RESHAPING REMINISCENCE, WEB BROWSING AND WEB SEARCH USING PERSONAL DIGITAL ARCHIVES

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Sudheendra Hangal

December 2012

This dissertation is online at: http://purl.stanford.edu/sj530gp9264

Includes supplemental files:

1. *(muse-snapshot.zip)*
2. *(muse-survey-responses.txt)*
3. *(slant-queries.xlsx)*

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Monica Lam, Primary Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Jeffrey Heer

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Terry Winograd

Approved for the Stanford University Committee on Graduate Studies.

Patricia J. Gumport, Vice Provost Graduate Education

Abstract

Millions of consumers are accumulating logs of their social interactions on the Internet. These logs chronicle people's lives at a level never before possible. Instead of just retaining a few old pictures, letters and mementos from their personal histories as in previous generations, the participants of the digital age can have access to their detailed thoughts, interactions and communications over years or decades. Over the long term, these archives can be a wonderful source of memories and can capture deeply meaningful experiences and stories.

However, many technical barriers obstruct the practical utility of these archives. Raw logs of activity are voluminous and not very interesting, unless people have engaging sense-making tools that help them easily organize the archive, spot patterns, view summaries, and navigate content. To this end, we design, implement and evaluate a system called Muse (Memories USing Email) which provides four novel types of cues to help spot interesting trends and messages in a large-scale email archive. These cues act as salient entry points into the archives, which can then be navigated with an interface that supports rapid browsing of messages. Muse is publicly available and has been downloaded over 6,000 times to date. Our user reports indicate a range of possible benefits from tools like Muse, from utilitarian ones such as summarizing work or backing up attachments, to reminiscence and remembering family events and grad school years with nostalgia, to reinforcing confidence, renewing relationships and playing memory games. In addition, Muse provides convenient ways for archival organizations to process the email archives of prominent individuals and to provide them to researchers, thereby unlocking the historical value embedded in these archives.

We also propose a new class of *experience-infused* applications that provide powerful, privacy-respecting forms of personalization with the help of personal archives. We demonstrate two important examples of such applications. The first is an experience-infused web browser that annotates web pages in real time as they are loaded, highlighting terms that the user has encountered before. Our studies find that this technique is useful to personalize crowded web pages and to serendipitously spot connections to things the user may have forgotten about. The second application is experience-infused web search. Here, we propose the idea of personal search engines that bias search results towards domains mentioned in the user's email or Twitter feeds. We find that these results can be used to boost user satisfaction with web search results and provide an analysis of the types of queries for which experience-infused search does well.

Taken together, these applications provide a glimpse of an exciting future where consumers can easily look up history, supplement memory and improve information efficiency, thus putting their archives to work for their own benefit.

For Dada and Appa

Acknowledgements

I would like to first thank my advisor, Monica Lam, whose infectious enthusiasm for research prompted me to return to grad school after several years of working in industry. I am in awe of Monica's infinite energy and boundless optimism, not to mention her skill at turning an incoherent jumble of text into a reasonable-looking paper in the last half-hour before a submission deadline. She has not just tolerated, but encouraged and guided my forays into unknown territory, and I have learned a lot from her.

Many thanks to Jeff Heer for collaborating with me on the MUSE project, as well as for providing feedback on this dissertation. Jeff has inspired me by going beyond publishing research papers to releasing working software systems which feed back into more research. The rest of the Stanford HCI faculty was equally helpful: Terry Winograd, Stu Card and Scott Klemmer always provided friendly advice and served on my university exam committee. Priya Satia chaired the committee and provided me with an idea of how historians work.

I would like to thank my many collaborators, without whom this dissertation would not have been possible: Abhinay Nagpal worked with me on the idea of experience-infused software and was largely responsible for boldly expanding the scope of our project to web search, when I initially had a much smaller idea (to introduce friends at TGIF) in mind. My most serendipitous moment during this work happened when I had a chance meeting with Peter Chan from Stanford Libraries. Peter saw the connection between MUSE and library applications and has provided invaluable advice to the project. Peter, Glynn Edwards, Glen Worthey and Stanford archivist Daniel Hartwig educated me about the field of archives and the digital humanities.

My work with them on the Robert Creeley archives in the library led directly to the idea of the experience-infused browser, as my lack of knowledge about poetry made me wonder how I could look up all the terms on Mr. Creeley's Wikipedia page *en masse* in his archive.

As we began the task of "productizing" Muse beyond a research prototype, my long-term friend and compatriot, Chaiyasit Manovit, jumped in to join me on yet another mission. His presence gives me great reassurance about the future of Muse. Diana MacLean did a lot of the work for both our grouping algorithms, and has been a close friend and source of lively discussions about topics ranging from opera to recipe visualization, as well as a reliable editor for drafts of my writing (including this work). T.J. Purtell provided several ideas for our second grouping algorithm (described in Chapter 3) and ran most of the experiments to generate the associated graphs. Peter Monaco from Xobni provided us an anonymized data set for experiments. Seng Keat Teh built the Facebook app for grouping from photo tags, and Rifat Reza Joyee ran some of the user studies for experience-infused web search. Cindy Chang designed the Muse logo and helped in the graphic design. I had many useful discussions with Rakesh Ghiya, and he was the one who suggested the Google custom search engine when we were stumped about experimental deployments for experience-infused search. Several of the Stanford Knight scholars over the years – including at least Krissy Clark, Maria Teresa Bouza Corral, Phuong Ly, Geoff McGhee and T. Christian Miller – provided great ideas and encouragement for Muse in the field of journalism.

Darlene Hadding livened up the fourth floor of Gates, apart from providing efficient administrative support. Jiwon Seo was always a ready source or destination for rants about programming languages. I had great interactions with many other friends, including Chris Brigham, Willem Bult, Ben Dodson, Phil Guo, Nikhil Handigol, Te-Yuan Huang, Neil Patel, Seok-won Seong, Ankur Taly, Ian Vo, KK Yap, the entire Stanford HCI and Mobisocial crews, and probably several others I've missed. I will always think of you guys, and perhaps re-share classic email messages once in a while.

Funding for this work was generously provided by a Stanford School of Engineering Fellowship, NSF POMI Expedition grant 0832820 and the sponsors of the Stanford

Mobisocial Laboratory.

Words cannot express my gratitude to Jayu, Rishabh and Basant, so I will not try. Suffice to say that they have made the last few years my happiest and most fulfilling. We had a wonderful time living on the Stanford campus, biking around the beautiful surroundings, and enjoying its stimulating intellectual environment. I would like to thank Shri Pramod Marathe, my guru on the classical harmonium, for many delightful musical evenings, and Ustad Amjad Ali Khansaab for the unforgettable opportunity to learn music from him at Stanford for a whole quarter.

I am lucky to have a large and extended family and circle of close friends – too large to list here, but they will recognize themselves – and I would like to thank each one of them for their support and friendship. I am especially grateful to my father, Gururaj N. Hangal, who got me interested in engineering, and my father-in-law, Prof. Krishna S. Deshpande, who made me realize the value of the humanities. I dedicate this thesis to them.

Sudheendra Hangal
Stanford, CA
hangal@stanfordalumni.org

Contents

List of Figures

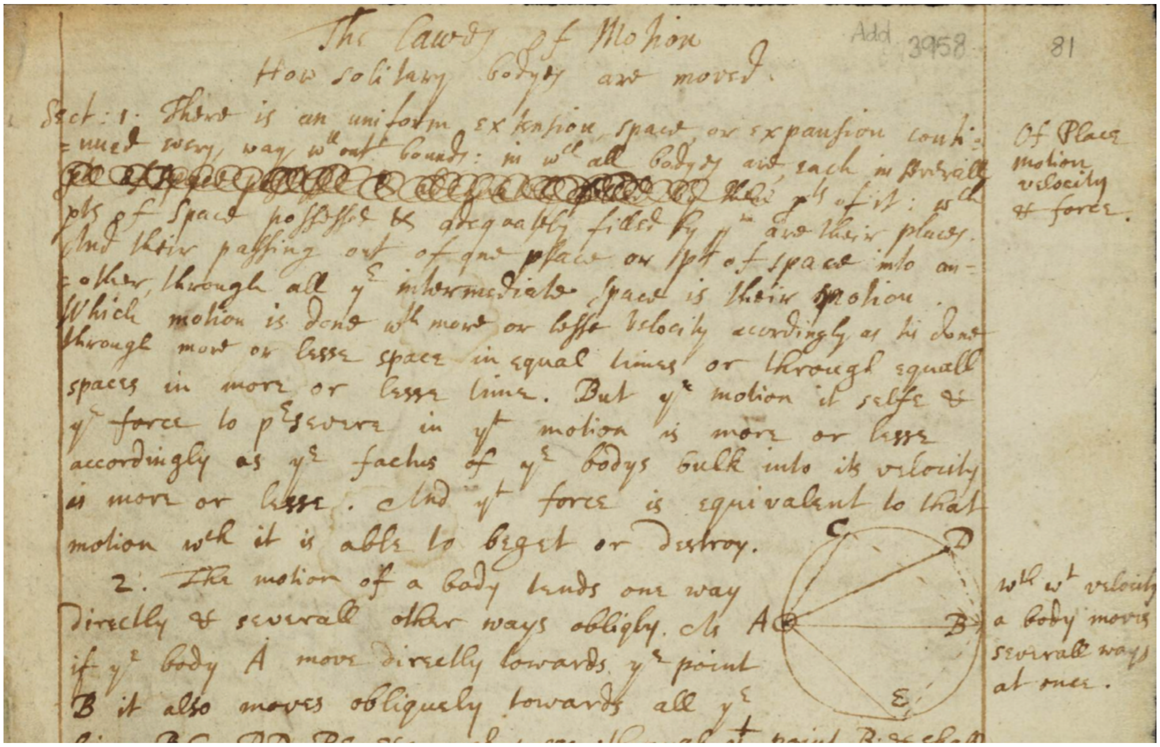
Chapter 1

Introduction

> *"Consider a future device for individual use, which is a sort of mechanized private file and library. It needs a name, and, to coin one at random, "memex" will do. A memex is a device in which an individual stores all his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility. It is an enlarged intimate supplement to his memory."*
>
> *- As We May Think, by Vannevar Bush*
> *The Atlantic, May 1945.*

Vannevar Bush presaged the age of digital archives with his vision of the memex [20]. In the past, the personal archives of prominent people, consisting of materials like papers, documents and diaries, have been valuable tools of record. For example, the fragment of Isaac Newton's notebook in Fig. 1.1 captures his thoughts as he was framing the laws of motion. While such personal archives have so far been preserved for a chosen few, the digital age makes it possible for ordinary consumers to easily accumulate logs that chronicle their lives at a level never before possible.

This dissertation considers ways in which personal digital archives may be put to work for the user's own benefit. We have prototyped and studied new user experiences that make use of these archives and may benefit a large number of users as they acquire long-term archives. Ironically, while corporations routinely mine personal data for



The Lawes of Motion
How solitary bodys are moved.

Sect: 1. There is an uniform extension space or expansion conti-
=nued every way wthout bounds: in wch all bodys are each in severall
~~[illegible]~~ their pts of it: wch
pts of space possessed & adequatly filled by them are their places.
And their passing out of one place or part of space into an-
=other, through all ye intermediate space is their motion.
Which motion is done wth more or lesse velocity accordingly as tis done
through more or lesse space in equal times or through equall
spaces in more or lesse time. But ye motion it selfe &
ye force to preserve in yt motion is more or lesse
accordingly as ye factus of ye bodys bulk into its velocity
is more or lesse. And yt force is equivalent to that
motion wch it is able to beget or destroy.

2. The motion of a body tends one way
directly & severall other ways obliqly. As
if ye body A move directly towards ye point
B it also moves obliquely towards all ye

Of Place
motion
velocity
& force.

wth wt velocity
a body moves
severall ways
at once.

Figure 1.1: A fragment from one of Isaac Newton's notebooks during the years 1665-1672. The Newton Papers © Cambridge University Library. Reproduced under CC BY-NC 3.0.

profiling and advertising purposes, there are relatively few tools to help individuals make use of their own data.

In previous generations, people maintained personal histories by retaining a few pictures, letters and mementos, and filled in the rest with their organic – and often fallible – memories. However, the participants of the digital age can have access to many of their detailed thoughts, interactions and communications over years or decades. Over the long term, these archives can be a wonderful source of memories, and capture experiences and stories that are deeply meaningful.

However, there are a host of barriers between these potential uses and what people can actually do with their archives today. For one, long-term archives tend to be voluminous. The relative ease of accumulating digital records is a double-edged sword: along with valuable history, it is easy to collect much incidental or irrelevant material – noise that drowns out the signal. Further, it is not easy to predict the significance of an item at the time it is captured, which may only become apparent with hindsight.

Long-term archives tend to lack good organization because it is tedious to tag many small items, and moreover, to keep the structure consistent over a long period of time. People's digital archives are usually scattered across different computers, programs, services and formats. While archives often contain valuable details relevant to a user's information needs, they are rarely consulted because searching archives in the flow of a task is cumbersome. Browsing archives to get an overall sense of their contents is even harder than searching them.

1.1 Research Overview and Approach

This dissertation explores two primary ways in which users can take advantage of their long-term personal digital archives. First, we consider how people may use their digital archives for reviving memories. As an extension, we discuss how archival organizations can process and host archives of historical importance. Second, we investigate ways in which personal archives can be used to re-invent the experience of everyday applications such as web browsing and web search.

For both of these approaches, we followed rounds of ideation with an iterative system design and implementation approach. We employed a process of ongoing user testing with prototypes and designed specific techniques in response to feedback about shortcomings. When the designs were fairly stable, we did controlled studies with targetted groups of users. We also elicited qualitative feedback with the goal of understanding design considerations for future applications.

Throughout this process, we ensured that all our software was publicly available on the web for anyone to download and use. To this end, we ensured that most of our techniques were lightweight enough to run on end-user's computers, and the software robust enough for relatively broad use. So far, over 6,000 users have downloaded various versions of Muse. Users were encouraged to fill out a feedback form after they had used the software. This allowed us to get feedback from real users "in the wild".

1.2 Why Email?

At an experimental level, our work is focused on email archives, although parts of it employ other social media like Twitter, Facebook and instant messaging. While our specific techniques have varying degrees of applicability in other settings, we chose to focus on email because it is one of the Internet's most enduring "killer applications" with an estimated 2.1 billion users and 3.3 billion accounts worldwide in 2012 [108, 109]. With the advent of free email services with virtually unlimited storage and the "Never delete anything!" mindset pioneered by Gmail, email is also an excellent archival platform. It is not surprising that the U.S. Library of Congress recommends email as one of the most valuable resources for people to save and archive [130].

We expect that millions of mainstream users will amass large email repositories in the future. Already, it is common to find people, especially in universities, with multi-decade email archives. This allows the possibility of realistic experiments today for something that is going to be increasingly common in the coming years.

There is a lot of interest in the HCI community and the Quantified Self movement about innovative devices that help in *life-logging*. Email archives represent the simplest and most pervasive form of life-logging that billions of people already use. It is also the form that is most likely to have been used consistently over a long period of time, which is why studying it can inform the design of other forms of long-term life-logging.

While the media is fond of routinely issuing predictions about the death of email (e.g., [143]), we believe the longevity of email is underestimated. According to recent Pew Internet polls [105, 154], email continues to be the top-ranked Internet application across all age groups in the United States, a position it has occupied since the polls started in 2002. It has been in use for over 40 years because it is an open, federated form of communication. Email protocols inter-operate seamlessly between large public service providers like Gmail, Hotmail and Yahoo, organizations like corporations and universities, and tiny servers hosted by individuals. Email is portable and can be moved from one system or account to another. It can be copied and backed up under a user's control, converted between different formats, and is not hostage to the fortunes of companies that come and go. In fact, lack of inter-operability may be one of the major reasons for the disappearance of formerly popular walled gardens like Compuserve and AOL. It is likely that any medium that displaces email will need to share some of its simple and open characteristics, and that many of the topics discussed in this dissertation will be applicable to any such new system.

Another reason for emphasizing email in our work is that, like physical letters, its ownership is relatively clear – it is more or less understood that people own their email messages. In contrast, the prospect of users storing and using a long-term archive of say, their Facebook or Twitter feeds including their friends' data, raises a variety of social, business, legal and ethical concerns that are not yet well understood.

1.3 Reviving Memories with Email Archives

Our first set of applications with personal digital archives involves users actively using them for reminiscence. Since the first email message was sent across the network in 1971 [135], the usage of email has evolved significantly. In much of the wired world, email is used for many daily activities, for everything from setting up meetings to getting business approvals; from making purchases on the Internet to sending emotional messages of love, joy, and condolence; from sending oneself reminders to sharing an interesting web link with friends, and so on. Unlike blogs, diaries and journals, email archives silently capture our experiences *in situ*, as they arise in our communication, with no additional action needed to record them. Further, email is directed and captures meaningful person-to-person communication, as letters have through much of history.

In the future, we envision that personal digital archives will fundamentally affect the processes of long-term memory, recall and reminiscence. For example, it should be possible for a 30-year old to recall a field trip he made in kindergarten, along with all the details associated with planning the event that are present in his parents' email archives. People should be able to put their archives to work when sharing their fond memories at events like weddings and 50th birthdays. And it should be easy for ordinary users, not just famous celebrities, to consult history and record their memoirs, which would be of great interest at least to their immediate friends and family. The ability to have total and instant recall of their past could also change the way people use their organic memories, much like having the world wide web at their fingertips gradually changes how people think and what they need to remember [124].

1.4 Experience-Infused Software

Our second set of applications proposes new forms of personalization using personal archives. There are many problems with the way personalization is done today by online services such as news, shopping and movie watching sites. All these services attempt to provide personalized experiences, but the user's profile ends up being fragmented across many providers. Personal profiles are not portable to new services. Personalization tends to be opaque, in that the user typically has no idea how her data is being used; its use is governed by arcane privacy policies that few people understand. It is not easy for users to view or control their own personalization profiles. Personalization often intrudes on user privacy by tracking users or making them hand over large amounts of personal information. Can we design paradigms for personalization that serve user needs better?

1.4.1 The Power of Archives

A personal digital archive, especially of social interactions, says a great deal about the owner of the archive, and can often be easily and continuously captured without explicit effort. Over a period of time, the archive accumulates many entities the user associates with, such as people, products, places, events and organizations. It also contains signals about the relative strength of each association. Therefore, it reflects in some sense a detailed profile of the user. While this profile is not perfect – it does not capture all the nuances that a human could – it is a useful approximation that has the benefit of scaling to large amounts of data that can be processed automatically. The personal archive could include data from a variety of personal sources including, for example, email, social networks, spending history, calendar appointments, personal notes, location data and any source of writing authored by a user. In particular, email often contains rich transaction histories in the form of shopping receipts, travel bookings, movie rental histories, and so on.

1.4.2 Core Ideas

This dissertation introduces the idea of *experience-infused* software, which aims to use the power of archives to enable new user experiences. The widespread accumulation of rich personal archives provides an opportunity to re-think the user experience of several applications, all the way from web browsers to mobile phone assistants to a programmer's software development tools. To illustrate the concept of experience-infusion, we have picked two of the most widely used applications – web browsing and web search – and will show in Chapters 5 and 6 how they can benefit the user by connecting to the data in personal archives.

The core ideas of experience-infused software are:

1. Experience-infused software connects many different applications with the history of the user, as captured in her digital archives, allowing personalization and customization with very little user effort.

2. Experience-infused software runs under the direction of the user, allowing the user to aggregate personal data from different sources; at the same time, users can turn on or off parts of their archive at will.

3. Experience-infused software is portable to new applications and services, so that each new piece of software does not have to profile and build up a user model.

As additional considerations, we desire that experience-infused software be privacy-respecting, especially when dealing with parts of the archive that contain private and sensitive information. In many situations, a user will be able to use personalized applications with no one else monitoring the data or usage; in others, she may have the option of revealing abstracted information to a service provider or of using a third party for storage and convenience. However, this decision rests with her. Further, in order to make it easy for users to control their data, we also desire that experience-infused software be transparent, allowing the user to see connections between the personalization results and the underlying archive.

Personalization profiles carried by the user benefit new services because they do not have to build user profiles from scratch. This helps prevent monopolistic services from locking users in based on the difficulty of moving their data from one service to another.

User-side personalization allows for sophisticated algorithms that may be too computationally intensive to run for each user "in the cloud". No cloud data center, however large, can match the aggregate computing power available in billions of powerful clients. This makes it feasible to consider personalization algorithms that may need a large amount of processing. For example, we may be able to perform deep text analysis of web pages relevant to the user that is too expensive to implement for the entire web.

To be sure, not all forms of personalization are amenable to this paradigm. For example, collaborative filtering (the identification of similar people in order to generate recommendations) may be harder to implement than in centralized services today. However, experience-infusion offers the possibility of new experiences in a wide range of applications, and this may attract users to it. We hope to see future work build extensions that allow some of the benefits of other paradigms.

1.5 Contributions

In this dissertation, we show that users derive significant and varied benefits from their email archives, but that they also need tools to help them make sense of these archives. From utilitarian purposes like summarizing work progress, exporting personal messages from a work account and saving attachments, to reminiscing with nostalgia by watching a child grow up over the years or resharing classic messages, to reinforcing confidence, renewing relationships and playing memory games, there are a whole range of benefits which users derive from email archives.

To help with the task of sense-making from large-scale archives, we contribute 4 novel types of cues that appear to be useful for this task: communication activity with automatically inferred groups, occurrence of sentimental words and phrases, image attachments and monthly summaries of the top names in the archive. Along the way, we design novel group detection and text mining algorithms that are more broadly applicable.

We hope to help improve the capture and study of the historical record by making it easier for libraries and other archival organizations to process and host the email archives of eminent individuals. While such archives are frequently collected, they are rarely processed and indexed today, let alone provided to end-users. For the first time, Muse lets libraries and archives provide patrons and the general public partial access to these archives.

We introduce the idea of experience-infused applications and demonstrate a novel web browser that annotates web pages in real time based on the contents of the user's archive. Our studies find that this technique is effective in picking relevant content from crowded web pages or long documents and in highlighting serendipitous connections to topics the user has encountered before.

We propose the idea of experience-infused web search through search indices that are automatically generated from social chatter. These indices provide personalized and spam-free web search results, and we find that they perform comparably with personalized Google search, despite indexing only a fraction of the web. We provide an analysis of the types of queries for which experience-infused search does well. We show that a significant increase in the quality of search results is achievable using social curation.

These systems point to a future where an individual's personal archives are available for instant recall and permeate different applications under the user's control and for her own benefit.

1.6 Dissertation Overview

Chapter 2 describes applications of personal email archives for reviving memories. We present MUSE (Memories USing Email), a system built for this purpose and describe the range of benefits that users derive from it.

Chapter 3 details a novel grouping algorithm embedded in MUSE that generates nested, overlapping social groups from patterns of co-occurrence in email headers and Facebook photo tags. We characterize this algorithm and compare it with existing algorithms.

Chapter 4 discusses applications of MUSE in the settings of libraries and archives. We describe techniques that enable archival organizations to process and screen the email archives of prominent individuals, and to make parts of the archives available to the public.

Chapter 5 introduces the idea of an experience-infused browser that annotates web pages based on a user's archives. We show that the browser is useful to personalize crowded web pages and to serendipitously recall connections that users have forgotten.

Chapter 6 presents experience-infused search, in which we create socially curated and personalized search indices that generate high-quality results for users, while eliminating web spam.

Finally, Chapter 7 provides overall conclusions and lessons from this body of work, and speculates about possible extensions.

Descriptions of background material and related work will be woven into each chapter and discussed in context.

This dissertation integrates, updates and expands upon material previously published at several conferences [52, 53, 51, 90, 107, 76]. The author wishes to think the collaborators who jointly performed much of this work, including Abhinay Nagpal (experience-infused applications), Chaiyasit Manovit (MUSE infrastructure), Peter Chan (library applications), T.J. Purtell, Diana MacLean and Seng Keat Teh (grouping algorithm), Cindy Chang (graphic design) and Rifat Reza Joyee (experience-infused web search).

The supplemental materials submitted with this dissertation include the source code and binaries for the current version of Muse, survey responses for the Muse user study described in section 2.5, and a spreadsheet with details about the experience-infused web search study in section 6.5.

Chapter 2

Reviving Memories Using Email

In this chapter, we consider the use of long-term email archives to revive memories. Email has become a de facto medium of record in the Internet age. Indeed, many people consciously deposit important information into email, knowing they can look it up later, and thereby use their email account as an informal backup device.

Among digital materials, email has several desirable properties from the point of view of long term data storage and archival. First, email archiving (particularly of sent messages) is virtually automatic and requires less effort compared to other forms of digital data such as videos and pictures. Second, email formats have been relatively stable, with a handful of formats that can be translated between one another. For example, the text-based mbox format has been around for a long time, and is easily inter-operable with other email storage formats.

