ji, M. Mitchell, J.-Y. Nie, J. Gao, and B. Dolan. (2015). "A Neural Network Approach to Context-Sensitive Generation of Conversational Responses". In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Denver, Colorado: Association for Computational Linguistics. 196–205. DOI: 10.3115/v1/N15-1020.

Spirin, N. and J. Han. (2012). "Survey on Web Spam Detection: Principles and Algorithms". *ACM SIGKDD Explorations Newsletter*. 13(2): 50–64.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

Sun, Y., L. Garcia-Pueyo, J. B. Wendt, M. Najork, and A. Broder. (2018). "Learning Effective Embeddings for Machine Generated Emails with Applications to Email Category Prediction". In: *2018 IEEE International Conference on Big Data (Big Data)*. IEEE. 1846–1855.

Sutskever, I., O. Vinyals, and Q. V. Le. (2014). "Sequence to sequence learning with neural networks". *Advances in neural information processing systems*. 27: 3104–3112.

Sutton, C. and A. McCallum. (2012). "An Introduction to Conditional Random Fields". *Foundations and Trends® in Machine Learning*. 4(4): 267–373.

Swaminathan, S., R. Fok, F. Chen, T.-H. (Huang, I. Lin, R. Jadvani, W. S. Lasecki, and J. P. Bigham. (2017). "WearMail: On-the-Go Access to Information in Your Email with a Privacy-Preserving Human Computation Workflow". In: *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology. UIST '17*. Québec City, QC, Canada: ACM. 807–815. DOI: 10.1145/3126594.3126603.

Tang, J. C., E. Wilcox, J. A. Cerruti, H. Badenes, S. Nusser, and J. Schoudt. (2008). "Tag-It, Snag-It, or Bag-It: Combining Tags, Threads, and Folders in e-Mail". In: *CHI '08 Extended Abstracts on Human Factors in Computing Systems*. 2179–2194.

Task Force on Technical Approaches for Email Archives. (2018). *The Future of Email Archives*. URL: https://www.clir.org/pubs/reports/pub175/.

Tata, S., A. Popescul, M. Najork, M. Colagrosso, J. Gibbons, A. Green, A. Mah, M. Smith, D. Garg, C. Meyer, and R. Kan. (2017). "Quick Access: Building a Smart Experience for Google Drive". In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2017)*. Halifax, NS, Canada. 1643–1651. DOI: 10.1145/3097983.3098048.

Taylor, B., D. Fingal, and D. Aberdeen. (2007). "The War Against Spam: A Report From the Front Line". In: *NIPS 2007 Workshop on Machine Learning in Adversarial Environments for Computer Security*.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

The Radicati Group, Inc. (2015). "Email Statistics Report 2015-2019". URL: https://www.radicati.com/wp/wp-content/uploads/2015/02/Email-Statistics-Report-2015-2019-Executive-Summary.pdf.

The Radicati Group, Inc. (2018). "Email Statistics Report 2018-2022". URL: https://www.radicati.com/wp/wp-content/uploads/2018/01/Email_Statistics_Report,_2018-2022_Executive_Summary.pdf.

The Radicati Group, Inc. (2019). "Email Statistics Report 2019-2023". URL: https://www.radicati.com/wp/wp-content/uploads/2018/12/Email-Statistics-Report-2019-2023-Executive-Summary.pdf.

Tran, B., M. Karimzadehgan, R. K. Pasumarthi, M. Bendersky, and D. Metzler. (2019). "Domain Adaptation for Enterprise Email Search". In: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. SIGIR'19*. Paris, France: ACM. 25–34. DOI: 10.1145/3331184.3331204.

Tsotsis, A. (2011). "ComScore Says You Don't Got Mail: Web Email Usage Declines, 59% Among Teens!" URL: https://techcrunch.com/2011/02/07/comscore-says-you-dont-got-mail-web-email-usage-declines-59-among-teens/.

Turtle, H. and J. Flood. (1995). "Query evaluation: strategies and optimizations". *Information Processing & Management*. 31(6): 831–850.

US Department of Health and Human Services. (2012). "Guidance regarding methods for de-identification of protected health information in accordance with the Health Insurance Portability and Accountability Act (HIPAA) Privacy Rule". URL: https://www.hhs.gov/hipaa/for-professionals/privacy/special-topics/de-identification/index.%20html.

Uzuner, Ö., Y. Luo, and P. Szolovits. (2007). "Evaluating the state-of-the-art in automatic de-identification". *Journal of the American Medical Informatics Association*. 14(5): 550–563.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

Van Gysel, C., B. Mitra, M. Venanzi, R. Rosemarin, G. Kukla, P. Grudzien, and N. Cancedda. (2017). "Reply With: Proactive Recommendation of Email Attachments". In: *Proceedings of the 2017 ACM Conference on Information and Knowledge Management (CIKM 2017)*. Singapore, Singapore: Association for Computing Machinery. 327–336. DOI: 10.1145/3132847.3132979.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. (2017). "Attention is all you need". In: *Advances in Neural Information Processing Systems*. 5998–6008.

Wan, S. and K. McKeown. (2004). "Generating Overview Summaries of Ongoing Email Thread Discussions". In: *Proceedings of the 20th International Conference on Computational Linguistics*. Association for Computational Linguistics. 549–555.