Third, email data volume is relatively small for the richness of information it captures, (on the order of a few gigabytes per decade for most users) compared to data-heavy formats like pictures and videos. Thus email archives are easy to back up and copy, and are less prone to being lost over time as physical formats change, computers are upgraded, and so on. While pictures and videos form vivid and precious snapshots, textual archives complement them by capturing ideas, thoughts and interactions, which is difficult with pictures alone.

Fourth, users tend to engage with email on a consistent basis – again unlike pictures which are often taken at special events and explicitly to capture memories –

and thus email archives capture a regular and detailed record including fairly normal happenings. Of course, we do not expect every event in people's lives to be reflected in their email archives – however, it is possible that reminiscing with email archives will trigger memories of associated events as well. We surmise that for a significant fraction of online users, more characters have been typed into email than into any other application, and these characters, accumulated over a lifetime, can provide a powerful window into history. Zalinger's dissertation [153] explores the role of Gmail as storyworld and its intensive participant interviews confirm the richness of narratives that are embedded in email archives.

For these reasons, email archives can act as a wonderful source of triggers for memories. It is no longer necessary to be a famous individual to have your personal histories preserved for posterity; it is possible for everyone to preserve their histories, perhaps only for the few friends and family who care deeply about them.

However, email archives are also large and messy, and making sense out of a loosely organized pile of tens of thousands of messages is a challenge, particularly since email consists mainly of free-form text along with some metadata and attachments. Indeed, Vannevar Bush noted back in 1945 that *"we can enormously extend the record; yet even in its present bulk we can hardly consult it."* [20] This observation is even more applicable today.

In this chapter, we investigate ways to let end users make sense of their own email archives. Our goal is to study questions such as: What kinds of information do users find valuable when detailed, day-by-day records of their communication are available? What techniques enable users to conveniently browse their life-logs and identify valuable information? How do these techniques interact with the user's own knowledge and memory? How can we engage people with their archives? Email is a good platform for studying these questions, as many people already have archives spanning relatively long periods of time.

To study these topics, we have designed Muse (Memories USing Email)[1], a system that analyzes long-term email archives and uses data mining techniques to automatically generate a set of cues intended to help spark users' memories. With the help

[1] Muse is publicly available at http://mobisocial.stanford.edu/muse.

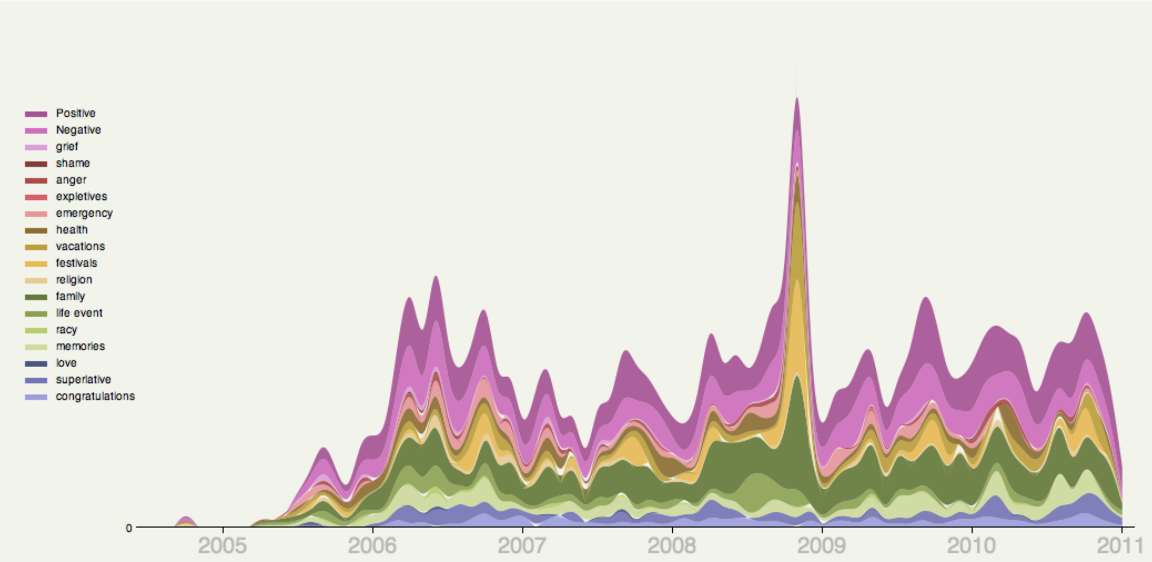


Figure 2.1: A Muse visualization of email sentiment. A stacked graph shows the number of email messages reflecting a particular sentiment category over time.

of these cues, users can identify important or interesting events and begin to explore their email archives, aided by an interface designed to enable quick skimming and navigation between messages. As an illustration, Figure 2.1 shows how Muse presents a timeline summarizing likely sentiments expressed in the archive. Muse also engages users by generating small games intended as memory exercises. To date, Muse has been used to explore email archives containing up to 100,000 messages.

Traditional email clients allow users to examine one message at a time, and to filter and to issue search queries. These clients are ill-suited for browsing a large-scale archive, where a user may not know exactly what to look for. Like others before us [145], we see a two part solution to this challenge. First, we can automatically generate cues likely to orient the user towards messages of interest. Second, once the user has acquired a cue, we can encourage exploration of the messages related to that cue and to other related cues. Such exploration mirrors the natural organization of episodic memory of autobiographical events [136].

The key innovations in Muse for the purpose of reviving memories are the following:

1. Based on iterative design and user feedback, we identify four novel types of cues useful for reminiscence: communication activity with inferred social groups, a summary of recurring named entities over time, occurrence of sentimental terms, and image attachments.

2. We propose specific and relatively lightweight mining techniques to identify, organize and present these cues, as well as techniques to enable rapid exploratory browsing of associated messages.

3. We discuss some of the memories evoked by Muse in users reminiscing with their own email archives, and present insights about the efficacy of different cues. Our studies uncover a range of benefits for exploratory browsing with Muse, from summarizing work progress to renewing old friendships to identifying milestones.

2.1 Related Work

While there has been much prior work in various forms of email analysis, there are relatively few usable systems that let end users explore large scale email archives. There are many email clients and plugins that aim to help users become more effective in tackling email correspondence. Systems like Xobni (xobni.com), Rapportive (rapportive.com) and Gmail's People Widget provide helpful supplemental information with email messages, and tools like Gmail's Priority Inbox and SNARF [91] prioritize messages that are likely to be more important. However, these systems are not targetted to the specific task of browsing long-term email archives.

2.1.1 Email Analysis

Several interesting visualizations have been designed with the goal of providing insights into patterns of email communication, although they restrict themselves to message meta-data and do not analyze contents [78, 100, 144]. These visualizations try and extract the "rhythms" of email communication; Tyler and Tang studied such rhythms in an interview study, and found that there are many contextual cues associated with email response patterns [138].

Themail by Viégas et al. does analyze message contents and aims to help users reflect on the dyadic relationship with each one of their contacts. Themail visualizes single words that have a high TF-IDF score in emails exchanged with a contact over time. Our starting point for generating one of the cues in Muse was similar to Themail; however, we found that several improvements were necessary to build a usable system. Themail correctly pointed to the need to enable visualization of big-picture themes and trends, alongside detailed exploration, though it appears to focus on the former and has little explicit support for exploratory browsing.

The TIARA system from IBM Research integrates LDA-based topic analysis and interactive visualization [74] (building upon the work of Dredze et al [30]). In the domain of email, it was tested with 10 users answering focused questions by studying a 12 month archive of work email belonging to a colleague. In contrast to Muse, TIARA was not designed or tested for the task of users reviving memories. The focus on reviving memories in Muse leads it to emphasize named entities, sentimental messages and social context (with its automatic grouping of people), along with rapid browsing techniques and data integration features for long-term email archives[2].

2.1.2 Life-logging and Reminiscence

Gordon Bell's MyLifeBits project brought widespread attention to the idea of life-logging [42]. Bell and Gemmell's book, "Total Recall" surveys many of the issues around life-logging technologies and applications [13]. Churchill and Ubois have discussed several technical issues related to the collection and long-term use of personal archives (as opposed to just logs), and in a series of articles based on user surveys, Marshall has captured some of the state of the practice in personal archiving [80].

Several research systems actively target reminiscence using digital materials[3]. Whittaker et al. have proposed a set of general design principles for digital tools that support memory [148]. Briefly, these principles are: promoting selected content

[2] Unfortunately, Themail and TIARA are not publicly available for a direct comparison.

[3] A recent special issue of the Human Computer Interaction devoted itself to this theme [141].

from the archive, supporting physical embodiment of data, working in synergy with unaided memory, and building systems that support reminiscence and remembering. Their conclusions are based on several projects that study how people use digital memorabilia: for example, MemoryLane studied users capturing and organizing their digital mementos [60] and FM Radio studied families capturing sonic memories of their previous holidays [101]. Some of these principles are also reflected by Muse. Horvitz' LifeBrowser attempts to use machine learning to infer events and activities that people may find important and memorable from their archive of events, photos, and e-mails [122]. (To our knowledge, this system is yet to be described technically and tested with users.) Pensieve actively solicits input from the user by periodically emailing personal questions and attempts to create a repository of reminiscences [99]. The YouPivot system facilitates searching for contextually associated activities on a desktop computer [50].

Crete-Nishihata et al. have also studied the impact of creating multimedia biographies from personal digital pictures, documents, music, etc. on patients with Alzheimer's disease or mild cognitive impairment [81]. They find that these biographies can have a profound effect not just on the patients, but also on their caregivers. Tools like Muse can potentially be used to create some types of biographies semi-automatically.

Many life-logging systems today are geared towards active life-logging, which involves deliberate actions by the user and/or the use of specific hardware, software or services. Rather than enumerate them here, we refer the reader to personalinformatics.org for an excellent list.

Mining spending data [116] is an example of analyzing a passively captured life-log. Muse shares this focus on passive life-logging, but works in the domain of email communication.

2.1.3 Long-term Personal Archives

There is also recent research on assessing attitudes towards long-term personal archives. Kaye et al. interviewed several academics and found that a personal archive is used not just for storing and accessing information, but also for building a legacy, sharing information, preserving important objects, and constructing identity [62]. Kim's surveys show that people are evolving various strategies for managing their digital legacies, ranging from deleting everything to sorting data and sharing it with specific people and entities [63]. Odom et al. have studied the design of technology heirlooms for passing digital objects between generations in a family [95]; another paper captures the ambivalent feeling of possession people have when their materials are in off-site cloud storage [96].

Lindley et al. state in the context of personal archives that *"The past is not simply something one has but is something one uses."* and study with interviews how personal memories translate into family history [71].

Other work takes an ethnographic approach to understanding family memories and archiving practices in the physical world (e.g., [64]). Petrelli and Whittaker contrast digital and physical mementos used in family memories; their fieldwork corroborates our thesis that email is frequently one of the digital sources of memories [102]. Elsweiler et al. conducted a field study of how people use refinding in the context of email and conclude that *". . . although people generally remember quite a lot about their emails, there are situations in which people remember less and in these situations it may be more difficult to refind the information required with existing tools*" [34].

2.1.4 Legal Discovery and Intelligence Analysis

There are some similarities between Muse and systems used for legal discovery and intelligence analysis. Users of these systems also need to spot cues and peruse large-scale, loosely organized data sets, often including emails. Muse shares techniques with such tools; for example, extraction of key entities as in Jigsaw [126], and the need for pre-built views of intelligence corpora [23]. However, the use case for discovery tools (trained analysts making sense of unfamiliar corpora in order to spot suspicious activity) is starkly different from that of Muse (mainstream consumers using their own email for reminiscence). This leads to different design considerations for Muse.

2.1.5 Interaction Techniques

Many systems use faceted navigation to support exploratory browsing (e.g., Flamenco [151], Phlat [27] and Stuff I've Seen [31]). Muse employs similar ideas in the domain of email, where the facets are people, months, years and automatically inferred groups and sentiments.

Systems like Cyclostar have demonstrated the effectiveness and versatility of elliptical gestures [77]; the Muse jog dial we describe in later sections is a simplified form of such an interface.

2.1.6 Text Analysis and Visualization

The Parallel Tag Clouds visualization [25] highlights the presence as well as absence of significant terms across the parallel text corpora of different circuit courts. As described later in this chapter, Muse has somewhat similar requirements for identifying key term in email messages across time, though it detects only the presence of terms, not their absence. Themeriver is a system for visualization of text content acquired over time, such as news articles [54]. It uses a visualization similar to the stacked graphs in Muse, but applied to "themes" instead of sentiments or groups. Work in the topic detection and tracking (TDT) area tends to focus on clustering and labeling the textual content of messages (e.g., [129]). Unlike Muse, all of the above systems do not target the task of reminiscence, and do not take advantage of personal and social context. However, there are possible synergies between the text analysis and visualization techniques used by Muse and these systems.

2.1.7 Sentiment Analysis

There is much work on sentiment analysis of tweets and reviews for the purposes of inferring public reaction to a product (e.g., [8]). We Feel Fine is a visualization of sentiments expressed in public blog posts and has been used to generate hypotheses about sentiments at a societal level [61]. Muse is the first system to apply sentiment analysis to email archives and can potentially be used by an individual to examine hypotheses about herself.

2.1.8 Memory Research

Ebbinghaus pioneered the experimental study of memory back in 1885 [33]. Foer has surveyed the role of memory through the ages in an engaging account about participating in memory competitions [40]. Tulving and others have studied the nature of the episodic memory of autobiographical events [136], and find that episodic memory tends to be centered around people, places and events. Muse uses this observation in its extensive use of names to evoke memory, as described later in this chapter. The effect of using "external storage" to aid memory is also well-studied in the transactive memory framework [146]. For example, one can avoid having to remember things that someone close by knows well. Personal archives are an extremely scalable form of such storage. Sparrow et al. examine the impact of having information available at our fingertips through computers and the Internet [124]. There is also research showing that emotional episodes tend to be well remembered [112], a fact exploited by Muse using its sentiment analysis.

van den Hoven and Eggen survey what is known about autobiographical memory and provide design recommendations for augmented memory systems. They state that "*Reminiscing is a recurring process, continuously shaping peoples personal histories and identities.*" [142] Some of their recommendations, e.g., that an augmented memory system must support memory cuing, are applicable to Muse.

2.2 Using Muse

To run Muse, a user typically downloads it to her own computer and launches it using Java Webstart. This reassures users about confidentiality, which is important given that email archives frequently contain highly sensitive information, from love letters to financial documents. Muse starts up an embedded web server in the background, and launches a browser window for the user to interact with it. This interface choice means users can use their favorite browser and its familiar features like multiple tabs and windows, navigation buttons, bookmarks and browser plugins.

2.2.1 Accessing and Selecting Email

The user specifies one or more sources of email to Muse, including online POP/IMAP servers or mbox format files stored on a local file system. Getting access to old email archives is not always easy; a typical comment we get from users is: *"I know my old email is lying zipped up on a CD somewhere in the basement, and I think I could get to it if it was really important."* However, most people we interviewed over the course of this work did want the assurance of knowing that they could get to it "somehow". Once access is available to the actual files, it is relatively easy to convert between email formats.

After Muse has access to the email archives, the user can select folders to analyze from each source and optionally tell Muse to only analyze their sent messages. Focusing on sent messages is a useful heuristic (lacking specific direction from the user) for reminiscence because they are handcrafted by the user and reflect the user's thoughts and actions. In contrast, the INBOX tends to be much larger (on average, 2.5 times as many messages involving 4.8 times as many people, according to an anonymized data set provided to us for research purposes by Xobni). Much incoming email is delivered via mailing lists, which are sometimes scanned casually or even ignored entirely. Since access to online email providers can be slow, Muse caches messages once fetched, though the cache can always be cleared under user control.

2.2.2 Data Cleaning and Entity Resolution

Once Muse has fetched the user's messages, it processes and indexes their contents and attachments, and builds an address book of contacts. Typical processing speed on a current-generation laptop or desktop computer is about 1,000 messages per minute. Frequently, data is poorly formatted particularly in old archives, so Muse tries to be tolerant of errors, for example by guessing missing data such as timestamps, or ignoring parts that cannot be parsed. The user has the option of reviewing these corrections in a data quality report. In extreme cases, Muse has been used with email messages that were scanned from printed email messages and put through optical character recognition, which leads to noisy input data.

MUSE performs entity resolution by unifying names and email addresses in email headers when either the name or email address (as specified in the RFC-822 email header) is equivalent. This is essential since email addresses and even name spellings for a person are likely to change in a long-term archive. Name equivalence is tested by ignoring case differences and equating commonly used variations in naming, e.g., with or without a middle initial, “Firstname Lastname”, “Lastname, Firstname” and “Firstname Lastname - Department”.

2.2.3 Archive Filters

Once the address book is available, MUSE infers social groups based on a grouping algorithm, described in the next chapter. The user can specify the number of groups to be inferred, but 20 is a reasonable default. Users can also refine the inferred groups manually.

MUSE then allows the user to browse four different kinds of cues and the messages associated with them. These cues are described in detail in the next two sections. Cue generation normally applies to all the messages that have been indexed by MUSE. However, the user can also easily apply filters to the complete set of messages indexed by MUSE and regenerate the cues to restrict her attention to a particular portion of the archive. Users can apply filters by search term, by date range, by associated groups or sentiments (which are automatically derived, as explained below), by person, by the name of the original folder that contained the message, or by message direction (sent or received).

Since users’ tend to become highly engaged while exploring their archives containing years or decades of messages, it is not uncommon for them to spend several hours with MUSE. To allow them to split this time across multiple sessions, MUSE lets users save session state and reload it later without having to reprocess the archive.

2.3 Memory Cues for Browsing Email Archives

In this section, we discuss a set of memory cues for browsing email archives and techniques for generating them. We discovered these cues by observing users interact with early versions of Muse on their archives and analyzing why they appeared to be useful. For each type of cue, we present the intuition, some details of its implementation, and the presentation technique used.

Automatically generated cues do not have to be fully precise or complete; rather, they should work hand-in-hand with a user's memory and the actual content of the messages. In practice, we expect that many cues will be ignored by a user because they are obvious, redundant, noisy or overshadowed by other, stronger cues. However, for a system to be engaging, it should surface a relatively high fraction of useful cues that lead to valuable memories and avoid flooding the user with misleading or irrelevant cues. We also prefer techniques that are sufficiently lightweight to run on end-users' own machines and do not require server-class hardware.

2.3.1 Group Cues

People routinely interact with thousands of contacts in the course of a few years over email. Since it is difficult to visualize messages, topics, and communication activity with so many individuals, Muse groups these contacts automatically as one way to organize the archive. Users can refine the groups if necessary, and then visualize and explore their communication with the group as a whole. This approach mirrors the way people mentally chunk their contacts into groups like family, colleagues, classmates, neighbors, and so on.

Muse automatically discovers likely groups by analyzing co-recipiency in messages. It employs a novel group mining algorithm [107] (described in the next chapter), that satisfies several properties that are important in social contexts. First, a group may consist of people who do not all appear together in any single message (e.g., a user's extended family.) Second, within a group, there may be important subgroups with a significant identity of their own (e.g., siblings as an important subset of extended family). Next, the same person could belong to multiple groups (e.g.,

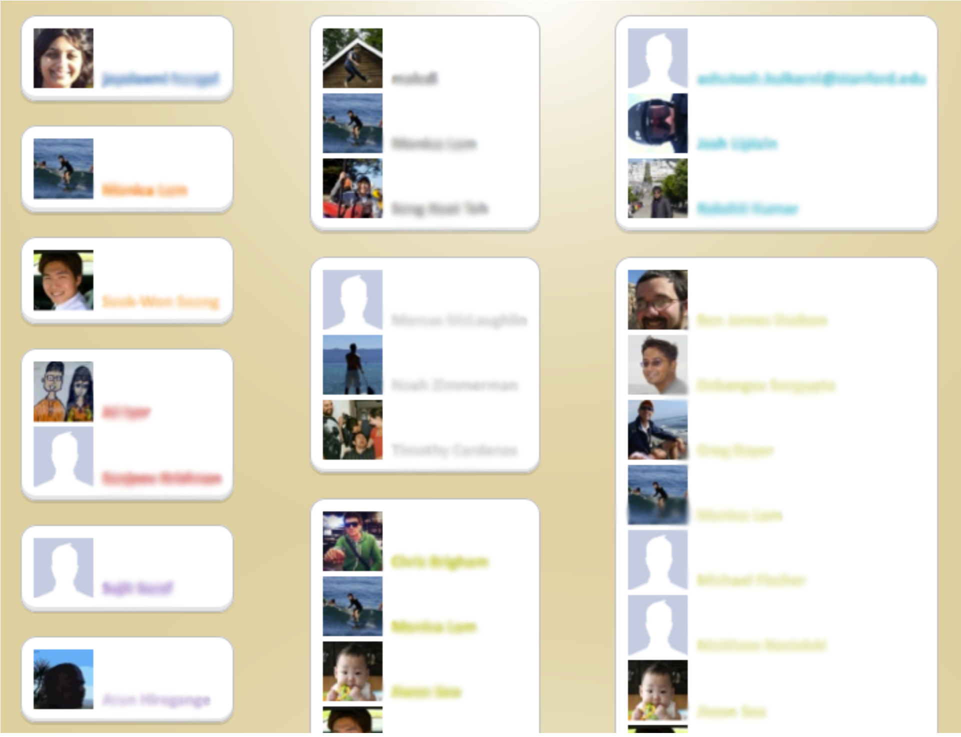


Figure 2.2: The groups editor showing automatically inferred groups that can be refined by the user. Names are blurred to preserve user privacy.

a colleague at work may also be part of a hiking group.) And finally, a significant “group”, for the purposes of organization, could consist of just one very important person with a high communication volume, such as a spouse or close friend.

Users can optionally refine the inferred groups manually using a drag-and-drop editor. The editor lets users move people between groups and create, clone or delete groups. Users can see the name of each person in a group, but if their browser supports the W3C contacts API [149], they also see the contacts’ pictures, if available. For example, Mozilla’s Contacts plugins for Firefox can fetch photographs for friends from networks like Facebook, LinkedIn and Gmail. Fig. 2.2 shows a screenshot of the groups editor with this plugin.

To present the cues associated with groups, Muse generates a stacked graph visualization with one layer per group as shown in Fig. 2.3. This visualization lets users spot relative patterns of communication with each group over time, and to correlate them with total communication volume. Most users find that this visualization tells

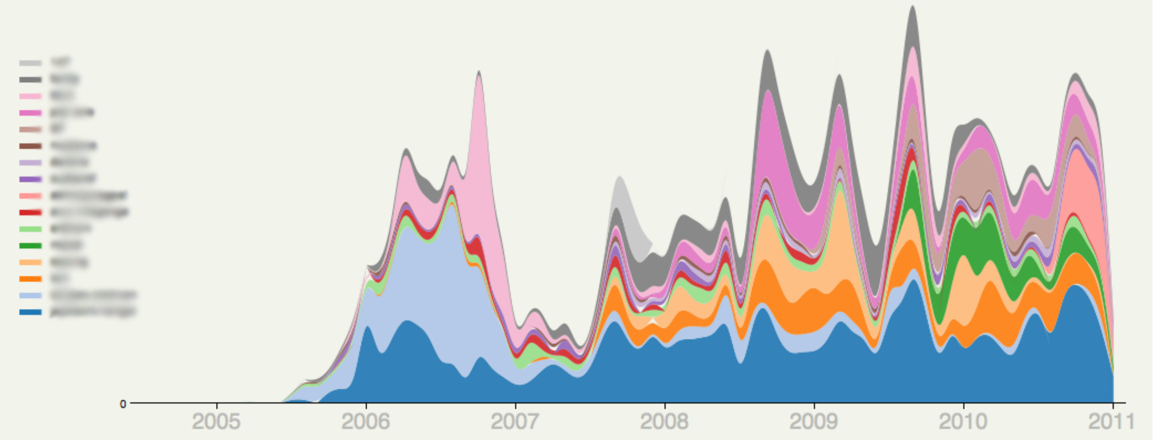


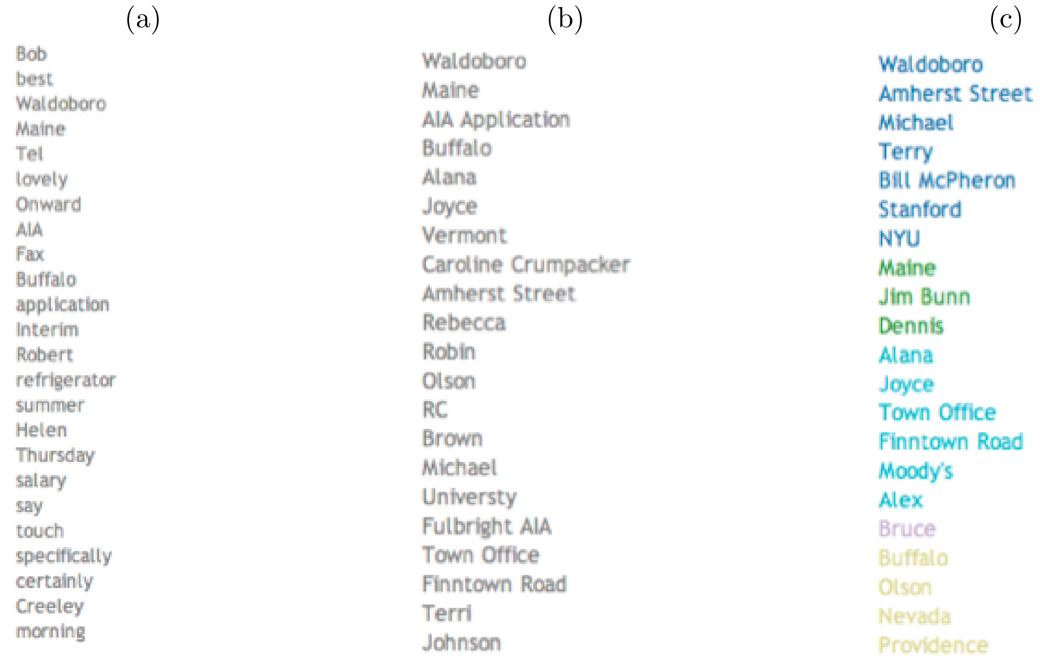
Figure 2.3: A stacked graph representation of communication with each group over time. Group names are blurred to preserve user privacy.

a story of when they started and stopped interacting with various groups, reflecting different phases of their lives. Users can click at any point on the stacked graph to launch into a view containing all the messages exchanged with that group. The view is initialized to the point in time along the X-axis that was clicked. Each group is also assigned a color, which is used to code important terms, as described below.

Optionally, users can view and explore separate, small multiples graphs of communication with individual groups. Another option for users is to apply the same visualizations to their top individuals instead of top groups.

2.3.2 Name Cues

To provide a quick overview of an archive, Muse creates a summary of important terms on a monthly basis. Terms that make the best cues are often names of various kinds, including people, places, organizations and so on, because names generally tend to carry rich associations in the user's episodic memory [136]. Hence Muse first extracts named entities from message contents and focuses its analysis on these terms. We apply the Named Entity Recognition package from the Stanford NLP toolkit [39], using the default training model. This decision was informed by early experiments using the standard TF-IDF (Term Frequency by Inverse Document Frequency) metric with single words, similar to Themail [145]. The results with this metric were very



(a)

Bob
best
Waldoboro
Maine
Tel
lovely
Onward
AIA
Fax
Buffalo
application
Interim
Robert
refrigerator
summer
Helen
Thursday
salary
say
touch
specifically
certainly
Creeley
morning

(b)

Waldoboro
Maine
AIA Application
Buffalo
Alana
Joyce
Vermont
Caroline Crumpacker
Amherst Street
Rebecca
Robin
Olson
RC
Brown
Michael
Universty
Fulbright AIA
Town Office
Finntown Road
Terri
Johnson

(c)

Waldoboro
Amherst Street
Michael
Terry
Bill McPheron
Stanford
NYU
Maine
Jim Bunn
Dennis
Alana
Joyce
Town Office
Finntown Road
Moody's
Alex
Bruce
Buffalo
Olson
Nevada
Providence

Figure 2.4: Comparison of top-ranked terms for the same month on a portion of the Robert Creeley archive. (a) single words, (b) named entities, and (c) named entities color coded and clustered by group.

noisy, despite using appropriate stop-word lists and other heuristics like factoring word commonality into ranking.

Fig. 2.4 illustrates the difference on an example corpus: a portion of the email archive of noted American poet Robert Creeley, which is hosted at Stanford University Libraries [29]. Using word-based TF-IDF, the top-scoring terms, shown in Fig. 2.4(a), include generic words like *best*, *lovely*, *say* and *touch* which are unlikely to be useful, and in fact tend to waste the user's time as she tries to understand the context in which they arise. (The Themail program is not available to us, but from the screenshot in its paper, it appears to suffer from the same problem.) However, we observe from this list that the terms most likely to be interesting are the named entities such as *Waldoboro*, *Helen*, etc. Fig. 2.4(b) shows the results of first extracting named entities from the message contents, and scoring only these terms. The overall results already appear to be better cues to memory. The observation that names of people and places can be especially evocative has been made by others as well [148].

We also experimented with 2 other options for identifying key terms: scoring multi-word phrases, and scoring noun phrases extracted with linguistic parsing. In an informal study with 5 users, we found that all users preferred named entities since names tend to have deeper associations in the user's memory compared to phrases, which may still lack context.

Time-based TF-IDF

To score terms, we use a metric based on TF-IDF scoring of the named entities, using the *ntn* variation described by Manning et al. [79]. This variation corresponds to natural term frequency, the regular definition of inverse document frequency and no TF normalization, which is the simplest form of TF-IDF computation. High TF-IDF scores are associated with terms that are specific to a particular document (in our case, the messages for a month) compared to the rest of the corpus.

To further improve term scoring, we introduce a simple, novel variant for *time-based* TF-IDF. The traditional IDF factor penalizes the score of terms that appear across many documents in the entire corpus. In contrast, we compute the IDF factor for a term in a document based on the number of documents *preceding* that document that contain the same term. More formally, in a time-based document corpus, where each document D has a timestamp $time(D)$, we compute the score for a term T in document D as:

$$\begin{aligned} SCORE(T,D) &= tf(T,D)\ \times\ t\_idf(T,D) \\ t\_idf(T,D) &= \log \frac{|d|,\ time(d)\ \leq\ time(D)}{|d|,\ T \in d,\ time(d)\ \leq\ time(D)} \end{aligned}$$

where $tf_{T,D}$ is the frequency of a term T in document D, and $t\_idf_{T,D}$ is the special, time-based version of inverse document frequency that is computed with respect to the documents in the corpus that precede or equal D in time. (The traditional definition of IDF assigns a fixed value to a given term based on its frequency across the entire corpus and does not vary it across documents.) For our purpose of generating summaries in MUSE, we use coarse time steps at a granularity of a month.

Intuitively, it makes sense to identify the most significant terms at a particular point in time without knowledge of the future. We have observed that people tend to find the onset of a new term particularly memorable; such scenarios are promoted by the time-based TF-IDF. Examples of terms that benefit from this metric include the name of a newborn family member, or a name like *Obama*, that emerges at some point in time and subsequently becomes commonplace. As time goes by with repeated use of the new term, its IDF score slowly reduces, making the term less prominent.

A concrete example from the author's email archives illustrates this point. Muse highlighted terms related to music in the first two months after he started taking music lessons. Thereafter, the music conversations continued, but these terms disappeared from the monthly summaries because they become relatively common. With the regular version of TF-IDF, these terms do not show up at all because of a relatively low IDF score across the entire archive.

To present these terms to the user, Muse lists the top named entities for each month in a month-by-month view, similar to a calendar. Users can ask for more or less terms to be displayed (the default is set to 30 terms for each month). Muse clusters and color-codes the terms by the group they are most closely associated with. Terms assigned to the same group (and therefore the same color), are displayed together, and are further sorted by descending score. This is useful because terms belonging to a group tend to be related to each other and the user can quickly scan them together; users frequently have varying levels of interest in different groups. An early round of informal testing with 5 users showed that organizing and color coding terms by group was unanimously preferred over just sorting terms by score. Terms not assigned to any group are colored gray and displayed last. The color encoding also makes it easy to scan all terms related to a particular group across different months. Fig. 2.4(c) shows the names view actually displayed by Muse for this example.

To avoid the problem of over-representation from a single message, we throttle the number of terms displayed that belong to a single message. After a preset threshold is reached (empirically, four works well), other terms from the message are suppressed, unless they are also present in a different, non-maximally represented message.

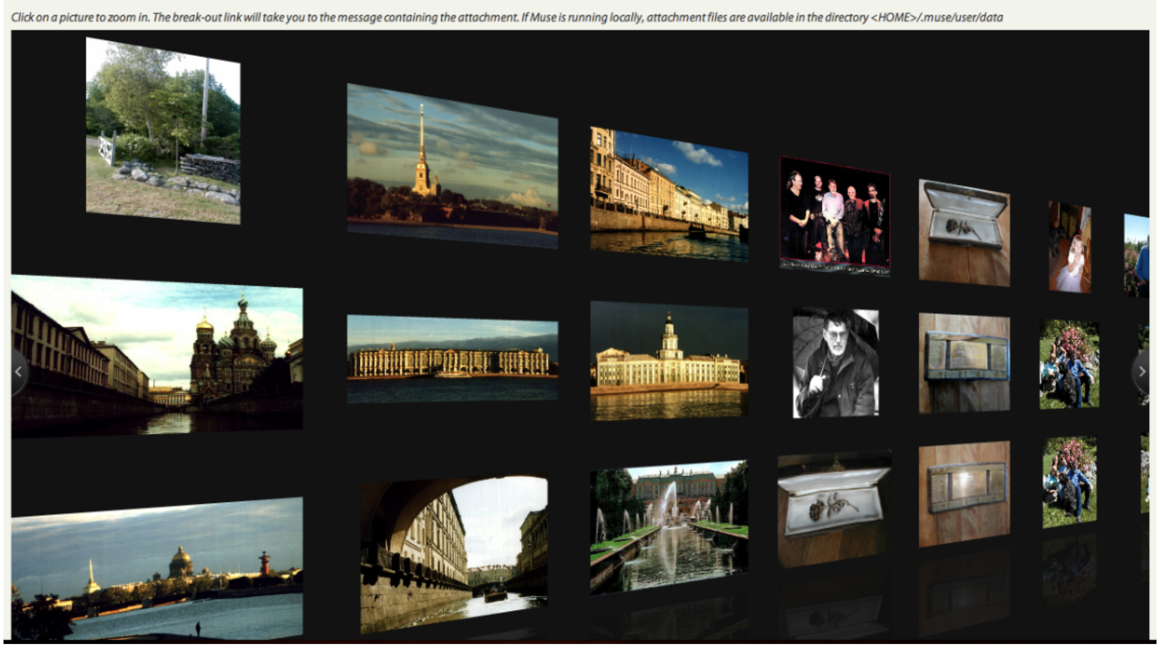
2.3.3 Sentiment Cues

During our experiments with early versions of Muse, we found that many of the messages that engaged users the most during the reminiscence process were those that reflected significant turning points in their lives and deep emotions, such as love, joy, grief and anger. There is much evidence that emotional episodes tend to be well-remembered, both for positive and negative emotions [112]. Muse therefore uses simple sentiment analysis techniques to let users quickly browse messages by the sentiment associated with them.

The most commonly used tools for sentiment analysis such as SentiWordnet [9] and LIWC [75] use word lists to detect sentiment. However, these lists are mainly meant for linguistic analysis of text corpora and contain many categories that are not relevant for personal messages. We have therefore generated our own lexicon (currently, mainly for the English language) that consists of about 20 categories with terms covering various emotions, family, health, life events, expletives, etc. that may be useful for our domain of reminiscence with email archives. These terms are matched (modulo stemming) with the contents of the message. Significant emotions that can be detected with high certainty, such as congratulations, are assigned their own category. Other kinds of emotions that can be classified with less certainty are grouped into two broad categories for positive (gratitude, pride, joy, humor, etc.) and negative (disappointment, anxiety, worry, etc.) sentiment.

Muse depicts the frequency of messages reflecting these sentiment categories across time using a stacked graph (see Fig. 2.1); each layer represents a particular sentiment category. Users can click on a layer to launch into a view containing all the messages reflecting that sentiment; the view is initialized to the point in time along the X-axis that was clicked. As with groups, users can view independent, small multiples graphs for each sentiment category.

We have found that while sentiments are among the noisiest cues provided by Muse, they are also often the most engaging. Users are curious about interpreting the sentiments graph, especially when exploring their sent messages. Even if a detected sentiment is due to a language artifact, it often gives users a new view into their own use of language. As we will describe later in this chapter, users can edit this lexicon



Click on a picture to zoom in. The break-out link will take you to the message containing the attachment. If Muse is running locally, attachment files are available in the directory <HOME>/.muse/user/data

Figure 2.5: Image attachments shown on the PicLens photo wall.

to suit their language and interests.

2.3.4 Picture Cues

Picture attachments in email messages are useful because the vividness of images provides strong cues to memory. Further, pictures are often taken for the explicit purpose of later remembrance and hence may be particularly worth recalling.

Muse extracts picture attachments from messages (and optionally PDF documents, which are converted to thumbnails) and displays them on the PicLens photo wall (from cooliris.com) which provides a 2.5D zoomable and draggable interface [103]. The images are arranged in reverse chronological order and it is easy for the user to rapidly scan pictures, and pan to different areas of the wall without waiting for the whole wall to render chronologically.

We find that users are often pleasantly surprised to rediscover pictures from their email attachments. While many more pictures are shared through formal mechanisms such as photo sharing sites, the memory of pictures sent in email attachments is not refreshed since it is not easy to browse email attachments. Therefore there is a sense of novelty in re-discovering a long forgotten picture, as several of our users found.

From the photo wall, users can click on an attachment to go to a view containing the message(s) with that attachment. From that point, the user can continue exploratory browsing using the usual facets such as people, groups, and sentiments. The browsing interface is described in the next section.

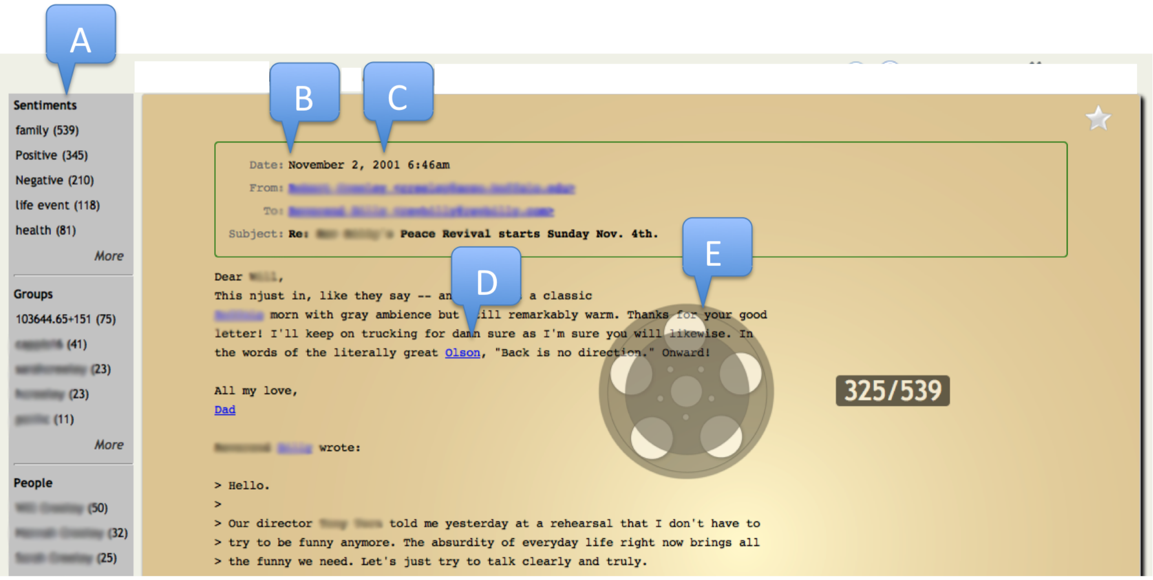
2.4 Exploratory Browsing

Generating interesting cues is often merely a step on the way to browsing the actual messages which hold the memorable details. In practice, most users spend more time browsing messages than browsing cues. Therefore it is important to support rapid exploratory browsing of a large set of messages. When a user follows up on a cue, e.g., by clicking on a name in the monthly summaries, or by selecting a point in a stacked graph visualization, Muse opens a *message view* for the associated messages. The message view displays the actual message header and contents, along with thumbnails of any attachments for the message at the bottom. Multiple views can be simultaneously active in different browser windows or tabs to encourage multiple chains of exploration.

2.4.1 Skimming with an On-screen Jog Dial

Our original implementation of the message view displayed all the messages in the view, one below the other. This tended to create long pages, and we found that when there were more than about 10-15 messages, users would get bored and stop scanning messages part of the way down the page. While following cues, however, some views (for example, all messages exchanges with a group or person) can consist of hundreds of messages and are tedious to click or scroll through.

To alleviate the tedium of scrolling down a long page, we load multiple messages into the browser but display only one message at a time in a fixed message frame. This has the advantage that it fixes the on-screen locations of the message headers



Sentiments
family (539)
Positive (345)
Negative (210)
life event (118)
health (81)
More

Groups
103644.65+151 (75)
[redacted] (41)
[redacted] (23)
[redacted] (23)
[redacted] (11)
More

People
[redacted] (50)
[redacted] (32)
[redacted] (25)

```
    Date: November 2, 2001 6:46am
    From: [redacted]
      To: [redacted]
 Subject: Re: [redacted] Peace Revival starts Sunday Nov. 4th.

Dear [redacted],
This njust in, like they say -- an         a classic
[redacted] morn with gray ambience but still remarkably warm. Thanks for your good
letter! I'll keep on trucking for damn sure as I'm sure you will likewise. In
the words of the literally great Olson, "Back is no direction." Onward!

All my love,
Dad

[redacted] wrote:

> Hello.
>
> Our director [redacted] told me yesterday at a rehearsal that I don't have to
> try to be funny anymore. The absurdity of everyday life right now brings all
> the funny we need. Let's just try to talk clearly and truly.
```

Figure 2.6: The messages view in Muse, with 539 messages loaded. (A) Facet panel for sentiments, groups and people. (B) Link to all messages for month and (C) for year. (D) Hyperlink inserted into message contents, connecting to other messages with the term. (E) Jog dial for rapid skimming.

and the beginning of the message contents, which are often the most important for sensing the relevance of a message. To enable rapid scanning of messages in the view, we provide a translucent on-screen circular jog dial that is summoned on the spot and dismissed by clicking anywhere in the message frame. The operation of the dial is similar to the physical dial on iPod music players: moving clockwise to the next octant causes the frame to display the next message; moving counter-clockwise displays the previous message. Fig. 2.6 shows the message view with the jog dial visible.[4] Apart from being somewhat playful, the dial allows fast interactive performance through the use of client-side Javascript – in our experience, users can rapidly rotate the dial to approach a skimming speed of 150 messages a minute while still being able to monitor the content passing by.

The dial affords finer-grained control than keyboard navigation, as users can slow down and speed up as they wish, depending on their interest level in the phase of

[4] Some details are blurred to protect privacy.

messages. The jog dial lets them travel relatively long distances (scanning through a few hundred messages is common) without the need for precise cursor positioning, mouse clicks, key presses, or switching gaze from the message view. The circularity of the gesture avoids the need to periodically reposition the cursor, and is a general advantage of elliptical gestures. For fast travel in extremely long views, we also provide the option of tabbing backwards or forwards to move month by month. Of course, the user can always use the keyboard arrow keys to move between messages as well.

It is important to manage browser load for a view with thousands of messages. Muse maintains a sliding window of pages around the page currently being displayed, and keeps only the pages in that window loaded in the browser. As the user moves along, the dial “pages in” new messages to maintain the window around the current page, while retiring pages outside the window. In practice, we find that a default window size of 100 pages (60 pages ahead and 40 pages backward) is adequate for good user experience with no stalls and is easily handled by current browsers.

The jog dial is a popular feature with many users of Muse, and is generally preferred to the keyboard for casual browsing. (In fact, our users have asked us for the ability to use the dial to flip through regular web pages.) We designed the dial to operate primarily with a trackpad, but users have also reported relatively high satisfaction rates with a mouse.

2.4.2 Facets and Hyperlinks

We encourage users to follow top-level cues by using links to related facets and annotating the message contents with the named entities identified as top level name cues. For example, if a name cue is mentioned in a message, we insert a hyperlink for that name to a view containing all the messages with that name. Users can optionally open this new view in a different browser tab without disturbing the current chain of exploration. Such associative linkages were a key feature in the original design of the memex [20]. Similarly, ordered lists of groups, sentiments and people associated with the messages in the current view are shown in a facets panel on the left (see Fig. 2.6).

From any message, the user can click on the month or year in the message header to launch a new view containing all the messages in that time unit. A message view can be generated by querying for a term or sentiment category, in which case that term or words related to the category are highlighted in the text of the message.

Throughout the interface, clicking on the name or email address of a person, or the description of a group, can be used to launch a message view containing all the associated messages. Similarly it is possible to launch an attachment wall consisting of all pictures in the messages in the current view. These features lead to a rich, associative environment for browsing and exploration.

Another valuable source of information embedded in email messages are links to web pages. In a long-term archive, the web pages pointed to by these links are likely to have gone dead. To alleviate this problem, Muse optionally hyperlinks URLs to the version of the page as it may have existed at the time the message was exchanged, according to the Internet Archive's Wayback Machine [7]. In Chapter 6, we will describe novel uses of the links embedded in email archives.

2.5 User Studies

Throughout the development of Muse, we conducted surveys and formative studies to inform our design. We also conducted two formal studies (with a total of 13 users), did other informal testing with early users, and obtained feedback from web users. We now report on a small study conducted in April 2011 with 6 users to test the then current version.

2.5.1 Methodology

In this study, we recruited 6 participants (P1–6) who had access to relatively long-term email archives. We required them to have at least 5,000 messages (preferably sent by the user), acquired over at least 5 years. We invited three professionals from our university library to participate in this study because they could offer us expert-level feedback based on their experience in dealing with archival material. Two were professional archivists who dealt with special collections, both physical and digital, and the third was a historian and professional curator for the library. The remaining three participants were working professionals. Participants had between 10 and 30 years of work experience, and had all been using email throughout their working lives for both professional and personal purposes. Two of the participants were female, and only one had briefly used an early version of Muse before this study. Participants were compensated with a nominal $10 gift coupon.

We conducted a pre-study meeting with each user to ascertain the state of their email archives. Without exception, all users had email archives in multiple accounts or sources (online service providers, company account, files on a hard drive, etc.) Some users had archives in older formats (like Eudora) and no longer had the program to read it; in these cases we helped them convert their archives to the mbox format which Muse can read. There were frequently discrepancies between what users thought they had in which folders, and what material was actually present, pointing to the difficulty of maintaining consistent foldering practices, across time and different accounts, even for professional archivists. Further, some service provides like Hotmail support only the POP email protocol, which allows access only to the Inbox folder. One user had a significant number of messages in a Hotmail account; we helped him import the Hotmail messages to Gmail for better access via IMAP. The problem of maintaining access to email in historical formats is well-known to archivists [45], and they pointed us to commercial software that they frequently use to convert from one email format to another.

In the actual study, we gave participants a 5-minute tour of the different types of cues and browsing features of Muse. We then asked them to spend 30 to 45 minutes examining the cues and following them to revive memories. At the end, we asked them to fill out a detailed survey with 41 questions (5 of the 6 users asked for more time to browse their archives, or if they could return the survey after running Muse on other parts of their archives, in which case we let them do so). The survey asked them to rate the usefulness of different kinds of cues, and to rate different aspects of the user interface such as the jog dial and faceted navigation. Other questions asked for feedback on sentiment categories that were useful or noisy, and for general comments about the automatically inferred social groups.

The supplemental materials accompanying this dissertation include survey responses from this study. The survey also refers to another type of cue that we tested in this study based on identifying place names mentioned in email, which were then visualized on a map. However, we found this cue noisy and unreliable, and therefore removed it from subsequent versions.

2.5.2 Results

Most users thought Muse provided useful cues to jog their memory and remind them of past incidents, which they had otherwise forgotten. While using Muse, people were deeply engrossed in their past. They were reminded of both high-level patterns (P5: "*That year is full of Europe for me, I traveled so much.*") and specific episodes (P1: "*I had to go to the DMV when I moved to a new city.*"). Commonly, users were surprised by the extent of material in their archives (P4: "*Wow – I'm writing a book on Warcraft, and I didn't realize my email had stuff about it back in 1999!*"), and generally enjoyed discovering long-forgotten messages.

Broadly, all four cue types got good ratings from users. On a 5-point scale, the attachments cues were rated the highest with an average of 4.25, closely followed by a tie between monthly terms and sentiments at 4.17 each. The groups cues got an average rating of 3.83. While our sample size in this study is too small to be conclusive, comments from other users of Muse are consistent with these findings.