Wang, X., M. Bendersky, D. Metzler, and M. Najork. (2016a). "Learning to Rank with Selection Bias in Personal Search". In: *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2016)*. Pisa, Italy. 115–124. DOI: 10.1145/2911451.2911537.

Wang, X., N. Golbandi, M. Bendersky, D. Metzler, and M. Najork. (2018). "Position Bias Estimation for Unbiased Learning to Rank in Personal Search". In: *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. 610–618.

Wang, Y., D. Yin, L. Jie, P. Wang, M. Yamada, Y. Chang, and Q. Mei. (2016b). "Beyond Ranking: Optimizing Whole-Page Presentation". In: *Proceedings of the 9th ACM International Conference on Web Search and Data Mining (WSDM 2016)*. San Francisco, California, USA: Association for Computing Machinery. 103–112. DOI: 10.1145/2835776.2835824.

Watson Marketing. (2018). "2018 Marketing Benchmark Report: Email and Mobile Metrics for Smarter Marketing". URL: https://www.ibm.com/downloads/cas/L2VNQYQ0.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

Wendt, J. B., M. Bendersky, L. G. Pueyo, V. Josifovski, B. Miklos, I. Krka, A. Saikia, J. Yang, M. Cartright, and S. Ravi. (2016). "Hierarchical Label Propagation and Discovery for Machine Generated Email". In: *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, San Francisco, CA, USA, February 22-25, 2016*. 317–326. DOI: 10.1145/2835776.2835780.

Wenyin, L., G. Huang, L. Xiaoyue, Z. Min, and X. Deng. (2005). "Detection of Phishing Webpages Based on Visual Similarity". In: *Special Interest Tracks and Posters of the 14th International Conference on World Wide Web. WWW '05*. Chiba, Japan: ACM. 1060–1061. DOI: 10.1145/1062745.1062868.

Whittaker, S. (2005). "Supporting Collaborative Task Management in E-mail". *Human–Computer Interaction*. 20(1-2): 49–88. DOI: 10.1080/07370024.2005.9667361. eprint: https://www.tandfonline.com/doi/pdf/10.1080/07370024.2005.9667361.

Whittaker, S., V. Bellotti, and J. Gwizdka. (2006). "Email in personal information management". *Communications of the ACM*. 49(1): 68–73.

Whittaker, S., T. Matthews, J. Cerruti, H. Badenes, and J. Tang. (2011). "Am I Wasting My Time Organizing Email?: A Study of Email Refinding". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2011)*. ACM. Vancouver, BC, Canada. 3449–3458. DOI: 10.1145/1978942.1979457.

Whittaker, S. and C. Sidner. (1997). "Email Overload: Exploring Personal Information Management of Email". *Culture of the Internet*: 277–295.

Wu, Y. (2018). "Smart Compose: Using Neural Networks to Help Write Emails". URL: https://ai.googleblog.com/2018/05/smart-compose-using-neural-networks-to.html.

Xu, X., A. H. Awadallah, S. Dumais, F. Omar, B. Popp, R. Rounthwaite, and F. Jahanbakhsh. (2020). "Understanding User Behavior For Document Recommendation". In: *Proceedings of the Web Conference 2020 (WWW 2020)*. Taipei, Taiwan: Association for Computing Machinery. DOI: 10.1145/3366423.3380071.

The version of record is available at: http://dx.doi.org/10.1561/1500000069

Yue, Y., R. Patel, and H. Roehrig. (2010). "Beyond Position Bias: Examining Result Attractiveness as a Source of Presentation Bias in Clickthrough Data". In: *Proceedings of the 19th International Conference on World Wide Web (WWW 2010)*. Raleigh, North Carolina, USA: Association for Computing Machinery. 1011–1018. DOI: 10.1145/1772690.1772793.

Zamani, H., M. Bendersky, X. Wang, and M. Zhang. (2017). "Situational Context for Ranking in Personal Search". In: *Proceedings of the 26th International Conference on World Wide Web (WWW 2017)*. International World Wide Web Conferences Steering Committee. Perth, Australia: Association for Computing Machinery. 1531–1540. DOI: 10.1145/3038912.3052648.

Zhang, A., L. Garcia Pueyo, J. B. Wendt, M. Najork, and A. Broder. (2017). "Email Category Prediction". In: *Companion Proc. of the 26th International World Wide Web Conference*. 495–503.

Zhang, S., H. Yang, and L. Singh. (2016). "Anonymizing Query Logs by Differential Privacy". In: *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2016)*. Pisa, Italy: Association for Computing Machinery. 753–756. DOI: 10.1145/2911451.2914732.

Zhang, Y., W. Chen, D. Wang, and Q. Yang. (2011). "User-Click Modeling for Understanding and Predicting Search-Behavior". In: *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2011)*. San Diego, California, USA: Association for Computing Machinery. 1388–1396. DOI: 10.1145/2020408.2020613.

Zhao, Q., P. N. Bennett, A. Fourney, A. L. Thompson, S. Williams, A. D. Troy, and S. T. Dumais. (2018). "Calendar-Aware Proactive Email Recommendation". In: *The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2018)*. Ann Arbor, MI, USA: ACM. 655–664. DOI: 10.1145/3209978.3210001.

Zhuang, J. and Y. Liu. (2019). "PinText: A Multitask Text Embedding System in Pinterest". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM. 2653–2661.