Two participants remarked that the monthly terms view was what they would use the most. The picture cues were also popular. One user found valuable pictures of her son's first year lying in her archives. P3: "*I've been looking for these and thought they were lost. Let me save them while I can...*"

While we expected that the names and attachments cues would be highly evocative for the user, we were somewhat surprised that users responded well to the simple sentiment cues. The sentiment cues achieved the same average rating as the name cues, which took considerably more time and effort on our part to generate and prioritize. P1 remarked, "*I just keep coming back to the sentiments view, it's so much* *fun.*" Users also tended to take the sentiment graphs fairly seriously. P5: "*I am relieved to see that 'positive' outweighs 'negative' by a considerable margin, especially in recent years!*"

When we asked users which sentiment categories were accurate, and which ones were not, the responses varied. P5: "*I thought positive and negative were pretty accurate, the negative and angry ones did capture some uncomfortable exchanges with a former roommate (there was money involved)*", and P3: "*The congratulatory messages are pretty useful.*" Users also experienced noisy or incorrect sentiments. P1: "*It thinks there is a lot of religion in my life because I used to work in a theology library... well, you know, maybe that's right*" (laughs), or P6: "*I deal a lot with 'Born Digital' documents, so* MUSE *thinks I have a lot of life events.*" For the most part, users ignored misclassifications and focused on the categories that did work well for them. Our observation is that the simplicity of the sentiment categorization has the advantage of being extremely *transparent*, which allows users to easily ignore any parts that do not work well. A surprise to us was that four of our six users voluntarily mentioned that they would like to be able to edit the lexicon and put in their own terms. Perhaps this should not be so surprising seeing the popularity of tools like Google's N-gram viewer that allow people to ask simple questions about the evolution of language. Following this study, we added features to MUSE to let users create new lexicons, customize them and switch between different lexicons.

While the social groups were rated lower than other cues, we noted that the activity graphs with groups straightaway told a story. Most users would look at the graph and say, "*Oh yeah, that makes perfect sense.*" Further, users would often enter group views from the faceted browsing interface (by clicking on a particularly important group or person) and may not have realized that they were following group cues.

Although it was not an explicit goal of this study, we were curious about how users would react to potentially unpleasant memories being brought up. One user volunteered a reaction that was particularly interesting. P1: "*This reminded me of the stress of looking for a job, how much work goes into a cross-country move, and how hard it was to sell my house after I moved. That sounds unpleasant, but being* *reminded of these things wasn't a bad experience – it made me reflect on how much I've been through over the past six years, and how glad I am that certain experiences are behind me.*"

While browsing, some users switched into the mode of looking for what their conversations were about at specific points in time. P5: "*Let's see, does Obama show up in the monthly terms around November 2008? Oh yes, he does. Cool!*"

On the interface questions, the average rating of the jog dial was 3.6 on a scale of 5. This rating is somewhat skewed by one user giving a rating of 1; she found it hard to use on her new desktop with trackpad that she was unfamiliar with. Surprisingly, 2 users with a mouse gave the jog dial a 5 rating. In hindsight, mouse vs. trackpad is a condition that we should have controlled for in our study, but we let users choose whatever computer they had available. The faceted browsing interface was generally liked (average rating 4.00) as was the fact that a regular web browser could be used to interface to Muse (average rating 4.25).

2.6 Uses of Email Archives

We have received a lot of feedback about Muse from users who have used it outside the context of the above study, including people who downloaded the public version of Muse, illustrating the value of making research software publicly available. To date, Muse has received about 6,000 downloads, aided mainly by some press articles about it [118, 6]. In this section, we present insights about the different ways in which users might be able to make use of tools like Muse, based on this feedback and our observations. We enumerate alongside some of the implications for design from these examples.

2.6.1 Summarizing Work Progress

Several of our users remarked after using Muse that they would find it useful to summarize their year when writing an annual report or performance review. A user commented: "*I wish I had this system at project reviews to quickly scan all the project* *group messages since the last meeting.*". These comments illustrate the utility of systems that allow browsing of archives, as opposed to just search.

2.6.2 Extraction and Organization

One user suggested that it would be useful to form a group of all personal contacts and use it to take personal email out with her when leaving a job. She said, "*My husband was leaving the newspaper company he worked at, and spent two days printing out all the personal emails in his work account*", as she rolled her eyes. More frequently, users report having lost personal messages in a work account when they left a job. Hence it is valuable to be able to segment an archive's contents and selectively save parts of it to a new one. In MUSE, this can be achieved by setting filters and exporting messages matching the filter. The filters include the automatically generated groups which can be used to easily identify personal groups and MUSE can then export messages exchanged with just those groups.

There is also a lot of valuable information embedded in email attachments. A user who is a software entrepreneur said that he had saved his email mainly because it had important documents embedded in it, such as "*company ownership spreadsheets, benefits packages, legal agreements – stuff that's nowhere else.*" MUSE lets users access, copy and backup their attachments easily by exporting them to a single folder.

2.6.3 Reminiscence and Recall

While browsing messages, one user noticed that her son's name was part of a message and had a hyperlink. Clicking on the link brought up a view with 224 messages with her son's name. As she skimmed through the messages, she remarked: "*Wow, this offers a pretty complete history of my first son's milestones. There is no other record of this. I've been trying to remember his milestones to compare with my other son's.*"

This example illustrates the value of narratives embedded in email. It is also a use of the automatic hyperlinking MUSE provides which prompts exploration, since this user had not thought of searching for her son's name.

One should caution that it is possible to miss information based on simple text search. For example, there may be misspellings, especially given the sometimes informal nature of email. Another mom said that she would frequently use a single letter abbreviation for her children's names in email, but after she became aware of Muse and the potential value of her email archives, she started spelling their names out completely. (We are planning to add "fuzzy" search features to Muse in future.)

Another use of Muse was for a graduating Ph.D. student to review his emails in grad school in preparation for writing a memoir about his Ph.D. years [49]. Tools like Muse might be particularly useful when people are ending a phase of their lives and would like to look back to reminisce and perhaps generate a memoir.

2.6.4 Resharing Messages

A common theme emerging from several users was that they said they would like to re-share old messages with the people they had sent the message to some years ago. The reasons for this were for fun (*"Oh, it was fun to see all that discussion we had with the family about possible names for the baby; I'd like to remind them of it now!"*) or nostalgia ("*I would like to remind my dad of the advice he gave me when I came to the United States for the first time.*")

This theme suggests a use for aesthetic user interfaces that enable re-sharing of email messages from the past. Apart from nostalgia, it also reflects a desire for people to re-live a shared past as a way of connecting with others. For example, people enjoy poring over old photographs together, and it may be worth designing interfaces to do the same over memorable email messages.

2.6.5 Family Groups

A consistent pattern (also noted by Viégas et al. [145]) was that users tended to spend much of their time browsing messages exchanged with their family group(s), perhaps more than any other group. This may have been due to the long-lived nature of such relationships, which makes introspection on them particularly valuable. In addition, it is common for people to tell distant family members about important milestones and events in their lives, such as job promotions and new romances. One user suggested that we allow editing of results so he could clean them up and share his memories with his family.

2.6.6 Picking up Forgotten Threads

Some users remarked after using Muse that they were reminded of unfinished work or goals they once had. For example: "*I'd like to remind my friend that we were planning this trip – wonder why it got dropped and we never went.*" Our hypothesis is that users may also find such life-browsing useful as a reminder of high level goals and ambitions they once had.

2.6.7 Renewing Confidence

A user reported that she felt a renewed sense of confidence by looking at her past achievements reflected in her archives. This indicates a need for ways to reinforce past accomplishments that are reflected in digital media. A common use of physical mementos such as university degrees hung on a study wall is to enjoy past accomplishments and provide inspiration for future ones [64]. Perhaps future designers will devise methods of subtly weaving in reminders of such accomplishments.

2.6.8 Renewing Relationships

Multiple users remarked after reviewing old conversations that they felt bad they were no longer in touch with people who had been very close some years ago: "*I had forgotten that we were such close friends, but then I moved, and we stopped talking*" and "*Wow! I had forgotten how nice one friend was in offering me a temporary place to stay (I ended up staying elsewhere) but she has been a little grumpy lately but I can forgive that a bit now that I remember that incident.*"

All users said that Muse revived their memories of topics that they had otherwise forgotten about. Sometimes these were topics in themselves, and sometimes they were satellite topics around other events that they did remember: "*I had forgotten about that lunch we organized right before my thesis defense.*" This suggests that Muse can add color and detail even to such “flashbulb” memories of significant and well-remembered events. Further, the archives add concrete evidence to the event; it is well known that even flashbulb memories are prone to incorrect recall with high confidence [112].

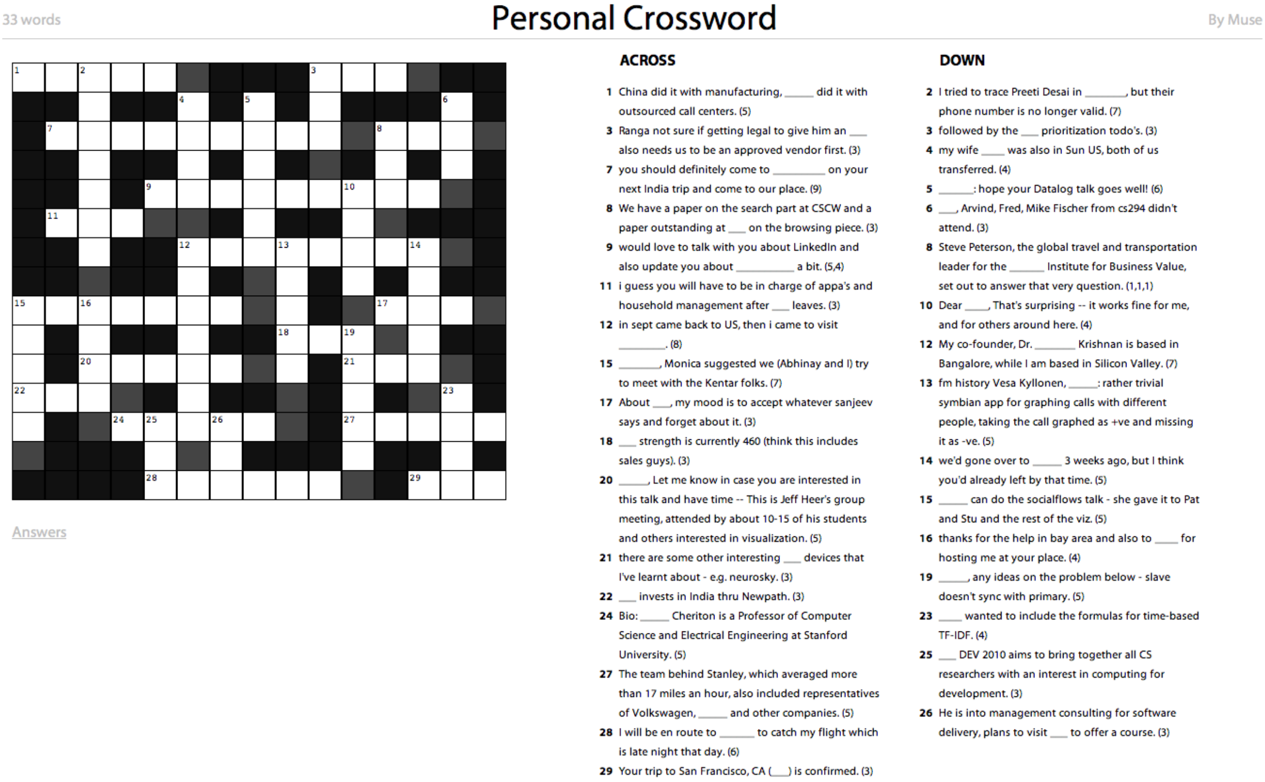
2.6.9 Memory Games

The following excerpt from a message that we received from a web user alerted us to possible applications of Muse for training memory:

“Three years later, I’m still slowly recovering from a disfiguring and near-fatal brain injury (R.frontal and temporal lobes, mostly) which has severely challenged me in many ways, not the least of which is (I hope) encapsulated in the following comment: If I ignore the gaping hole that used to be my life, I’m doing rather well. The doctors tell me if my brain was a house, the room my “self” used to live in is gone. You might imagine it has been very challenging to try to rebuild some semblance of my life, post trauma. I’ve been using Muse to try and figure out who I was and who was who to me in my past, and I want you to know it is surprisingly useful to me.”

This feedback led us to consider whether tools like Muse could be used to create small memory games, whether for memory training or simply for fun. We picked a crossword puzzle format since it is familiar to many people, and has a useful built-in mechanism for providing hints. To generate the puzzle, Muse ranks key names occurring in the personal archive and places as may of them as possible on a crossword grid (Fig. 2.7). The clues for the names are simply sentences in the archive that contain the answer term, with the answer blanked out. We try and generate interesting sentences by scoring sentences containing the term with respect to our sentiment analysis – sentences reflecting strong (positive) sentiments and emotions are more likely to be memorable and fun for the user. If the user is stuck, he can ask for a hint for a particular word, which is simply the header of the email message (the participants in the message, the date and the subject). As usual, the user can apply filters selecting a particular set of messages to draw the puzzle from.

When the puzzle is solved, users are invited to refresh their memories about the



33 words

Personal Crossword

By Muse

ACROSS

1 China did it with manufacturing, _____ did it with outsourced call centers. (5)

3 Ranga not sure if getting legal to give him an ___ also needs us to be an approved vendor first. (3)

7 you should definitely come to _________ on your next India trip and come to our place. (9)

8 We have a paper on the search part at CSCW and a paper outstanding at ___ on the browsing piece. (3)

9 would love to talk with you about LinkedIn and also update you about __________ a bit. (5,4)

11 i guess you will have to be in charge of appa's and household management after ___ leaves. (3)

12 in sept came back to US, then i came to visit ________. (8)

15 _______, Monica suggested we (Abhinay and I) try to meet with the Kentar folks. (7)

17 About ___, my mood is to accept whatever sanjeev says and forget about it. (3)

18 ___ strength is currently 460 (think this includes sales guys). (3)

20 _____, Let me know in case you are interested in this talk and have time -- This is Jeff Heer's group meeting, attended by about 10-15 of his students and others interested in visualization. (5)

21 there are some other interesting ___ devices that I've learnt about - e.g. neurosky. (3)

22 ___ invests in India thru Newpath. (3)

24 Bio: _____ Cheriton is a Professor of Computer Science and Electrical Engineering at Stanford University. (5)

27 The team behind Stanley, which averaged more than 17 miles an hour, also included representatives of Volkswagen, _____ and other companies. (5)

28 I will be en route to ______ to catch my flight which is late night that day. (6)

29 Your trip to San Francisco, CA (___) is confirmed. (3)

DOWN

2 I tried to trace Preeti Desai in _______, but their phone number is no longer valid. (7)

3 followed by the ___ prioritization todo's. (3)

4 my wife ____ was also in Sun US, both of us transferred. (4)

5 ______: hope your Datalog talk goes well! (6)

6 ___, Arvind, Fred, Mike Fischer from cs294 didn't attend. (3)

8 Steve Peterson, the global travel and transportation leader for the ______ Institute for Business Value, set out to answer that very question. (1,1,1)

10 Dear ____, That's surprising -- it works fine for me, and for others around here. (4)

12 My co-founder, Dr. _______ Krishnan is based in Bangalore, while I am based in Silicon Valley. (7)

13 fm history Vesa Kyllonen, _____: rather trivial symbian app for graphing calls with different people, taking the call graphed as +ve and missing it as -ve. (5)

14 we'd gone over to _____ 3 weeks ago, but I think you'd already left by that time. (5)

15 _____ can do the socialflows talk - she gave it to Pat and Stu and the rest of the viz. (5)

16 thanks for the help in bay area and also to ____ for hosting me at your place. (4)

19 _____, any ideas on the problem below - slave doesn't sync with primary. (5)

23 ____ wanted to include the formulas for time-based TF-IDF. (4)

25 ___ DEV 2010 aims to bring together all CS researchers with an interest in computing for development. (3)

26 He is into management consulting for software delivery, plans to visit ___ to offer a course. (3)

Answers

Figure 2.7: A crossword puzzle generated automatically from the user's email archive.

answer terms by browsing the original messages containing them. While our current implementation is an initial prototype, early reactions from users who have tried it are encouraging – they find that it is engaging and evokes forgotten memories.

2.6.10 Browsing Other Email Archives

The expert users from our library were very interested in using Muse to enable browsing of archives of famous people whose papers they help to acquire and organize. P6: "*We have so many donors wanting to donate their documents (including email) to us, we don't have enough people to look through all the materials.*" Journalists have also told us that they would like to use Muse to process email archives, since email is frequently a source of record for news stories.

In the future, it is likely to become common for people to inherit and preserve digital archives from deceased family members [21]. Digital archives reflecting family history are likely to be extremely valuable, just like physical ones. Reflecting this fact is the emergence of long-term archival services like Amazon Glacier [11], though there is considerable debate about the true long-term costs of storage [114]. Similarly, organizations may very well preserve the email archives of key individuals (for example, the founder of an institution or university department). There may also be cases where individuals simply want to publish historical emails freely as a matter of record. Providing users tools to easily do all of the above may spur these phenomena.

There are a host of socio-technical issues yet to be explored in this area, including the privacy of parties involved in the communication. Given its pervasiveness and volume, the use of emails raises issues that go beyond mores that have been adopted for the use of letters.

As a result of these observations, we have extended Muse to include features useful for researchers who are interested in browsing the email archives of donors. These enhancements are described in Chapter 4.

2.6.11 Summary

What struck us, as we saw our users' reaction to Muse, was the variety of ways in which users derived utility and benefits from a system that jogs their memory. Though our initial goal with Muse was only to support the task of reminiscence, the stories above include an example of each one of the "5R's" described by Sellen and Whittaker [119]: recollection, reminiscing, retrieving, reflecting, and remembering intentions. Further, they illustrate that browsing and remembering the past is not just idle "navel-gazing", but can be deeply meaningful and can affect actions in the future.

2.7 Conclusions

We have designed and evaluated four novel types of cues that, along with a system for rapid browsing, ease the task of sense-making from large email archives. Users are surprised by the extent of information directly or indirectly reflected in their archives, and broadly enjoy discovering forgotten topics. From utilitarian purposes like summarizing work progress, exporting personal messages from a work account, and saving attachments, to reminiscence and remembering with nostalgia, watching a child grow up, or re-sharing classic messages, to reinforcing confidence, renewing relationships and training memory, there are a wide range of purposes for which email archives can potentially be effective. Further, we discovered that tools like Muse can be useful to archivists and other curators of digital content, a topic discussed further in Chapter 4.

In the next chapter, we will describe the grouping algorithm used in Muse. Readers not interested in algorithmic details can go on to Chapter 4 without loss of continuity.

Chapter 3

Grouping Algorithm

In this chapter, we tackle the problem of inferring social groups in personal archives. This is an important problem since a lot of personal data involves some social context, and using the people involved in the context can be a useful way to organize and retrieve information. However, a long-term archive can easily involve thousands of people and therefore, it is more efficient and intuitive to associate items with significant groups, rather than just with individuals.

While it is tedious for users to construct social groups by hand, we exploit the observation that social groupings are captured implicitly in routine communications, photographs, and others forms of personal data. For example, the recipient list of every email message provides a hint that there's some commonality between the recipients of the message. We can mine this information to automatically infer the people who are likely to be in the same group, and the groups likely to be the most significant in the corpus. To achieve this goal, we had to come up with a novel algorithm appropriate for social contexts where groups may be overlapping, nested, or consist of single individuals. As mentioned in the previous chapter, this algorithm is implemented in Muse, where it is used to organize the archive and browse and filter email archives by group. However, it is generally applicable to various settings where it is useful to infer a fixed or variable number of groups. As a result, we will study several variants of the algorithm suitable for different applications. Our algorithm has also been used to help users set up friend lists for nuanced sharing in

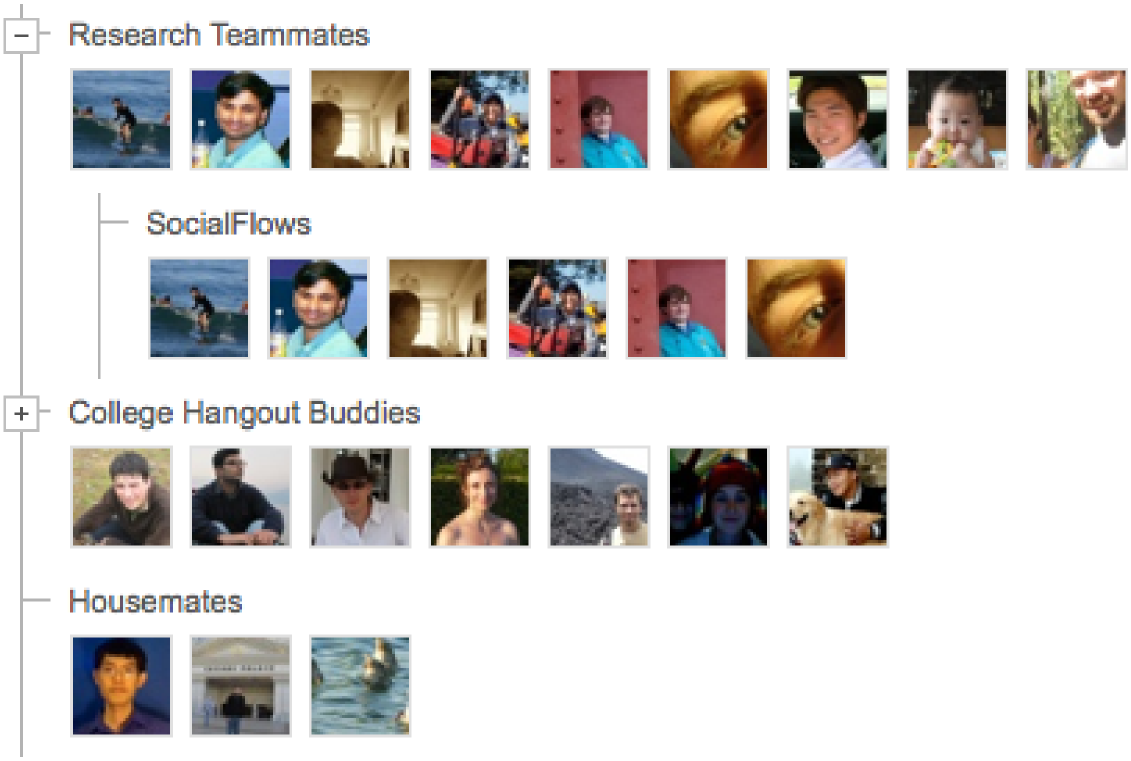


Figure 3.1: An example of a social topology.

social networks, which can address some privacy problems. Setting up friends lists manually is a cumbersome task left undone by most users [120].

In its general form, the group extraction problem takes as input a list of groupings of people, and tries to derive a small number of overlapping and nested groups that best represent the input. We have designed a novel algorithm for this task and evaluated it on a data set consisting of about 2,000 personal email accounts and 1,100 tagged Facebook photograph collections.

3.1 Social Topology

We originally proposed the concept of a *social topology* as the structure and content of a person's social affiliations, comprising a set of overlapping and nested groups [76]. Figure 3.1 shows several defining properties of a social topology. First, any individual in the topology may appear in several groups. This models people who play several roles in the user's life, such as being both a colleague and a friend. Second, it may contain manufactured groups – group of people who never occur together in a single item in the original data set. Consider a university lab whose membership changes annually: a "core" group, such as a faculty team, might persist in the group over time, but all the group members never appear on a single message. Third, social topology groups may be nested. This captures specific subgroups within a supergroup, such as immediate family within an extended family. Finally, a "group" may consist of just a single individual who is sufficiently important.

Formally, we formulate the problem of deriving a social topology as follows: given a data set d of group communications data, a value function v which measures the significance of a group with respect to d, and a budget of b groups, find b groups whose aggregate value is maximized. We derive these groups only from the data that is directly visible to the user, making these groups ego-centric.

In brief, this chapter contributes:

- A family of greedy algorithms for constructing social topologies from many input instances of grouping. The algorithms make different trade-offs and can be tuned based on the target application. These algorithms are embedded in Muse as well as in a Facebook application called GroupGenie.
- A validation and comparison of our algorithm using two data sets: a collection of 1,995 personal email archives containing over 24 million sent email messages and a set of 286,038 tagged photos from 1,099 Facebook users.
- An evaluation and comparison of social topologies constructed from these data sets. The evaluation includes a comparison with Newman's clustering algorithm using edit distances as an information-theoretic metric.

A Java implementation of our algorithm is available as part of the Muse source code and can be used relatively independently of the rest of Muse.

3.2 Related Work

Analysis of social data uses 2 broad approaches: one uses a global view of the network (e.g., [65, 69]) and the other involves an ego-centric view, i.e., one person's view of the network (e.g., [26, 44, 83]).

Clustering algorithms aim to elicit communities from a graph structure. Traditional algorithms based on hierarchical agglomerative clustering partition the input graph, disallowing node overlap between clusters [24, 93]. This approach is not suitable for our purposes, as one person can adopt several social roles simultaneously.

Palla et al. present an algorithm that discovers overlapping communities in global, unweighted networks [97]. Communities are generated in a bottom-up fashion from k-cliques. The algorithm relies heavily on the structure of the input graph, and does not permit weighted edges. In our case, however, the "weights", or frequencies, of communications are significant; moreover, their algorithm requires significant computation time, which is unsuitable for real-time use by consumers.

While there is work on detecting overlapping clusters (e.g, Banerjee et al. [10], Huberman et al. [139], Lancichinetti et al. [66]), all these approaches make the assumption that the global structure of the network is available. Second, the input model of the graph is simply edges between people, ignoring grouping information even if it is available. In contrast, we retain this information and our algorithm performs operations on groups instead of on individuals. Third, they are evaluated on networks formed by publicly available information, while we evaluate our algorithm on personal data, where there may be different patterns of group formation.

Visualization and interface techniques such as ContactMap [147], Vizster [55] and LinkedIn InMaps [72] help users view and organize their social networks. We have previously published a different algorithm to derive overlapping and hierarchical groups, and an interface to edit those groups [76]. This algorithm required the use of several parameter settings and was evaluated in a smaller study involving email data sets of 19 users; however, it does not seamlessly handle individuals. In contrast, the work reported in this chapter presents an algorithm with better accuracy, and has been evaluated on a larger scale on multiple data sets.

Association rule mining is a technique for finding related item sets in a corpus, given a specific seed [3]. Roth et al. present a group-finding algorithm for Gmail in which the goal is to complete the group as accurately as possible given an initial seed [115]. Like us, they assume that communications reflect implicit social structure, and use communication frequency as a proxy for tie strength. They develop an interactions rank metric that gives an ordering over unique recipient groups, allocating points according to communication frequency, recency, and direction. However, seed-based approaches are generally inadequate for the purposes of helping users construct a social topology; for example, Gmail users cannot access the set of probable groups or use them for other purposes. As a result, the algorithm does not directly create a summary of the input groups.

3.3 Algorithm Description

We now describe our core algorithm that generates social topologies from a single user's ego-centric grouping data such as email archives or tagged Facebook photographs. The algorithm is in part inspired by, and a simplification of, an earlier algorithm we had designed for the same purpose [76]. Towards the end of this chapter, we describe some enhancements that improve the algorithm by incorporating user feedback for the specific case of email archives.

3.3.1 Choosing Representative Groups

We define a social topology to be a set of unique, potentially overlapping and nested groups, comprised of people drawn from the set of all people mentioned in an archive. Permitting nested groups allows fine granularity in the topology, while permitting overlapping groups allows us to represent people who play multiple roles.

Ego-centric group communication data sets already contain a natural social topology: the unique groups that occur together on items in the data set. Consider a user's archive of all sent email. The natural social topology would simply include all the unique recipient sets. However, the size of this topology is rather large and unwieldy

– typically in the thousands. Therefore, our task of social topology construction is essentially one of compression, in which we reduce the natural social topology into a manageable size, while retaining as much information as we can.

Depending on the objective, there are different trade-offs in generating this compression. For example, we may wish to create a topology that includes mostly core persons from different parts of the input. Alternatively, we may wish to create a topology covering many people, combining those who seem to be related to each other.

3.3.2 Intuitive Approach

Our overall goal is to identify a small number of groups that represent the input as well as possible. To this end, we assign each unique group in the input a value that is based on its frequency in the input and the number of members it has. The higher the frequency, the more important the group, and therefore the higher the value. Starting with this set of unique groups, our algorithm iteratively reduces the number of groups till the desired number of groups is reached, while maximizing the aggregate value of groups in the topology.

In each iteration, the algorithm applies one of several moves like intersecting or merging two groups or dropping a group. The possible moves are described in section 3.3.6. Each move reduces the size of the topology by one and incurs a loss in value because it loses some information – the input is represented by one less group. The loss in value for each move is computed by mapping each group in the input to a group in the compressed summary, with a configurable penalty associated with oversharing, which is described below in section 3.3.4. For example, if an email message is mapped to a group that contains more members than the actual recipients of the message, the topology loses some value. At each step, the algorithm greedily chooses the move that incurs the lowest loss in value.

3.3.3 Problem Statement

To state the problem precisely, we now introduce the notion of a representation map that maps each group in the input to one of the groups in the output, and a value function that uses this map to estimate the value of each group in the output. The cumulative value of a social topology is simply the sum of all the groups in it.

Formally, we define the social topology compression problem as follows. Given

- a set of friends F,
- a natural social topology S consisting of unique groups $g \subset F$, where the value function $v_0(g)$ denotes the significance of the group g,
- a size b which is our budget, or number of groups allowed in the final topology,
- a value function $v(g, r)$, defining the value of input group $r \subseteq S$ mapped to output group g.

find a social topology S' consisting of unique groups $g \subset F$ and a representation map R, mapping each $g \in S'$ to non-overlapping subsets of S, such that $\sum_{g \in S'} v(g, R(g))$ is maximized.

3.3.4 The Sharing Value Metric

Our value function is based on a model of information sharing and over-sharing. Intuitively, if group g in the original social topology maps to group g' in the final social topology, the relative value should be high if g' has many members in common with g and should suffer over-sharing penalties if g' has many members who were not in g. The ratio between the number of common elements to the size of g determines the fractional positive contribution of g's value to g'. Since different uses of social topologies may desire different over-sharing penalties, the algorithm is parameterized with a penalty function $w(f, g)$ that determines the penalty to be applied to each unit of over-sharing with friend f not in group g. Thus, the value function based on information sharing can be formally defined as

$$v(g', r) = \sum_{g \in r} \frac{v_0(g)}{|g|} \left(|g' \cap g| - \sum_{f \in (g' - g)} w(f, g) \right),$$

where $w(f, g)$ is the over-sharing penalty to be applied to friend f for group g.

A simple penalty weighting function is a constant, i.e.,

$$w(f, g) = C$$

If C is 0, there is no penalty for over-sharing; if C is 1, every person that a data item is over-shared with costs as much as the value contributed by a person who was in the original group that the item was shared with.

We also evaluate a more sophisticated weighting function that factors in the relationship between the original group and the specific friends an item was over-shared with. For example, it is desirable to lower the penalty if the over-sharing is with a friend who is in the same social space as the original group (i.e., present in many other messages with this group), compared to a person from a completely different sphere. For example, if $P(\overline{f}|f')$ denotes the conditional probability of not finding f in groups containing f', we could define the over-sharing penalty thus:

$$w\,(f, g) = \frac{1}{|g - \{f\}|} \sum_{f' \in (g - \{f\})} P(\overline{f}|f')$$

3.3.5 A Greedy Algorithm

Our algorithm works by starting with the original set of groups in the input, and then applying a set of permissible actions, called moves, on these groups. Each move reduces the social topology size by 1, and also reduces the value of the topology, because some information is lost when the size of the topology is reduced. For each move, our algorithm greedily picks the move with the lowest loss in value until the topology attains the desired size b.

The possible moves and their error functions are defined below. The initial representation mapping R simply maps each group g to itself; if g is a group in the original topology, $v(g, \{g\}) = v_0(g)$.

- DISCARD. Discard a group from the topology, thus losing the group's entire value.

$$E_{\text{DISCARD}}\left(g, r\right) = v\left(g, r\right)$$

- MERGE. Merge two groups to create a union that inherits the combined value, appropriately penalized to account for their membership mismatch. The over-sharing penalty built into the value metric ensures that the most closely related groups have the lowest error.

$$\begin{gathered} E_{\text{MERGE}}\left(g_1, r_1, g_2, r_2\right) = v\left(g_1, r_1\right) + \\ v\left(g_2, r_2\right) - v\left(g_1 \cup g_2, r_1 \cup r_2\right) \end{gathered}$$

- INTERSECT. Intersect two groups to capture the importance of a shared subset.

$$\begin{gathered} E_{\text{INTERSECT}}\left(g_1, r_1, g_2, r_2\right) = v\left(g_1, r_1\right) + \\ v\left(g_2, r_2\right) - v\left(g_1 \cap g_2, r_1 \cup r_2\right) \end{gathered}$$

- TRANSFER. Partially transfer the value of the second group to the first, taking into account the over-sharing penalty, and discard the second.

$$\begin{gathered} E_{\text{TRANSFER}}\left(g_1, r_1, g_2, r_2\right) = v\left(g_1, r_1\right) + \\ v\left(g_2, r_2\right) - v\left(g_1, r_1 \cup r_2\right) \end{gathered}$$

To illustrate with a concrete example, assume there are 2 groups, $\{A, B, C\}$ and $\{A, B, D\}$, with some values. Some examples of possible moves are: a) drop $\{A, B, C\}$; b) merge the two groups to form $\{A, B, C, D\}$, dropping both of the original groups; c) intersect the two groups to form $\{A, B\}$ and drop the original groups; d) transfer some value from $\{A, B, D\}$ to $\{A, B, C\}$, dropping the former. The move picked would be the one with the least error, computed according to the definitions above.

3.3.6 An Approximate Algorithm

In the algorithm above, each group's value is defined as a function of the values of the original groups they represent. As a simplification, we assume each member in a derived group contributes equally to the group's value. We can compute the error term for each move based on the approximate value of its operands. The approximate value function is defined as

$$\overline{v}(g) = \begin{cases} v_0(g), & \text{if } g \in S, \\ \overline{v}(g_1) + \overline{v}(g_2) - E_m(g_1, g_2), & \text{if } g = m(g_1, g_2) \end{cases}$$

where $m(g_1, g_2)$ is the result of applying move m to g_1 and g_2. (Unary moves like discard are similarly defined.) With our assumption that the value of a group is considered uniformly distributed across all its members, the over-sharing error simply depends on the ratio of additional people compared to the original group. The error functions are thus analogously defined as:

$$\begin{aligned}
&E_{\text{DISCARD}}(g) = \overline{v}(g) \\
&E_{\text{MERGE}}(g_1, g_2) = \\
&\qquad \textstyle\sum_{f \in g_2 - g_1} \frac{\overline{v}(g_1)}{|g_1|} w\left(f, g_1 \cup g_2\right) + \\
&\qquad \textstyle\sum_{f \in g_1 - g_2} \frac{\overline{v}(g_2)}{|g_2|} w\left(f, g_1 \cup g_2\right) \\
&E_{\text{INTERSECT}}(g_1, g_2) = \sum_{f \in g_1 - g_2} \frac{\overline{v}(g_1)}{|g_1|} + \sum_{f \in g_2 - g_1} \frac{\overline{v}(g_2)}{|g_2|} \\
&E_{\text{TRANSFER}}(g_1, g_2) = \\
&\quad \textstyle\sum_{f \in g_1 - g_2} \frac{\overline{v}(g_2)}{|g_2|} w\left(f, g_1 \cup g_2\right) + \sum_{f \in g_2 - g_1} \frac{\overline{v}(g_2)}{|g_2|}
\end{aligned}$$

With these definitions in place, our approximate algorithm is conceptually quite simple, and is summarized in Algorithm 1. In practice, we implement several optimizations to ensure that each iteration of the loop is as efficient as possible by keeping the list of best possible moves per group in a heap. We also maintain dependencies between groups and moves so that we can efficiently recompute best moves when a group is dropped or changed.

Algorithm 1 compressTopology(S)

```
Input   Initial topology $S = \{g_i\}$,
         a set of unique groups and values $\overline{v}(g_i)$.
Input   A budget $b$, the size of the final topology.
Output $S$, the final topology

while $|S| > b$ do
    $g^* \leftarrow m(g_1, g_2)$ where $m$ is the lowest loss move $\forall g \in S$
    $\overline{v}(g^*) \leftarrow \overline{v}(g_1) + \overline{v}(g_2) - E_m(g_1, g_2)$
    $S \leftarrow S + g^* - g_1 - g_2$
end while
```

3.4 Experimental Evaluation

We have evaluated different versions of our algorithm on 2 types of data sets, one using (anonymized) personal email archives, and another using Facebook photo tags.

3.4.1 Email

Our email data set is comprised of email headers from 1,995 users' personal email archives, totaling over 24 million sent email messages. The data set, provided by Xobni Inc., was collected from a subset of users of their Xobni Cloud service. The data we received was fully anonymized; all personally-identifiable information had been removed, and we had no access to message contents. Most of these users connected to Xobni via the Microsoft Outlook client, so we expect that much of the email activity was work related. Figure 3.2 outlines statistical properties of the corpus. Note that most people have thousands of groups. We restrict our algorithm input to sent email only, noting that this is a more accurate signal for social importance, as sending an

	Messages	People	Groups	Group Size
Lower Quartile	2,038	329	373	1
Median	6,640	738	1,104	1
Upper Quartile	14,684	1,422	2,451	1
Max	159,697	20,813	24,306	2,825
Mean	11,521	1,109.5	1,814.9	1.5
Std Dev	15,205.1	1,328.6	2,231.4	2.3
Total	24,228,571	2,213,486	3,816,668	35,781,399

Figure 3.2: Summary of the 1,995-person email data set.

email incurs a cost on the user, whereas receiving one does not. This decision also has the advantage of excluding spam messages. We did see some startling anomalies in the data set, such as an individual who sent as many as 160,000 messages, and a message addressed to 2,826 recipients. Note that the majority of messages are sent to only one person.

3.4.2 Tagged Photos

Just as emails capture co-occurrence of recipients on mails, tagged photographs capture physical co-occurrence of people and also provide evidence of grouping. Given the fact that photo sharing is one of the most popular forms of online social activity, tagged photographs are an excellent source of social topology data. To evaluate our algorithm on tagged photos, we developed GroupGenie, a publicly released Facebook application that allows Facebook users to infer their social topology from their tagged photos.

GroupGenie users have found the social groupings suggested to them by our algorithm helpful for both data sharing and communication tasks, and for a certain degree of personal self-reflection. An informal pilot study of about 30 users aged 17-19 found that the groups suggested to them were good enough, with a few minor edits, to publish to their profile pages as Facebook Featured Friends [37]. Some found it useful to use their groups in Facebook Chat [36] to do group-wide chats.

At the time of evaluation, 1,099 Facebook users had used GroupGenie. Most of

	Photos	People	Groups	Group Size
Lower Quartile	31	28	19	1
Median	106	62	54	2
Upper Quartile	325	130	142	3
Max	3,062	594	1,050	111
Mean	260.3	90.9	109.6	2.4
Std Dev	392.2	88.8	143.7	2.8
Total	286,038	99,910	120,457	682,126

Figure 3.3: Summary of the 1,099-person Facebook photo tags data set.

these users discovered GroupGenie through friends, and from a news article about an earlier version of our work [121], suggesting strong interest among Facebook users in tools to help them create groups.

Figure 3.3 provides summary statistics of the tagged photograph corpus. Note that the owner of the Facebook account, if present, is excluded from the input groups. There are notable differences between this data set and the email data set. In particular, the average group size in a photograph is 2.4, compared to the average number of recipients on an email, which is 1.5. Moreover, more than half of the photographs are tagged with at least 2 people excluding the user; in contrast, a majority of emails involve only one other person excluding the user. On the other hand, there are significantly fewer tagged photographs per user than email messages, presumably due to the larger effort required to take, upload and annotate photographs.

3.4.3 Algorithm Versions

To characterize our algorithms and data better, we conducted experiments with four variants of Algorithm 1:

DISCARD. Considers only discard moves. This is a strawman version of the algorithm that simply returns the top b initial valued groups for a given budget b.

MERGE. Considers discards and merges, with a simple fixed penalty weight of 0.5.

COND-MERGE. Considers discards and merges, with a conditional probability metric for sharing penalty.

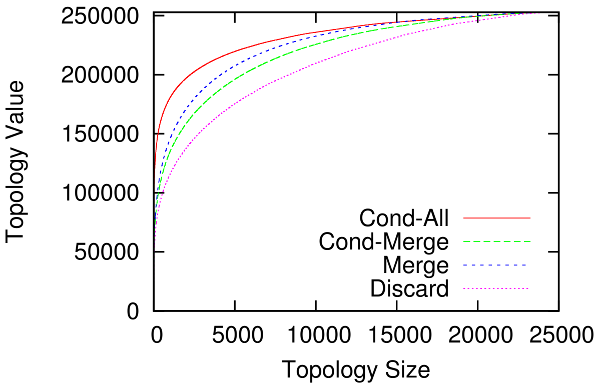


Figure 3.4: Social topologies for a representative data set.

COND-ALL. Considers all moves (discard, merge, intersect, transfer), with a conditional probability metric for sharing penalty.

We define the initial value, or significance, of each input group g as $v_0(g) = min(|g|, sizeThreshold) \times count(g)$, where $count(g)$ is the number of times g appears in the input. Intuitively, this captures the value of group g in the original input. The parameter $sizeThreshold$ prevents large groups, which are often one-off mailing lists, from being awarded excessively large initial values. Empirically, we set $sizeThreshold = 20$.

3.5 Groups in Email Archives

To provide insight into our algorithm, we first present its behavior on a single, representative user's data, for different sizes of the output topology. As shown in Figure 3.4, all algorithm variants capture a significant fraction of the total value with a small percentage of groups, with DISCARD, COND-MERGE, MERGE, and COND-ALL in increasing order of value captured for a given topology size. DISCARD allows no oversharing; its sharing penalty is effectively ∞. COND-MERGE allows sharing preferably among those who are already sharing other input items. For MERGE with a fixed penalty weight of 0.5, the algorithm is allowed to perform more merging.

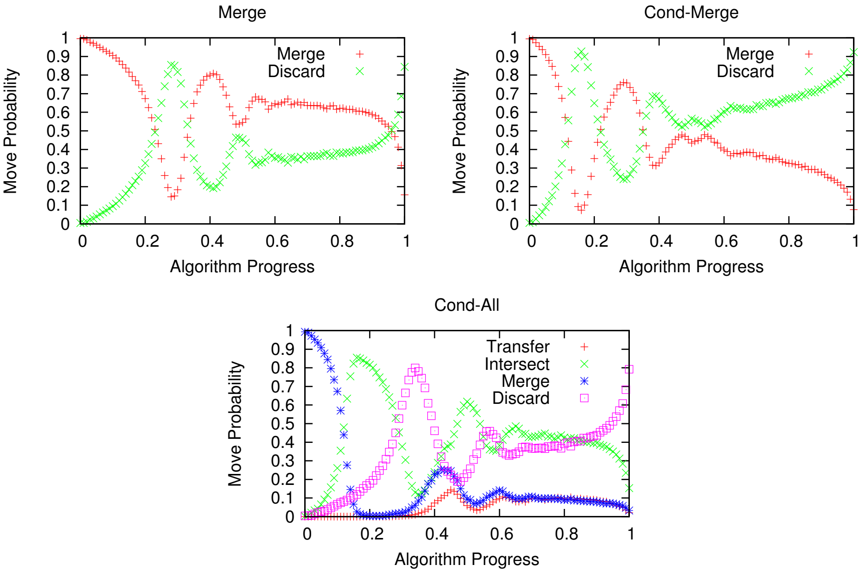


Figure 3.5: Algorithm behavior over the email corpus.

COND-ALL has the best compression ratio, though we find that it tends to discourage merged groups in the final topology. Because the value of one group can be transferred to another with a sharing penalty, COND-ALL tends to identify the super individuals or groups that may play different roles in a user's interaction. Consider, for example, a secretary who is carbon-copied on all work-related emails. The secretary can amass a very large value as partial credit is transferred to him when low-frequency groups are dropped.

Aggregate Behavior

Let us now try to understand the algorithm's behavior in terms of the distribution of move types. Figure 3.5 shows move frequency plotted against normalized fractional algorithm progress, aggregated over the entire email data set.

We see that in the MERGE and COND-MERGE variants, there are distinctive alternating phases of merges and discards. The periodicity reduces over time with merges dominating at the beginning and discards dominating near the end. As the algorithm takes the move with the minimum value reduction, the periodicity results from the fact that there are many initial groups with values 1, 2, and so forth. Many discards of groups of value 1 kick in as the minimum drop in the algorithm reaches 1. Since the merged groups no longer have integral values, the choice between discards and merges become more irregular. Near the end of the algorithm, the remaining groups are distinct enough that merging them would incur a higher penalty than discarding them, thus we see many discards near the end. The COND-MERGE variant is similar to MERGE, except that MERGE performs more merges since it has a lower sharing penalty.

COND-ALL has two more moves than MERGE: intersects and transfers. Almost all the intersect moves occur between supersets and subsets. In such cases, intersects produce the same topology as discards of the larger group, but the smaller group now accumulates more value due to the transfer of value. Including this move favors the creation of smaller groups and emphasizes the core people in each group. Similarly, transfer moves also create pressure to produce smaller groups, since values can be transferred from one group to another. Together, intersect and transfer moves reduce the number of merges.

3.5.1 Value Concentration

How does overall value reduce as we reduce the number of groups? Figure 3.6 plots the median fraction of summary groups that capture a given fraction of value.

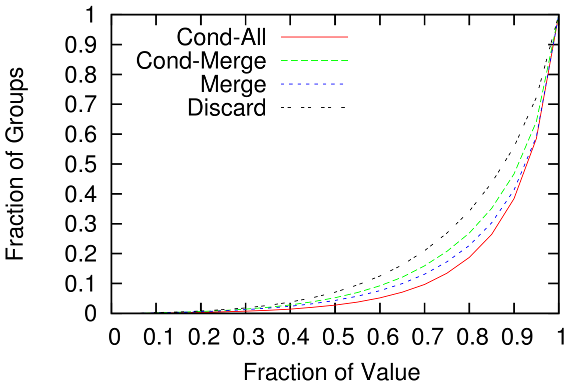


Figure 3.6: The values of social topologies obtained for the email corpus.

We see from the summary in Figure 3.7 that 50% of the value can be captured

	DISCARD	MERGE	COND-MERGE	COND-ALL
50%	0.07	0.04	0.05	0.03
60%	0.13	0.08	0.09	0.05
70%	0.21	0.13	0.16	0.10
80%	0.34	0.23	0.27	0.19

Figure 3.7: Fraction of groups needed to achieve a given fraction of the value.

by the DISCARD variant with just 7% of the original groups. The other algorithms can compress the social topology further. MERGE needs only 4% of the number of groups to capture 50% of the value, and 23% of the groups (versus DISCARD's 34%) to capture 80% of the value. COND-MERGE promotes the merging of closely related friends, and needs slightly more groups (27% for the same value). As discussed above, since COND-ALL allows the value of a group to be transferred to another, without having to include all members of the group, COND-ALL achieves the best value with the smallest number of groups. To reach 80% of the value, COND-ALL needs a social topology whose size is less than 20% of the original, indicating a rough "80-20" rule.

3.5.2 Fixed Size Social Topologies

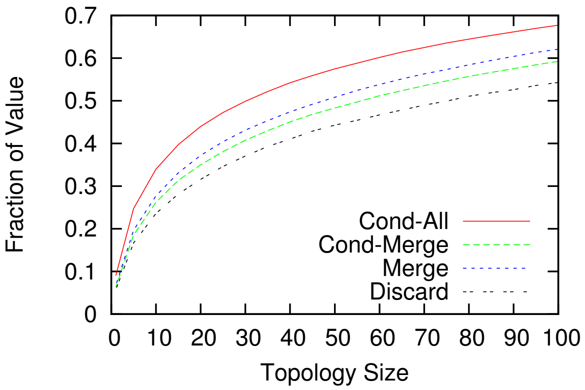


Figure 3.8: Values of fixed-size social topologies.

If the application requires a human to review and make sense of the social topology,

	DISCARD	MERGE	COND-MERGE	COND-ALL
10	0.24 (2)	0.28 (4)	0.26 (3)	0.34 (3)
25	0.35 (8)	0.40 (11)	0.38 (10)	0.47 (8)
50	0.44 (21)	0.51 (25)	0.49 (24)	0.57 (18)

Figure 3.9: Values of social topologies with a fixed number of groups. The number of non-singleton groups is shown in parentheses.

a topology consisting of even 10% of the number of starting groups can be overwhelming, since email archives commonly start with over 1,000 groups. What if we wanted to derive a small, fixed number of groups which can then be manually reviewed by a user? Muse is one such application, since the user is presented with graphs of communication activity with the top inferred groups. Each group is assigned a color when presenting the monthly summaries, which also limits the number of groups that the user can deal with at one time. Therefore, when this algorithm is used in Muse, we typically set its limit to 20 groups by default, though the exact number can be configured by the user.

Figure 3.8 shows the median of the values captured for the fixed size social topologies and Figure 3.9 tabulates the values for topologies with 10, 25, and 50 groups. We find that the top 10 groups capture 24-34% of the value and the top 50 groups capture 44-57%, depending on the algorithm variant used.

Our algorithm gracefully treats groups with a single person the same as any other, allowing us to rank individuals uniformly alongside groups. However, certain applications may not have a use for singletons. For example, a tool that helps users name groups only needs to show non-singleton groups, since individuals already have a name. We list the number of non-singletons in Figure 3.9 for reference. Not surprisingly, the majority of the top groups in email turn out to be singletons. As the allowance for over-sharing grows from cond-all, cond-merge, to merge, the number of non-singleton groups increases slightly. Thus for applications that work with only non-singletons, just 2-4 groups are needed to reach 24-34% of the value and 8-11 groups reach 35-47%.

Which version of the algorithm should an application use? The different variants

	DISCARD	MERGE	COND-MERGE	COND-ALL
Non-singleton	21	25	24	18
New groups	0	14	6	0
Group size	2.6	6.1	3.5	2.5
People covered	60	162	84	71
Avg. groups per person	2.0	1.8	1.9	1.6

Figure 3.10: Properties of social topologies of size 25 in the email data set.

produce different topologies. Figure 3.10 shows additional properties of social topologies of size 25. It is clear from the figure that MERGE generates the largest social topology in terms of number of people covered, followed by COND-MERGE. Therefore, if it is important that more people be covered by the groups, these variants can be used.

On the other hand, COND-ALL is suitable for distilling key members of each group. We observe that no new groups are created for the COND-ALL case; since it tends to identify the core groups, which are likely to have been emailed together at least once.

3.5.3 Inferring the Number of Significant Groups

For applications without a fixed budget, how can we automatically set the number of groups? We can leverage our valuation framework to try and make this decision within the algorithm. The average value of a group in the input data set serves as a baseline for each individual's corpus. We can identify groups that stand out by simply running our algorithm and reducing the number of groups till the error for a move exceeds a threshold of say, 1 standard deviation above the baseline. Figure 3.11 shows that, using this technique, a median of 11 groups was directly identifiable from the input data set (DISCARD); COND-ALL and COND-MERGE identified a median of 15 significant groups for the email data sets and MERGE identifies 14. This approach seems to quite reasonable in practice, and automatically calibrates the number of significant groups to the input data.

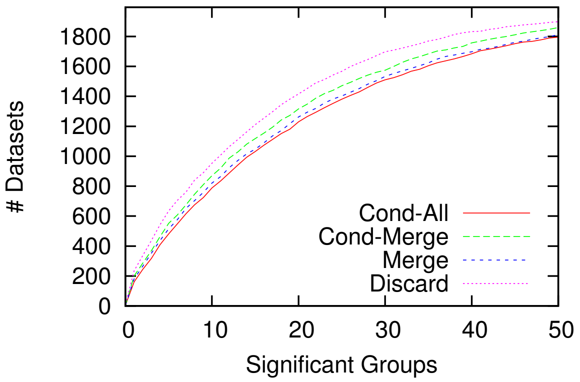


Figure 3.11: The cumulative distribution of the number of significant groups in the email corpus, with a threshold of one standard deviation above the baseline.

3.6 Groups in Facebook Photos

Let us now similarly analyse the four variants of the greedy algorithm described in the previous section for Facebook photos data set. Once again, all figures represent the median observed in the data set.

3.6.1 Value Concentration

We observe the same overall trends with the photo data set as we saw in the previous section with the email data set. In Figure 3.12, the fractional value curve climbs less steeply than in Figure 3.6, suggesting higher diversity in the photo data set compared to email. Figure 3.13 shows that COND-ALL requires 15% of the groups to capture 50% of the value, and 42% to capture 80% of the value.

One observed difference from the email data set is that all variants other than DISCARD have almost identical curves. This suggests that the photo tags may be capturing tighter friendships since the tendency of the COND-ALL variant of the algorithm to track core friends is similar to the MERGE variant which tends to create larger groups including more peripheral relationships.

From Figure 3.13, we see that DISCARD needs 26% of the groups to capture 50% of the value, whereas COND-ALL needs only 15%. That is, COND-ALL is better than DISCARD at compressing the social topology by a factor of 1.7. COND-ALL has a

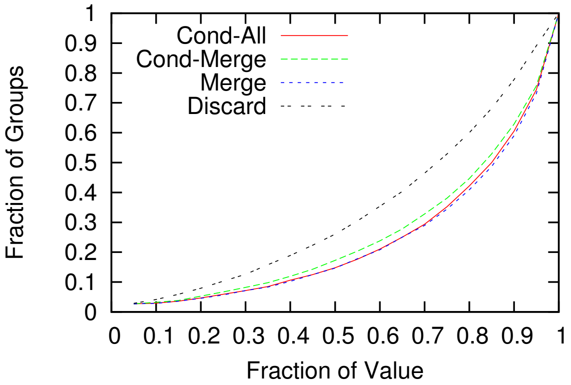


Figure 3.12: The values of social topologies obtained for the photo corpus.

	DISCARD	MERGE	COND-MERGE	COND-ALL
0.5	0.26	0.15	0.17	0.15
0.6	0.35	0.21	0.24	0.21
0.7	0.46	0.29	0.33	0.29
0.8	0.60	0.41	0.45	0.42

Figure 3.13: Fraction of groups needed to achieve given fraction of the value for photos.

compression improvement of 1.4 to 1.7 times for photos over DISCARD; it has an improvement of 1.8 to 2.6 for email.

3.6.2 Fixed Size Social Topologies

If we wish to help users create and use a manageable number of Facebook friends lists, it is important that we do not overwhelm them with too many groups. Even though a higher fraction of groups is needed than email, since photos are a smaller data set, the value is captured by a relatively small number of groups. Figure 3.14 shows the median of all values obtained for group sizes up to 100, and Figure 3.15 tabulates results for group sizes of 10, 25 and 50, along with the number of non singleton groups in the output. For example, with just 10 groups, 60% of value is captured by the COND-ALL algorithm, compared to 34% for the email data set. We see that

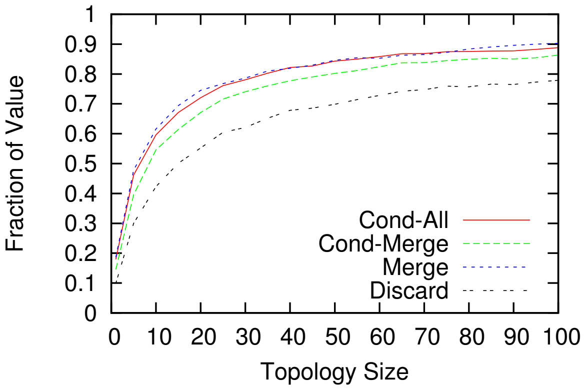


Figure 3.14: Values of small social topologies derived from photos.

	DISCARD	MERGE	COND-MERGE	COND-ALL
10	0.42 (8)	0.62 (9)	0.55 (8)	0.60 (7)
25	0.60 (21)	0.77 (21)	0.72 (21)	0.76 (18)
50	0.70 (42)	0.85 (41)	0.80 (42)	0.84 (37)

Figure 3.15: Values of photo-based social topologies with selected sizes. Non-singleton groups are shown in parentheses

the percentage of non-singleton groups is much higher, reflecting the fact that photo-taking is a gregarious activity, unlike email which often involves correspondence with only one other person.

More characteristics of social topologies with 25 groups are shown in Figure 3.16. Note that the number of non-singleton groups included here are determined more by the data set than the algorithm. In this case, even the COND-ALL variant creates a few new groups; it is harder to take a photo of a cohesive but broad group, whereas it is common to write at least one message to it. The median of the average group size is much higher across the board. MERGE still derives larger groups and includes more people, but not substantially more. The results show that people on average participate in about two groups, confirming the importance of our algorithm's ability to find overlapping groups.

	DISCARD	MERGE	COND-MERGE	COND-ALL
Non-singletons	21	21	21	18
New groups	0	11	4	1
Group size	4.5	6.9	4.8	2.9
People covered	54	90	64	57
Avg. groups per person	2.1	2.0	1.9	1.8

Figure 3.16: Properties of social topologies of size 25 in the Facebook photos data set

3.6.3 Significant Groups

Photo tagging data shows a distinctly lower number of significant groups than the email data set. In Figure 3.17 we see that the photo data set has a median of 7 significant groups.

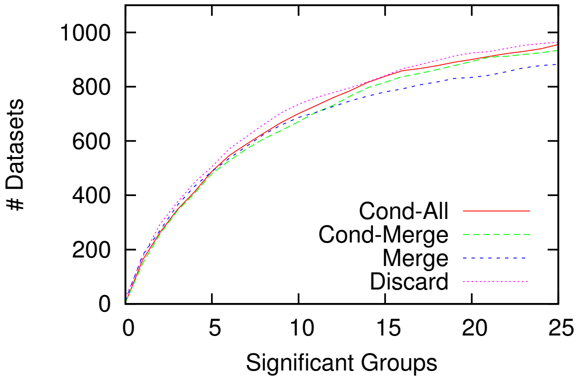


Figure 3.17: The cumulative distribution of the number of significant groups for the photo corpus.

3.7 Comparison with Newman's Algorithm

We now compare our algorithm variants with Newman's fast greedy clustering algorithm [93], which is a commonly used algorithm for discovering communities in social graphs. Newman's algorithm partitions the nodes in the graph into clusters via optimization of a modularity metric. It disallows overlapping groups, and automatically picks the number of groups to output. We simply treat all the clusters generated by

	Group Size	Number of Groups	Fractional Edit Distance
Email	1	118	0.93
Photos	3	6	0.84

Figure 3.18: Median group parameters and edit distance ratios for Newman clustering.

Newman's algorithm as the social topology for a user. We used the implementation of Newman's algorithm in the igraph package of R.

So far, we have characterized our algorithm with respect to the notion of group values. To objectively compare our algorithm with Newman's algorithm, we select edit distance, a measure that is not a direct objective for either algorithm. The edit distance between two words is defined as the minimum number of character alterations required to modify one of the words until it is equivalent to the second. Intuitively, this metric captures the minimum number of insertions and deletions needed to derive each input group starting from one of the groups in the social topology. The fractional edit distance is the ratio of the sum of the minimum edit distances for each input group, divided by the size of the input, i.e., the sum of the number of people in all input groups. Obviously, a lower fractional edit distance is better, because fewer edits are required to reproduce the input from its summary.

Formally, the edit distance for a collection of groups C given a social topology S is

$$\text{EditDistance}(S, C) = \sum_{c \in C} \text{min}_{s \in S}(|c \cup s| - |c \cap s|)$$

We performed an experiment where we compute the fractional distances for both the email and photo data sets using Newman's algorithm and the four variants of our algorithm. Figure 3.18 shows that the fractional edit distances with Newman's algorithm are fairly high, with the medians being 0.93 and 0.84 for email and photos, respectively.

For our algorithm, the fractional edit distance metric depends on the number of groups in the social topology. The medians of the fractional edit distances are plotted for different topology sizes in Figure 3.19. The results show that our algorithm clearly outperforms Newman clustering; all variants of our algorithm beat the clustering

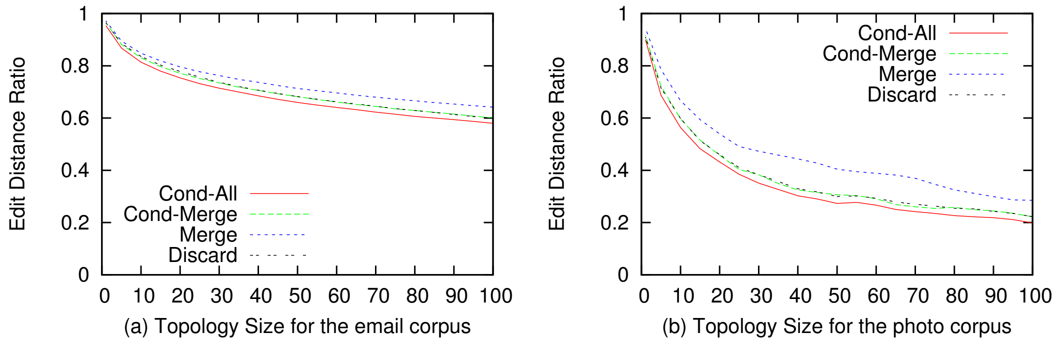


Figure 3.19: Comparison of the EditDistance metric across all 4 algorithm variants for the (a) email corpus and (b) photo corpus.

algorithm even with a budget of just 4 groups for email and 3 groups for photos. There is a significant difference in fractional edit distances between the email and photo data sets. 10 groups generated by cond-all yield median ratios of 0.81 and 0.56 for emails and photos respectively; 25 groups yield ratios of 0.74 and 0.38.

While most of our algorithm variants perform similarly we note that merge produces a worse fractional edit distance than discard. This makes sense given that merge uses a penalty of 0.5 for over-sharing whereas the penalty of a deletion for edit distances is 1. The goal of the merge variant is to find related people and not to optimize edit distance.

3.8 Application in Muse

To illustrate a concrete application of the algorithm described in this chapter, let us take the example of how it is used in Muse. We implemented the algorithm in Muse to automatically derive groups from email archives and solicited user feedback about the groups. The default number of groups is set to 20. Users were broadly happy with their groups, since graphs like the one in Fig. 2.3 tend to reflect important events such as the start of a large project, the beginning of a new relationship, a move to a new town, etc. For example, one user emailed us to say *"Groups view is SUPER useful, since those people are probably all related in some 'thread' of my life... totally makes sense"*. Of course, the algorithm does not always come up with the perfect groups, but users can always apply a few refinements using the groups editor (Fig. 2.2). Our users also provided examples of limitations, based on which we performed 2 minor modifications to the algorithm.

First, users found that Muse was sometimes identifying groups that were very significant in terms of communication volume, but not as meaningful from a personal point of view. Consider scenarios where people come together to organize an event, or students work together on a quarter-long project at a university. There tends to be a lot of bursty communication in such groups, which dies out after the specific event or project. Users generally prefer to see groups that emphasize more long-term relationships, like siblings or close friends, even if they do not carry as high a volume. To address this requirement, Muse attenuates the value due to bursty communications by modifying the definition of initial group value in section 3.4.3. Empirically, we found the square root function worked well for attenuation. Therefore, if a group g occurs with frequency $count(g, m)$ in month m, the initial value of the group is:

$$v_0(g) = min(|g|, sizeThreshold) \times \sum_m \sqrt{count(g, m)}$$

We also set $sizeThreshold$ to 10 in this application based on user feedback; recipient lists larger than 10 tend to be related to very large, one-off events or automated announcement lists.

A second enhancement requested by users was particularly relevant to family groups. One user pointed out, *"These people have the same last name... surely your algorithm should have inferred that they belong to the same group."* The same last name frequently captures members of a family, and similarly, the same domain name frequently captures affiliations with a company, organization or university. To account for such scenarios, we introduced the notion of affinity between pairs of individuals who have the same last name, or the same domain in any of their email addresses:

$$affinity(p1, p2) = 0.25 \times sameLastName(p1, p2) + 0.1 \times sameDomain(p1, p2)$$

The function *sameLastName* returns 1 if any of $p1$'s names has the same last name as any of $p2$'s names (due to entity resolution, a person can have multiple associated names); it returns 0 otherwise. Similarly, the function *sameDomain* returns 1 if any of $p1$'s email addresses have the same domain as any of $p2$'s and the domain does not belong to a list of well-known providers; 0 otherwise.

We can now generalize this notion to group affinities, which are computed as the sum of pairwise affinities between people who are not common to either group, divided by the total number of such pairs and the E_{MERGE}, $E_{\text{INTERSECT}}$ and E_{TRANSFER} error functions in section 3.3.5 are multiplied by the inter-group affinity subtracted from 1.

We have derived the constants above from empirical feedback; our experience indicates that while our grouping algorithm is fairly general, it is useful to tune it to specific domains for better performance. Our algorithmic framework allows such tuning.

3.9 Conclusion

Unlike most other social network analysis algorithms that detect groups from global network data, our algorithm helps individuals automatically identify and use their social groups by analyzing their online social actions. Our algorithm is targetted for real-life, ego-centric social contexts where individuals may play multiple roles.

We formulated the social topology extraction problem as the compression of a natural social topology, where initial groups are labeled with their significance value, to a desired size according to a metric function that biases the composition of desired groups. We proposed a simple greedy algorithm derived from this value metric. Our algorithm can be used to produce the best representation of a social topology for a given size budget, though it can also automatically determine the number of significant groups a user has. Based on our public released applications that incorporate this algorithm, it appears that the results are good enough to be interesting to many users.

We have performed an analysis of our algorithm over approximately 2,000 email archives and 1,100 photo collections, the latter collected by our Facebook application.

We show that our algorithm is significantly different from the popular Newman's clustering algorithm for community detection. Using edit distances as an information-theoretic metric, we see that even a tiny topology consisting of 4 groups for email and 3 groups for Facebook produces significantly smaller edit distance ratios than Newman's algorithm.

We found that both the email and photo corpus are amenable to compression, allowing our algorithm to produce social topologies that capture much of the value in the input set with a small percentage of groups. We show that the algorithm can capture 80% of the value with 20% and 42% of the groups for email and photos, respectively. Interesting, we also found that there are less than 15 significant groups in our email communications and 7 groups in photos for half of the population in our data sets. These results offer insight into people's social relationships as captured by their online activities.

In the next chapter, we will return to email archives and consider applications of Muse to browse archives of historical importance.

Chapter 4

Historical Research With Email Archives

As historians know well, letters and documents belonging to individuals serve as invaluable tools of record and provide important insight into the past[1]. In the digital age, it is increasingly apparent that history is being created and captured in digital archives. Therefore, archival organizations frequently make it a point to capture "born digital" materials when acquiring the "papers" of eminent individuals. The digital archives of writer Salman Rushdie at Emory University and those of the poet Wendy Cope at the British Library are just two of many well-known examples [140, 150].

Among digital materials, email is perhaps the most significant. A project to assess the relative value of digital archival materials ranked it the highest among images, speeches, press releases, personal websites and weblogs, presentations, and other artifacts [131]. Prom makes a compelling case for preserving email and surveys many of the associated socio-technical issues [104]. The detailed record embedded in email provides access to the donor's thoughts and actions at a level that has rarely been available in the past and enables researchers to probe questions like: What was the process the donor used to come up with a particular breakthrough? What were they reading at the time and how may it have influenced them? [150] Further, these archives are being accumulated not just by famous people; email reaches just about

[1] See lettersofnote.com for some fascinating examples.

every section of wired societies. Indeed, the British Library has collected sample email messages from ordinary Britons as a way of capturing a sense of life in the 21st century [155].

In this chapter, we discuss considerations in processing personal email archives in the settings of archival organizations and making them available to researchers and to the general public. Much of this work was performed in collaboration with Stanford University Libraries and tested with the email archives of two individuals – the poet Robert Creeley and the computer scientist Richard Fikes – that are hosted in the library's special collections. Our solutions are built as extensions to Muse, meaning that all the features described in the previous chapter are accessible in this setting as well.

Email archives have also become valuable sources of public information. For example, journalists routinely acquire email archives via Freedom of Information requests or from their other sources. The email archives of Gov. Sarah Palin and U.S. Supreme Court justice Elena Kagan are recent, prominent examples. The Archivist of the United States, David S. Ferriero, reports that emails have been collected from every U.S. administration since the 1980s, and that the archives in the George W. Bush presidential library include about 210 million email messages [28].

4.1 Related Work

Several projects in the archives community have already recognized the importance of email archives for historical research and are actively working on defining best processes to deal with them [4, 104, 131]. Zalinger provides an engaging account of trying to identify narratives from the email archives of Ben Shneiderman, a professor at the University of Maryland [152]. In the journalism world, projects like Overview and DocumentCloud are popular for processing text corpora; however, they are not focused specifically on email archives. Newspapers have attempted to use crowdsourcing research into newsworthy email archives, for example, with the Sarah Palin emails [48, 92].

4.2 Archiving Process

In this section, we briefly summarize some of the considerations in long-term preservation and archiving for the benefit of readers who may be unfamiliar with this topic.

4.2.1 Stakeholders

There are four main stakeholders in the process of acquisition and use of email archives: the donor of the archive, the curator who actively seeks out collections and works with donors to acquire them, the archivist who processes the collection, and the researcher who uses it. (Sometimes, the creator of the archive may not be the same as the donor; if the creator is deceased, the donor is often a close family member or an estate agent.) Each of these stakeholders has different requirements and expertise.

Donors are sometimes hesitant to turn over their email archives to curators as they may contain deeply personal information such as family or financial records, confidential information about other people, and health matters. Donors are often busy people and may not have the time to perform a detailed assessment of their archives. (The British poet Wendy Cope reportedly spent over 2 years screening her email archives before turning them over to the British Museum [150].) Curators have to ensure that donors' interests are properly protected. A bad relationship with one donor may affect a curator's reputation and thus his/her ability to solicit future donation. Curators are often subject matter experts and may sometimes be researchers themselves. Archivists are responsible for processing the archives. Frequently, archival organizations have years of backlog in terms of material that has been collected but is yet to be processed by an archivist. While archivists aim to provide broad access to the archives and encourage exploratory use, they also have to make sure that access to materials are properly managed according to privacy considerations, copyrights, embargoes established by the donor, and other legal restrictions.

Researchers like doctoral students, authors, journalists, etc. are the end-users of the archive. Often, they first like to gain a sense of the content in the archive, e.g., whether certain people or subjects are mentioned in the archive, before investing the time and expense of making a trip to the archive's reading room, or raising funding for a project. To give potential researchers a sense of what material is an archive, archivists manually create finding aids, which are similar to tables of content. The traditional finding aids for a fully processed collection with letters typically lists the names of correspondents along with the date range of the correspondence.

4.2.2 Comparing Email with Traditional Correspondence

To illustrate the challenges in processing email archives using the traditional approach, consider the 7,000 letters in the paper component of the Robert Creeley archive. The correspondence listing in the finding aids for this archive takes 122 pages out of a total of 251 pages (when printed), indicating the importance of letters. Note that this listing had to be painstakingly and manually generated by an archivist. In contrast, Creeley's email corpus consists of over 80,000 pieces of email, spanning about 13 years. The messages are loosely organized with relatively little folder structure, with many duplicates; after de-duplication, the number of messages drops to 28,650. In the Richard Fikes collection, there are about 108,000 messages (99,140 unique), mostly spanning a period of about 15 years. The scale of these archives makes it extremely difficult for the archivist to process each message manually, or for the researcher to examine them individually.

However, email archives have several benefits: they can be digitally searched and form a detailed and consistent record over a long period of time that provides a wonderful window into the thinking of the donor [131]. Another advantage is that copies of messages often exist with both the sender and the receiver, unlike paper letters where it is relatively difficult to get a copy of letters sent by the donor. This allows chains of conversation to be reconstructed easily. Emails also capture group conversations between multiple people or on mailing lists, and frequently include supplemental images and documents in the form of attachments. While letters tend to be relatively formal, the language used in email is frequently informal, colloquial, and uses its own abbreviations and emoticons.

There are also differences due to media: physical letters and documents carry useful attributes such as signs of wear or tampering, corrections, margin notes, and even the smell of age. History researchers report feeling motivated to work on projects by the feeling of direct connection to the donor obtained by touching their physical materials. (Indeed, this author has felt the same when going through family diaries from over a century ago.) This sense of connection is harder to replicate in the online world.

However, digital media do have the advantage that they can be easily duplicated at reasonable effort and cost and may be easier to preserve over a long period of time with careful archival practices.

4.3 Preparing Email Archives

Our solutions for processing email archives in special collections are implemented on top of Muse. We continue to use all four types of cues in Muse: communication patterns with automatically inferred groups, automatically generated monthly summaries, identification of different types of sentimental messages, and a zoomable and draggable 2.5D attachment wall for image attachments.

While these features are useful for a researcher who has full access to the archive, a different set of features is needed for an archivist preparing the archive, and for users who might be interested in getting some sense of the archive's contents but may not be authorized for full access. To support different use cases, we repackage Muse into three modes: public mode, archivist mode and reading-room mode, further described below.

4.3.1 Data Ingest and Cleaning

Long-term email archives span a variety of technologies, media and formats. To ingest email in various email formats, we use commercial tools such as Mailstore Home or Emailchemy, which are well known tools in the world of archives. These tools can read a variety of email formats and convert between them. We use these tools to first convert all input files to the open mbox format that Muse can read. Muse can also fetch data from email servers using the IMAP or POP protocols if needed.

A common problem is that personal archives are frequently acquired over multiple rounds of accession spanning many years. This frequently leads to duplication and changes in folder structure. As mentioned earlier, Muse detects and ignores duplicates, and performs entity resolution to merge identities of people who may have multiple email addresses or name spellings. We also see cases when some metadata like message recipients or date stamps are missing due to format changes, discrepancies between tools or data corruption. Muse attempts to deal with these problems as gracefully as it can, by providing reasonable defaults if the data is missing or obviously incorrect.

4.3.2 Archivist Mode

Though Muse may attempt to auto-correct some problems, it is useful for an archivist to look over these corrections and investigate any anomalies in the data set. In the archivist mode, we provide a data quality report for all corrections made by Muse. A feature of the data quality report that we have found particularly useful is to report messages with exceptionally large text size in their bodies, which are usually anomalies related to bad data formatting. Common reasons for such bad data are errors by email clients that treat “uuencoded” attachments as being part of the message body, or bad headers that make it seem like a whole series of messages is contained in a single message. An archivist can also review all attachments by file extension, to ensure that all file types are supported by an external program or an optional external viewer, described below.

A second important feature is the ability for archivists to screen messages for sensitive information that may need to be redacted. We provide a regular expression search feature for identifying certain types of sensitive information which have specific formats or patterns such as social security or credit card numbers. Archivists can also search terms that may reflect other confidential data such as financial matters or health and student records. They manually examines these messages and can tag the ones that are genuinely sensitive. Archivists can also use the browsing tools described below to get an overall sense of the contents of the archive. When screening is complete, they can export a version of the archive with the sensitive messages removed. They can also export a separate version of the archive for use in public mode.

We note that the archivist mode can also be of use to a donor. During the user study described in section 2.5, one of the archivist participants made the following comment: *"Many of [the Stanford faculty members] are reluctant to donate email . . . without sorting through it first, since email very often contains sensitive content that is not necessarily relevant to their careers. Sorting through years or decades' worth of email can be an overwhelming task, but a tool such a Muse could make it a) more fun, and b) easier to identify material that should not be donated to the archives."*

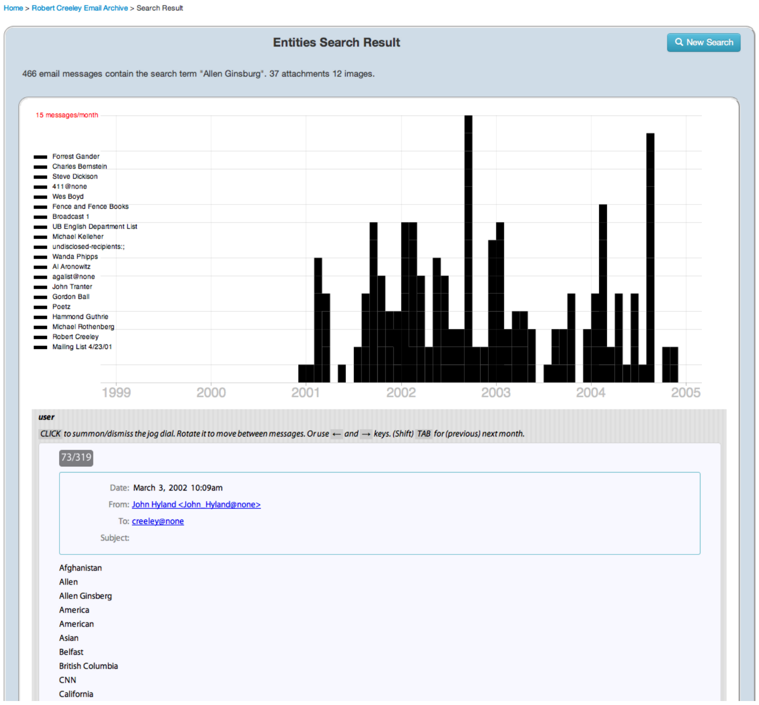
We envision that in the future, tools like Muse will make it easy for donors and archivists to process email archives. This will make it easy for universities or other archival organizations to capture history, instead of being selective about which archives to acquire and process.

4.3.3 Public Mode

The public mode allows anyone on the web partial access to the archive. The use of a browser-based user interface in Muse made it relatively easy to break up the application so that the server component could be run on the archive servers, while users can access the archive through a regular web browser. There are two main concerns when implementing the public mode: confidentiality and scalability.

Confidentiality

For archives whose contents are completely public, confidentiality is not a concern and we can provide full access over the web. This scenario occurs occasionally with the archives of public officials obtained through legal means, or voluntarily made public by a donor. However, most personal archives cannot be made entirely public due to the sensitivity of email messages, issues about copyrights, etc. Our main goal is to



Entities Search Result

466 email messages contain the search term "Allen Ginsburg". 37 attachments 12 images.

CLICK to summon/dismiss the jog dial. Rotate it to move between messages. Or use ← and → keys. (Shift) TAB for (previous) next month.

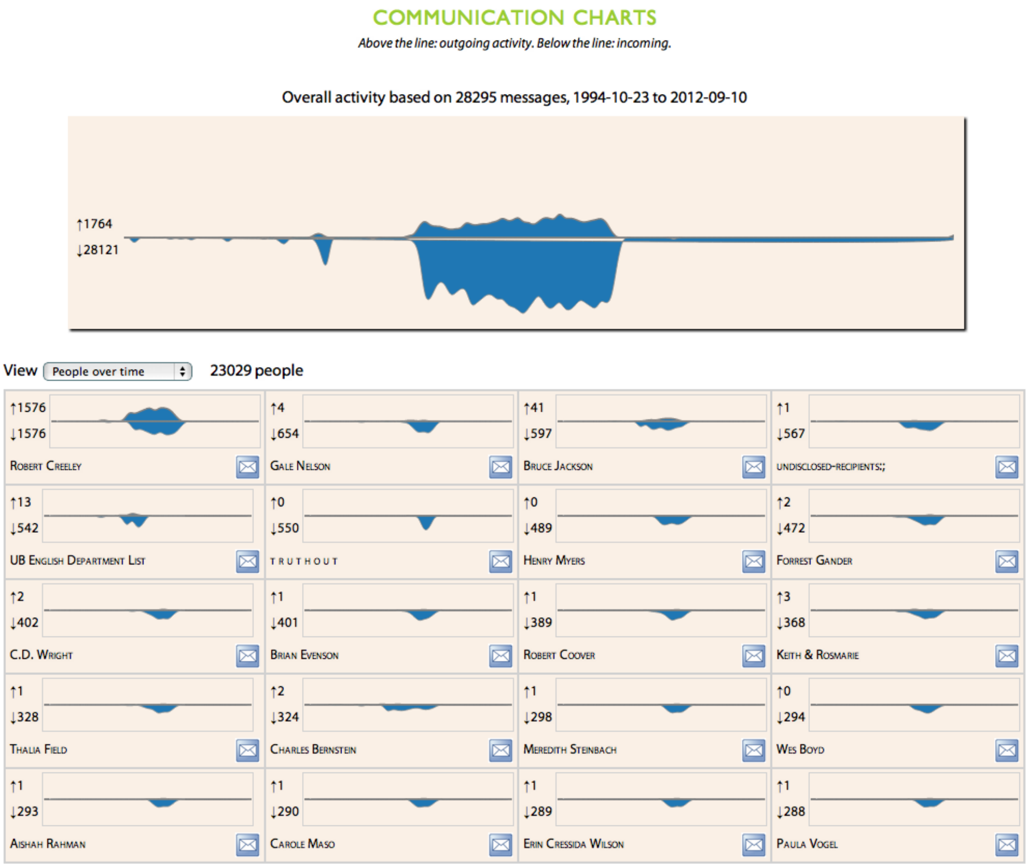
Date: March 3, 2002 10:09am
From: John Hyland <John_Hyland@none>
To: creeley@none
Subject:

Afghanistan
Allen
Allen Ginsberg
America
American
Asian
Belfast
British Columbia
CNN
California

Figure 4.1: A screen from the "public mode" for the Robert Creeley archive. Results of a search for the term "Allen Ginsberg" are shown. The graph shows the number of messages matching the search term across time. The message view below shows the message metadata and partial contents (only names).

allow partial (and of course, read-only) access to the archive to the general public over the web.

How can we let researchers look for leads into the archive and find the presence of possibly interesting information without divulging sensitive details? We use a simple observation: the traditional finding aids in fully processed collections list the names of all correspondents and topics. Hence, when preparing an archive for the public mode, we extract all the correspondent names as well as entity names from subject lines and message bodies, using the Stanford Named Entity Recognition (NER) tools [125]. These entity names are indexed as if they were the only content in the message and the archive is stored in a public MUSE server. The public mode lets anyone search for names in the archive (see Fig. 4.1), browse messages with their original timestamps, view names of senders and receivers (exact email addresses are never



COMMUNICATION CHARTS

Above the line: outgoing activity. Below the line: incoming.

Overall activity based on 28295 messages, 1994-10-23 to 2012-09-10

View: People over time

23029 people

↑1576 ↓1576 Robert Creeley	↑4 ↓654 Gale Nelson	↑41 ↓597 Bruce Jackson	↑1 ↓567 undisclosed-recipients:;
↑13 ↓542 UB English Department List	↑0 ↓550 TRUTHOUT	↑0 ↓489 Henry Myers	↑2 ↓472 Forrest Gander
↑2 ↓402 C.D. Wright	↑1 ↓401 Brian Evenson	↑1 ↓389 Robert Coover	↑3 ↓368 Keith & Rosmarie
↑1 ↓328 Thalia Field	↑2 ↓324 Charles Bernstein	↑1 ↓298 Meredith Steinbach	↑0 ↓294 Wes Boyd
↑1 ↓293 Aishah Rahman	↑1 ↓290 Carole Maso	↑1 ↓289 Erin Cressida Wilson	↑1 ↓288 Paula Vogel

Figure 4.2: Top correspondents in the Robert Creeley archive and their activity over time.

exposed) and to view all the named entities in the message, listed alphabetically. However, the full body of messages is not shown in order to preserve confidentiality. In our experience, this is a good compromise between making the archive useful to the public and protecting sensitive information. (We have not encountered a case where even the names are too sensitive to be shown, but in case this happens, the archivist can redact the entire message from the public view.) Interested researchers can follow up by visiting the archive reading room to get full access to its contents. Users can also apply message filters and see graphs of communication activity with individuals (see Fig. 4.2).

In the public mode, the names of a message's source email folders are not available.

Neither are email attachments; only the number of email attachments is displayed. The sentiment and graph groups are currently not available in public mode, though we may enable these under archivist control in the future.

The public server never contains the original email corpus in raw or indexed form, thus ensuring that sensitive data is not lost even in the extreme event of a server compromise.

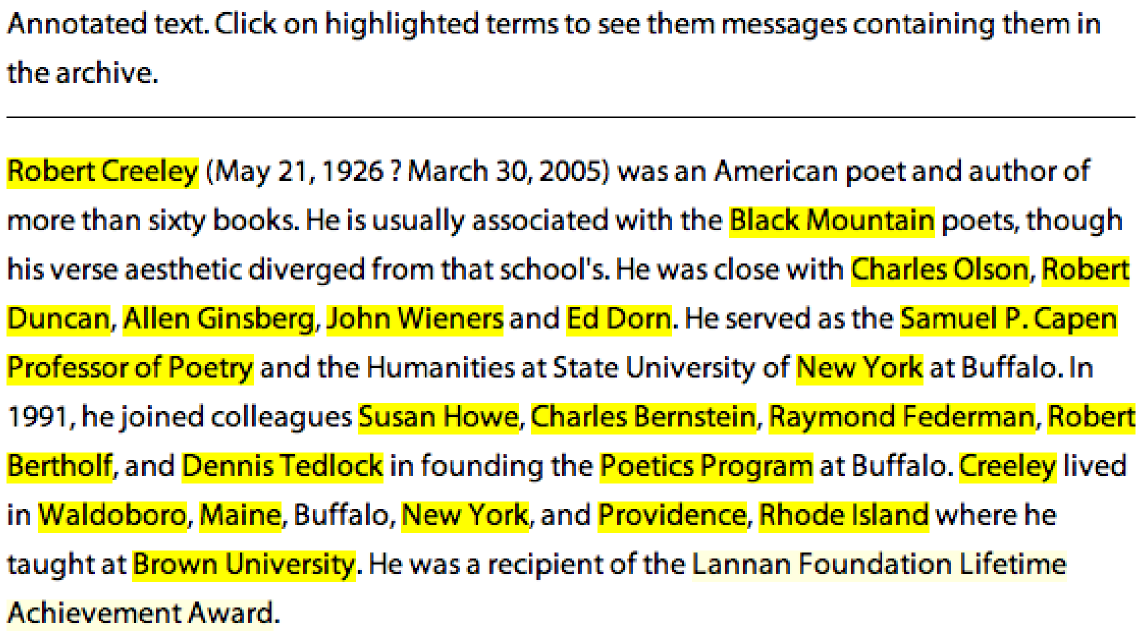
Scalability

Normally, users download and run Muse on their own computers since each user only works with their own personal archive. In an archives setting, however, there may be many concurrent users accessing a single archive over the web. This is especially likely when there is some breaking news related to the archive. An engineering consideration for us was to support concurrent users sharing access to the same archive in public mode, without requiring a linearly scaled up Muse server with a large amount of memory. To contain memory consumption, we re-engineered objects representing the archive to be shared across concurrent HTTP sessions. Thus, each archive is loaded only once into the web server, and only a small amount of per-session state is needed per connected client.

4.3.4 Reading Room Mode

The reading room offers full access to the archive that has been prepared by an archivist. A researcher in the reading room can view the full contents of messages with all attachments, similar to the archivist mode except that redaction and export of messages is not possible.

One problem frequently encountered in long-term email archives is that messages include attachments in legacy formats such as WordPerfect, Wordstar and Lotus 1-2-3, which users cannot open today. To address this problem, we allow attachments to be viewed with an external viewer for legacy file formats that is installed in the reading room environment. (Our reading room currently uses a commercial tool called QuickView Plus that supports a few hundred legacy formats, but users can



Annotated text. Click on highlighted terms to see them messages containing them in the archive.

Robert Creeley (May 21, 1926 ? March 30, 2005) was an American poet and author of more than sixty books. He is usually associated with the Black Mountain poets, though his verse aesthetic diverged from that school's. He was close with Charles Olson, Robert Duncan, Allen Ginsberg, John Wieners and Ed Dorn. He served as the Samuel P. Capen Professor of Poetry and the Humanities at State University of New York at Buffalo. In 1991, he joined colleagues Susan Howe, Charles Bernstein, Raymond Federman, Robert Bertholf, and Dennis Tedlock in founding the Poetics Program at Buffalo. Creeley lived in Waldoboro, Maine, Buffalo, New York, and Providence, Rhode Island where he taught at Brown University. He was a recipient of the Lannan Foundation Lifetime Achievement Award.

Figure 4.3: An excerpt from Robert Creeley's Wikipedia page, annotated with the results of a bulk search of the names on his archive.

plug in whatever program they prefer.) When browsing messages, users can click on attachment names to open it optionally in this legacy file viewer.

4.3.5 Bulk Searches

In the public archives setting, a major difference from the original purpose of Muse is that the user may not be already familiar with the contents of the archive. However, a few features of Muse are useful to orient users, such as the summaries of key named entities per month and graphs of communication activity with different individuals.

While users can search the archive with Muse (whether full-text in reading room mode or just the names in public mode), they may not know the terms to search for in an unfamiliar archive. One way users might look for interesting messages in the Robert Creeley archive might be to go to a biography of Creeley (such as the one on Wikipedia), look for the terms with which he is most prominently associated, and search for them in the archive. We essentially automate this process with a novel idea

– we let the user paste in arbitrary text in a search box, extract all the named entities within this text and look them up in the archive. We then reflect the original text to the user, highlighting terms that had some matches in the archive. In this way, the user can quickly see connections between the pasted text and the archive. Clicking on a highlighted term leads to the set of messages in the archive that contain the term. This is an efficient way of performing a lot of potentially useful queries at once. Users can also use the experience-infused browser described in the next chapter to achieve the same effect during regular browsing. The idea of the experience-infused browser was, in fact, inspired by this use case.

4.4 Limitations and Future Work

Currently, Muse can smoothly handle personal archives with about 100,000 messages. Some features in Muse make the assumption that the archive belongs to a single person, for example, communication frequency graphs compute the number of incoming and outgoing messages from a single person's perspective. These features may not be applicable in some settings that are not centered around a single person, such as the archive of a mailing list. Our future plans are to improve scalability and to provide cross-collection search so that archive patrons can search multiple collections at once.

4.5 Conclusions

We have shown how long-term email archives can be processed relatively efficiently, and how they can be made partially available to the general public.

As a result of our enhancements, the department of Special Collections at Stanford University, and the Stanford University Archives has agreed to provide a public mode view of Fikes and Creeley collections. Muse is also being tested out by four partners: Columbia University, Oxford University, the New York Public Library and the Smithsonian Institution. Our experience with the Creeley and Fikes corpora and the resulting system should be useful to other people who need to process large-scale email archives. Our system is currently available at the test URL: http://sulmuse-dev.stanford.edu/muse/archives.

Currently, the utility of email archives for historical research is limited by the cost of acquisition, processing and delivery. Donors are hesitant to donate their archives because of a lack of tools to review and filter their archives. We hope tools such as MUSE will increase the collection and use of email archives for the historical record. For example, it should be routine for universities to capture and preserve the archives of eminent professors.

Chapter 5

Experience-Infused Browsing

So far, we have discussed ways in which personal archives can be used by users to revive memories or to consult the historical record. We will now turn our attention to implicit uses of personal archives and illustrate the concept of experience-infused software. We will show how it is possible to create new user experiences in web browsing and web search using personal archives. This chapter describes a system and studies related to web browsing, and the next chapter will discuss web search.

We encounter an overwhelming amount of information through web browsers today through crowded news portals, social networking feeds, shopping sites, blogs, documents, etc. In response, we frequently resort to rapid skimming and selective reading of web pages. When skimming a piece of text, whether online or offline, terms we know jump out at us and grab our attention, based on factors such as our memories, interests and affiliations. Different parts of the same content can attract different users.

As we have seen so far, personal archives capture a significant portion of life experiences for users in the digital world. This has led to the promise of "Total Recall" [13, 42] – the powerful idea that people can recall with great precision everything they have encountered in the past. Can we use this recall to help us process information more effectively as we browse the web?

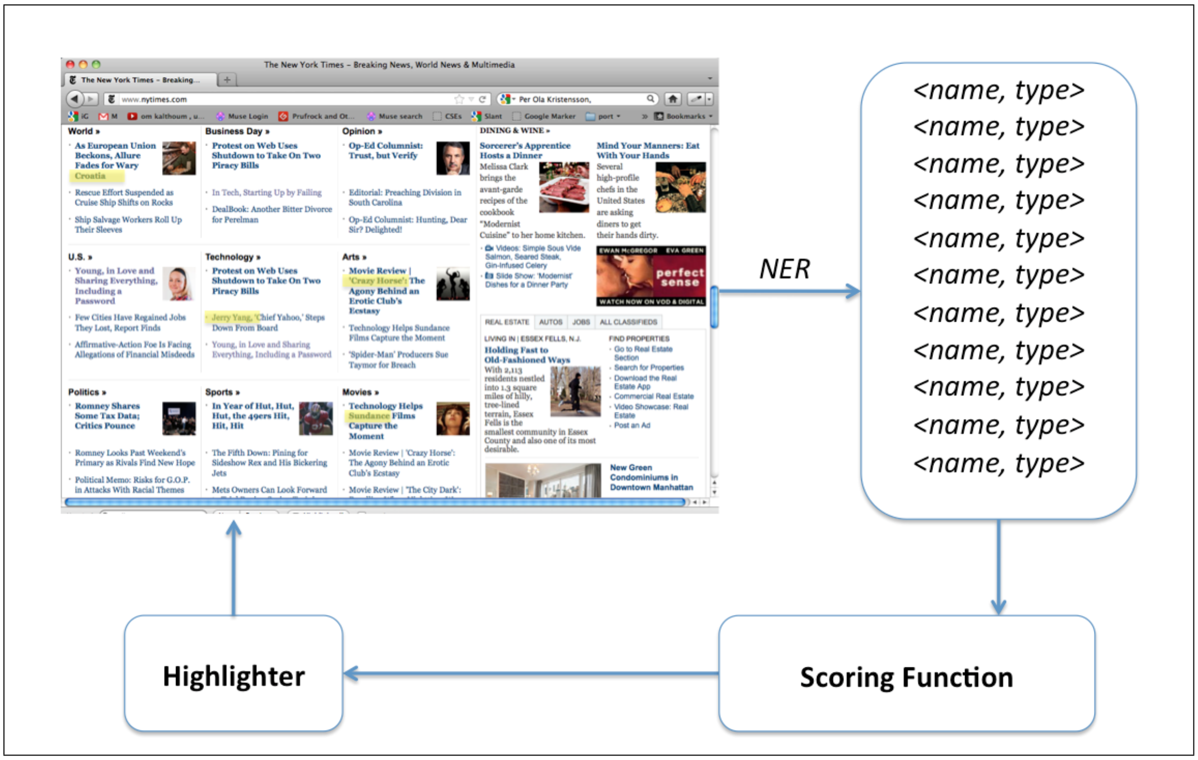


Figure 5.1: The experience-infused browsing flow. The Muse program is running in the background and has indexed the user's email archive and chat logs, and creates an index. When the user loads a page in the browser, the browser extension extracts names from the text on the page and sends them to Muse which looks them up in the archive and scores the results based on frequency and type. The browser extension updates the page DOM to highlight terms scoring above a threshold and inserts hyperlinks from each term to a Muse message view consisting of all messages with that term.

Infusing a Browser with the User's Experience

The basic idea behind the experience-infused browser is simple (see Fig. 5.1): it annotates every web page, effectively personalizing the page for the user without any explicit action on her part. It brings the user's attention to possible terms of interest on a page by highlighting them on the page, making their background yellow, as if someone intimately familiar with the user had prepared the page for her and marked it up with a highlighter. The terms to be highlighted are selected based on their presence in the personal archive, which has been indexed by Muse, as described in earlier chapters. This allows the user to notice these terms easily as she skims and scrolls across the page. Each highlighted term is hyperlinked back to a message view in Muse, making it easy to reach messages containing it. This feature is essential because no human memory is perfect, and the user may sometimes not recognize a name and wonder why it is highlighted.

In addition, the browser inserts a small call-out at the bottom of the page listing the highlighted terms. This feature is especially useful on long web pages spanning several screens, where a user may not even want to scroll all the way across the page. Thus we can enhance the effect of a user noticing personally relevant terms on a page, and scale it to long pieces of content that are difficult to skim manually.

Our current system is limited to highlighting terms that a user has already seen in the past; it does not suggest new or related content. Our goal is to avoid imposing a high cognitive overload on the user, since she may not always be receptive to annotation, or the algorithmically generated highlights may not be useful. Highlighting terms in-place makes it relatively easy for the user to ignore the annotation and interact with the page normally.

The primary components of our system are a Firefox or Chrome browser plugin and a specialized version of Muse that indexes the user's email messages and instant messaging logs, if available, and exports a lookup service over this data. Both these components run entirely on the user's own computer, thus providing the benefits of privacy-preserving personalization. Apart from privacy, this model of personalization gains the other benefits of experience-infused applications – the ability to potentially bring in different sources of data into the personal archive, and the ability to have the archive apply instantly to any site viewed in the browser.

Contributions

The novel contributions of this chapter include the following:

1. We propose the concept of an experience-infused browser that brings a user's entire digital archives to bear on the task of web browsing. The browser personalizes web pages by highlighting terms of potential interest to the user. Our approach honors users' privacy, yet provides the benefits of personalization.

2. We have publicly released a prototype of the experience-infused browser, and report design and implementation trade-offs discovered while building this prototype.

3. Results from a study of 9 users suggest that our system achieves these goals effectively. 7 out of the 9 users indicated that they were interested in using the system beyond the duration of the study. While our study is relatively small, we provide qualitative examples to show that a simple engine, infused with a massive amount of personal information, can be useful in many different ways.

5.1 Related Work

Most research in the area of life-logging, archives and recall focuses on the capture, preservation and explicit use of personal data. Our research goal is to investigate scenarios and techniques to make implicit use of this data for the task of web browsing.

5.1.1 Web Browsing

Rhodes's Margin Notes [113] takes the paragraph currently being viewed on a web page and tries to find documents or messages in an archive that have similar words in them. Margin Notes aims to supplement the linear reading of a document and suggest related material for each paragraph that a user may have missed. In contrast, our context is one of skimming rich websites that agglomerate all kinds of names, events and stories on a single page, where a personalized user experience is useful in selecting parts of the content. Therefore, our primary motivation is in some sense almost the opposite of that of Margin Notes.

These factors lead to different design decisions in the relevance matching algorithm and user interfaces. As we will describe later, the experience-infused browser looks for named entities instead of all words, which makes our analysis less sensitive to the language used in the text. Margin Notes does not annotate text in-place; it lists related content for each paragraph in a sidebar. Our implementation runs on the end user's computer to ensure privacy of their personal archives, while Margin Notes used a proxy web server implementation.

There are a variety of systems that attempt to display pages similar in content to the page a user is browsing. The Google related toolbar plugin for the Chrome browser, a product that has now been retired, is one example. Lieberman and others explore the idea of "zero-input" interfaces in the context of reconnaissance agents that learn about a user's interests by watching web browsing patterns, and then suggesting pages that are likely to be most relevant to the user [70]. In a series of papers, Teevan, Adar and others studied web revisitation patterns and built Diff-IE, a plugin for Internet Explorer that focuses attention on the parts of a page that have changed since the user's previous visit [2, 134, 133]. Tools like SparTag.us [56] let users manually highlight text snippets and allow them to refer to these tags easily. Semantic web browsers like Magpie use ontologies to find related concepts and point the user to such content [32]. Unlike our browser, none of these systems exploit users' personal archives, but some of the techniques could be used in synergy with our browser in the future.

5.1.2 Web Personalization

Many web services attempt to personalize content for a user. While the mechanisms used for personalization are generally opaque to users, these services need to either build up a profile of users either implicitly based on their interactions with the service or ask users to enter information explicitly. The former approach suffers from a cold-start problem, and the latter imposes a burden on the user.

Gauch et al. survey techniques for generating user profiles [41] in a broader collection of articles related to the adaptive web [19]. Liu et al. studied Google News logs and proposed a Bayesian framework for personalized news recommendations using the users inferred interests and local news trends [73]).

5.1.3 Email Enhancements

Systems like Xobni and Rapportive bring in specific content from the web (like a public profile of the message sender) to supplement messages arriving in the inbox; our system can be viewed as an inversion of this model, where email archives are used to illuminate web pages instead.

Mesarina et al. built Sidebar, a browser plugin to show emails containing the URL of the current page to the user [85]. However this is limited by the fact that, more often than not, an email message may not contain the exact URL of the page being visited. In contrast, our technique matches terms on a web page with the actual contents of the user's email archive and highlights terms that may potentially interest the user. We use standard natural language processing and information retrieval techniques to achieve this goal.

5.1.4 Serendipity

The Oxford English Dictionary defines serendipity as: "The faculty of making happy and unexpected discoveries by accident." Serendipitous discoveries clearly entertain and captivate users, but Gritton also discusses whether they might contribute to learning [47]. Lawley and Tompkins provide a useful breakdown of serendipity into different types and phases, and attempt to identify precipitating conditions for each [68]. It is acknowledged that serendipity is hard to define, let alone engineer, and therefore somewhat hard to study formally [5]. Beale presents techniques to model serendipity and interest based on pages recently viewed by the user; the author uses this prior information to highlight interesting links on the page using special coloring mechanisms [12]. Our experience-browser provides a possible trigger for serendipity by identifying terms on web pages that the user may not have been expecting, or by illuminating interesting personal connections to those terms.

5.2 System Design

Our system consists of two parts – a background service in Muse that accesses and indexes the user's personal history from email files or online email accounts, and a Firefox or Chrome browser plugin that performs user interface tasks. We chose this separation because the indexing component needs to be highly performant due to the size of personal archives, often running into tens of thousands of messages and spanning hundreds of megabytes of text. The background service is implemented in Java. The front-end part is implemented in Javascript, and is kept relatively lightweight so it can be ported to other browsers in the future. Fig. 5.1 illustrates the components of the system and the flow when a typical web page is loaded.

The background service in Muse accesses and indexes email message contents. Instant messaging (chat) archives can also be indexed if available through an IMAP interface such as that of Gmail. The indexer runs in the background on the user's own computer. It has to run only once, and for most users, indexing is a one-time operation that takes only a few minutes. The index is serialized and saved to disk, so that it can be quickly loaded after a computer or browser restart.

The front-end code is implemented in about 1,200 lines of Javascript. After the web page is completely loaded by the browser, the script extracts page text and posts it to Muse that is running on the same machine in the background. We extended Muse by enabling it to look up a posted a web page in its index. The results of the lookup are used to inject Javascript into the page to highlight the relevant terms. For most web pages, the lookup completes in two to three seconds, a perceptible delay, but shorter than the time most users spend on a page. In any case, users can begin to interact with the page as soon as it is loaded, without waiting for the highlights to appear. Performing annotations within the browser has the advantage that it can seamlessly handle encrypted HTTPS pages or pages available only after login.

5.2.1 Identifying Named Entities

With some early experimentation, we realized that blindly matching words or phrases on the page with terms in the archive led to a lot of noise, often inundating users with terms of little or no interest. A key heuristic we used based on our prior experience with identifying memorable terms in MUSE was to focus on named entities. This is a useful heuristic – while names are a relatively small fraction of the text on the page, they capture a large number of significant and memorable associations compared to ordinary words. Therefore, the use of named entities reduces the number of hits on a page to a manageable number, improving precision and reducing noise, while still allowing a relatively high degree of recall.

The background service in MUSE extracts named entities from the page text using the Stanford NLP toolkit [125]. We have to format contents appropriately in preparation for the recognizer, since it depends on signals such as capitalization, nearby words and whether the term appears at the start of sentence. Therefore, we traverse all HTML elements in the web page, extract text from them, and concatenate them to form the page text. As a special case, we insert full stops if needed after the text extracted from division elements (<div>), list elements (<li>), paragraphs (<p>) and headers (<h1>, . . . , <h6>) since it is unlikely that normal sentences would span these elements.

5.2.2 Ranking and Filtering Hits

To determine the importance of each term that hits in the personal archive, we assign it a score. Through experimentation, we observed that users almost always find people names more interesting than other named entities (such as names of places and countries, which can be fairly generic). We use the fact that our named entity recognizer can classify names as likely to belong to a person, place or organization to bias the scoring based on entity type. The scoring formula for a term t on the page is:

$$score(t) = w(Type_t) \times |\{d|t \in d, d \in D\}|$$

where w is the weight assigned to an entity type and D is the document archive. The score of a term t is the number of documents in the archive that contain t (whether in its headers or in its body), biased by the weight for its type. Our default weight is 1,000 for the person type, and 1 for all other types, strongly boosting the score of person names. We expect this scoring function may become more sophisticated in future implementations.

5.2.3 Highlighting Hits

Since not all terms may be of equal interest to the user, our current implementation uses two levels of highlights, strong and weak (see Fig. 5.2). Strong highlighted terms have a strong yellow background, making the term prominently visible on the page, allowing them to catch the user's eye when displayed on the screen. Weakly highlighted terms have a light yellow background, making the term less prominent, and noticeable only if the user is viewing the general area of the page. We determined experimentally that this design strikes a balance between remaining noticeable and not causing information overload. More sophisticated policies of highlighting (including using more colors or more shades of highlighting) are possible in future work, but will have to balance information richness with higher cognitive overload for the user.

To decide whether to highlight the term strongly or weakly, we check whether its score is above or below a threshold (the default threshold is 5). The translucent call-out at the bottom displays terms in order of decreasing score, prioritizing the most important terms. This ranking is useful because it may not be possible to display all the highlighted terms in the call-out; the call-out size is limited by default to 3 lines to avoid significant obtrusion, and moreover can be set by the user according to her taste. An additional filtering step we found necessary was due to false hits generated by common single-word names (e.g., names like *James* and *Mary*). We eliminate single-word names that belong to the list of each of the 1000 most common (English language) last names, male first names and female first names.

5.2.4 User Controls

We provide two important controls for the user to control highlights. The first is the ability to quickly filter the set of messages in the archive that are checked with respect to the page. Users can apply filters by search term, by date, by associated groups or sentiments (which are automatically derived, as explained in Chapter 2), by person, by the name of the original folder that contained the message, or by message direction (sent or received). These filters are useful because for some pages, users may want to enable only parts of their archive that are related to the current context and the page. The filters are especially useful when a user has a huge archive which may result in incidental hits for many terms.

Second, users can provide feedback to the browser about which terms are relevant to them and which are not. This is provided by means of a vote up/down button which appears when hovering on a term in the callout (See Fig. 5.2, bottom right). The voting affects a multiplier which is applied to the score for each term. The default multiplier is 1.0; when a term is voted up or down, its multiplier is increased or decreased by 1 respectively. This control is commonly used to vote out terms that the user things are too broad or irrelevant for her, even if they appear in the archive.

5.3 User Studies

We designed our browsing system by conducting formative studies with a few lead users and by using it ourselves for routine browsing for a few weeks. When our design was reasonably stable, we conducted a formal round of studies by inviting 9 users (3 female) to test out our system. The users were in the age group 23-29.

5.3.1 Methodology

We deployed the system on the participants' own machines to assure them about privacy. We asked the participants to download Muse and run it on email folders of their choice, which were typically sent message folders and other folders that contained messages interesting or important to them. Gmail users also included their instant messaging logs, which are accessible over IMAP as messages in a special folder with one message per chat conversation.

Our users had between 2,600 and 21,835 emails and chat conversations. Participants were asked to install our browser extension. They were given a tour of the features of the system and asked to browse normally for an hour and report any interesting terms that the browser highlighted for them that they otherwise might have missed. They were also told that they could vote the terms up or down, but few put in the effort to train the browser, perhaps because they thought they were only using the browser for a limited amount of time in the context of this study. After an hour, we collected their browsing logs and gave them a questionnaire asking for their opinion about various aspects of the browser.

We chose this setup to give our experiment broad ecological validity, and because we wanted to learn about the utility of the system on a diverse range of web pages. While the fact that users were aware that their browsing was being logged may have slightly altered their normal browsing behavior, we do not think these changes were significant enough to change our conclusions.

5.3.2 Results

The overall results of this user study were very encouraging. When asked if they would want to continue to use the browser beyond the study period, 7 of 9 users replied in the affirmative. One participant replied "not sure" whereas another wanted to keep using the system "but only on search and news websites".

The primary goal with this user study was to obtain qualitative user feedback and uncover scenarios in which the experience-infused browser is useful. We therefore asked users for their detailed impressions of the system, as well as specific examples that did or did not perform well for them. We present and categorize below some examples that are representative of their experiences. Users in the controlled study are referred to as P1 to P9, though we have also included feedback from other users who have informally used the system outside the context of this study.

People Names

A common theme was for users to discover the names of someone they either knew directly or whose name they had encountered in the body of an email message. A Stanford student, while scanning an op-ed article about online education in a news site, found that it mentioned John Hennessy, the president of his university. John's name was mentioned in the bottom half of the article ("below the fold"), so the user would have missed it had it not been presented in the browser call-out. In this instance, the student had not directly exchanged email messages with John, but his name was present in email conversations. This incident illustrates that it is useful to go beyond just the names of correspondents in an email archive to the actual message contents.

P4 was looking at a company website and found a person's name highlighted by the browser which he did not recognize. On exploring further, he discovered that the person had given a talk some years ago which he had attended. He remarked, *"I would like to have such a tool present everywhere which helps me reach for such hidden information, which I have forgotten about."*

One user who is a professor was attending a conference and browsing its program web page, when she found highlighted several occurrences of a author name which was not familiar to her at all. By examining the relevant email messages, she discovered that the person had applied for a faculty position at her university several years earlier and was turned down for an interview despite her recommendation. The university should have considered the candidate after all! This story demonstrates that our browser can be effective in illuminating the serendipitous re-crossing of paths.

P2 was browsing a website for applying to a foreign internship and found a testimonial on the website by a person he knew and responded *"I would not have noticed this name had it not been highlighted."*

P9 was on a page with a long list of startup companies that had recently been incorporated, along with the names of their founders, when she was surprised to find highlighted the name of a former customer of hers who had interacted with her 3 years earlier.

Yet another user found that the name of the teaching assistant for a course that he was considering was highlighted. Examining the related messages reminded him that the person had emailed him in a completely unrelated context – as a response to a campus classified advertisement.

The above examples show that highlighting previously encountered person names, along with the messages in which they were encountered, can be useful in a variety of contexts: personal or professional, either directly known or merely discussed, and well known to the user or forgotten about after a chance encounter.

Products and Commerce

P1 found the browser useful as he was browsing an airline's website and found its loyalty program named *Skywards* highlighted. The user responded *"The airline had emailed me that I have accumulated sufficient miles to get a discount and I had almost forgotten about it."*

Another user found an article on a news site about *Toyota Prius*, the car owned by him. This indicates that personalization based on products owned by a user is possible using names present in ordinary email conversations or in email receipts. Interestingly, since our browser treats all text on a page equally, it sometimes highlighted relevant names in page advertisements as well. This indicates that it may be effective in highlighting relevant ads without compromising user privacy.

Organization and Place Names

One user found that a news blog happened to mention Driptech, the name of the company he had interned at the previous summer, and it helped him notice its presence. Another user from India discovered an article that mentioned Indian cooking in the cooking section of a news site, to which he ordinarily does not pay any attention.

In these examples, users were surprised to find the particular name on a page, and were happy that our browser helped surface information that they otherwise might have missed.

Long Pages and Documents

Fig. 5.2 illustrates a real example of how the browser helps a user zoom into possibly interesting content in a document too long to read entirely. It shows the Facebook, Inc. filing with the Securities and Exchange Commission for its initial public offering. The document has over 180 pages when printed and 652 KB of text, and would normally take a few hours to read entirely. However, for someone with a casual interest

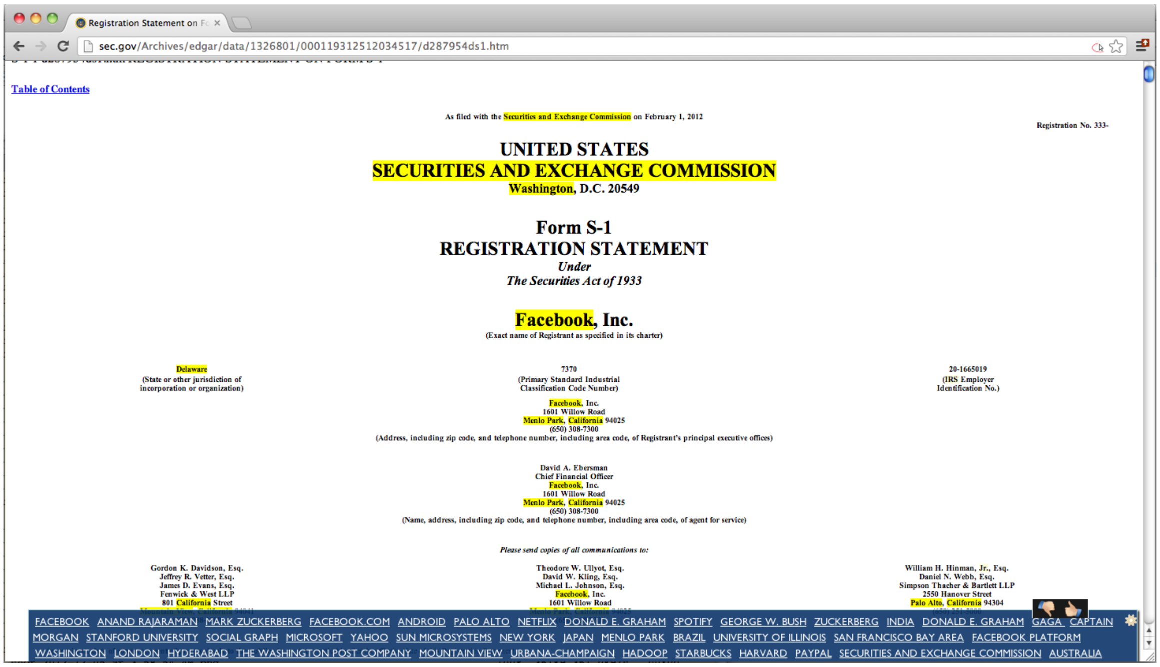

Registration Statement on Fc
sec.gov/Archives/edgar/data/1326801/000119312512034517/d287954ds1.htm

Table of Contents

As filed with the Securities and Exchange Commission on February 1, 2012

Registration No. 333-

UNITED STATES
SECURITIES AND EXCHANGE COMMISSION
Washington, D.C. 20549

Form S-1
REGISTRATION STATEMENT
Under
The Securities Act of 1933

Facebook, Inc.
(Exact name of Registrant as specified in its charter)

Delaware	7370	20-1665019
(State or other jurisdiction of incorporation or organization)	(Primary Standard Industrial Classification Code Number)	(IRS Employer Identification No.)

Facebook, Inc.
1601 Willow Road
Menlo Park, California 94025
(650) 308-7300
(Address, including zip code, and telephone number, including area code, of Registrant's principal executive offices)

David A. Ebersman
Chief Financial Officer
Facebook, Inc.
1601 Willow Road
Menlo Park, California 94025
(650) 308-7300
(Name, address, including zip code, and telephone number, including area code, of agent for service)

Please send copies of all communications to:

Gordon K. Davidson, Esq.
Jeffrey R. Vetter, Esq.
James D. Evans, Esq.
Fenwick & West LLP
801 California Street

Theodore W. Ullyot, Esq.
David W. Kling, Esq.
Michael L. Johnson, Esq.
Facebook, Inc.
1601 Willow Road

William H. Hinman, Jr., Esq.
Daniel N. Webb, Esq.
Simpson Thacher & Bartlett LLP
2550 Hanover Street
Palo Alto, California 94304

FACEBOOK ANAND RAJARAMAN MARK ZUCKERBERG FACEBOOK.COM ANDROID PALO ALTO NETFLIX DONALD E. GRAHAM SPOTIFY GEORGE W. BUSH ZUCKERBERG INDIA DONALD E. GRAHAM GAGA CAPTAIN
MORGAN STANFORD UNIVERSITY SOCIAL GRAPH MICROSOFT YAHOO SUN MICROSYSTEMS NEW YORK JAPAN MENLO PARK BRAZIL UNIVERSITY OF ILLINOIS SAN FRANCISCO BAY AREA FACEBOOK PLATFORM
WASHINGTON LONDON HYDERABAD THE WASHINGTON POST COMPANY MOUNTAIN VIEW URBANA-CHAMPAIGN HADOOP STARBUCKS HARVARD PAYPAL SECURITIES AND EXCHANGE COMMISSION AUSTRALIA

Figure 5.2: Example of highlighting terms in a long, 180-page document – Facebook, Inc.'s SEC S-1 filing. Terms of potential significance to the user are highlighted, and also listed in a callout at the bottom. The user can vote terms up or down by hovering on them in the callout (see bottom right). Filters can be applied to the archive by clicking on the settings icon in the callout.

in such a dense document, the experience-infused browser offers a quick alternative. While the extreme length of this page means that our browser takes about a minute to generate the highlights, this is still much faster than trying to browse the page manually. In this example, the author serendipitously found that a name of one of his former professors was highlighted, a fact about which he was unaware.

This example prompted a journalist to comment that the experience-infused browser may be useful in contexts where reporters need to quickly scan documents for possibly relevant material before a filing deadline. It may also be useful to them when they are researching a story to keep an eye out for news related to their interests. Their interests could be automatically collected from previous articles they have written and imported into the archive.

This author regularly uses the browser on long conference program listings (such as the ACM CHI conference) to illuminate people and topics of potential interest.

General Comments

Participants in the study found our browsing extension an effectively way of viewing a web page through a uniquely personal lens. They found it especially compelling on sites that they usually only skimmed, or that had a lot of content or listings of people names.

P5 said *"I feel like this almost presents me with a personal synopsis of the (web) page."* P8 also responded along the same lines: *"This tool lets me skim through websites faster."*, while P6 said *"I like how it recognizes certain topics that I am interested in–the highlight helps me walk through the site better."*

Users generally liked how the browser helped them find interesting material to read when they were skimming through certain websites. Interestingly, some users learned to use it in unexpected ways: For her daily news site, P9 remarked, *"After I've got used to it, and know what to expect for this site, it's easy for me to see that there's no new news on it for me today."*

Most users liked the fact that the browser ran completely locally on their own computer, and hence their private information was not being shared with anyone else. P3 said, *"I like the idea of personalization without giving my data to anyone."*

5.4 Discussion

We believe that the principle of supplementing organic web browsing with color gleaned from personal history can be very useful in many settings. For example, in the domain of news, the Nielsen company reports that many users spend more time on social networking sites and blogs than on reading news [94]. One reason for this may possibly be that social networking feeds are inherently personalized, while generic news sites are curated by editors for a wide audience. Hence, people may pay more attention to the news that they are getting from friends than from mainstream media. Our browser highlights a possibly new way of disseminating news: mainstream news sites provide content, but the browser automatically picks up content that is likely to be relevant to the user.

We have found that experience-infused browsing is particularly useful for reading news or long blog posts. For instance, the website of The New York Times typically has over 120 stories on its front page. A small number of these are likely to be interesting to any particular user. Our browser can help users identify news stories related to particular places, people and organizations that the user is interested in or affiliated with. Similarly, the browser can be useful for the news site of a small community (such as a campus newspaper) because there is a chance that the user knows specific people in the community who may be mentioned on a particular day. It can also be used to highlight interesting items in social network feeds like those from Twitter and Google Plus.

One way of thinking about our browsing system is that it effectively creates lightweight “alerts” for thousands of terms in the personal index. When any of these terms are present on the current web page, the browser tries to ensure that the user notices their presence.

We found that over time, users built up a mental model of the kind of things our browser was good at highlighting on a particular site. They used our index to quickly guess when there was no useful content, which saved them time.

Of course, not everything that the browser highlights, even if accurate, is necessarily useful, and it may not always be desirable to only promote terms that the user has already encountered in the past. A general problem with personalization is that it can lead to online “filter bubbles” [98]. More work is needed to determine how, when, and to what extent an experience-infused browser should intervene. We hope that deploying systems such as ours will shed more light on this problem.

Another scenario (and indeed, the original motivation for the experience-infused browser) is similar to the “bulk-search” feature mentioned in the previous chapter. Instead of the user being connected to their own archive, they could connect to some other archive of special interest. And instead of pasting in text from which names are extracted and matched against the archive, the user can simply browse to any page, and see if any connections surface between the page and the archive.

5.5 Limitations and Future Work

We have thus far introduced the idea of experience-infused browsing, and described an initial implementation. With the help of a research prototype, we have tried to identify scenarios in which it may be useful. Several improvements and synergies with other techniques are possible to improve our prototype implementation, which can be explored in future work.

Currently, the browser is relatively easily fooled by common names. A name like *Michael Scott* may result in false hits. Our current way of solving this problem is to let users notice that a name is ambiguous and then eliminate it, if desired. A more effective way might be to look for ways of automatic disambiguation. For example if the *Michael Scott* one knows works at *Dunder Mifflin*, the name could be highlighted only if both names are present on the page. Techniques like semantic network profiles can be applied to solve this problem [41]. Similarly, the browser sometimes highlights broad place names like *U.S.A.*, which is generally not useful. Better semantic analysis to infer the specificity of a term can solve this problem. Inferring broader concepts from a page and not depending on just names is another area for future work.

Further, the importance of terms sometimes depends on context; the name of a university on the site of its campus newspaper is unremarkable, but probably deserves to be highlighted when it is in on the front page of a national newspaper. Combining our highlighting technique with detecting changes if the user previously visited the page (like Diff-IE [134]) may be one approach to filtering out unsurprising terms and focusing the user's attention. This may work well for sites that are frequently re-visited, such as online news sites. Identifying smart techniques that are effective in solving these problems, yet lightweight enough to use during regular browsing is an interesting area for future work.

While email and chat logs are conveniently accessible and capture a rich fraction of personal history for many users, a more comprehensive archive would incorporate more social streams such as those from Facebook and Twitter. Eventually, one could also consider importing a subset of web pages the user has visited (and perhaps spent a minimum amount of time on) in a browser into the archive. Needless to say, web browsing history is even more sensitive than email; assuring users about the security of a comprehensive personal digital archive is essential. Studying ways to integrate and score terms encountered via disparate sources and at different times in such an archive is another interesting challenge.

We can explore richer annotations that convey more information, such as using differently colored highlights depending on which group of people is most closely associated with the term, or providing pictures of associated people with the term as a means of expert-finding within one's network. Clearly, richer visual cues have to be presented in a subtle manner, so as to not overwhelm the user. Perhaps there could be user controls for what types of annotations she wishes to see; based on our experience so far, we feel that users will learn to sense what kind of cues are useful on different types of pages (news, listings, forums, etc), and use the controls accordingly.

Currently our browser does not change page layout in any way. It also accesses content only within the current page. This could be expanded to items that may be of interest to the user but are hidden deeper down on related pages. Some of our users commented that they would like the browser to chase navigation links from the front page of a news portal into its various sections to uncover stories that they might be interested in. A useful approach might be to identify relevant articles for the user from a news feed of all articles, and then generate a customized newspaper by laying out just the relevant articles on a single page.

The experience-infused browsing technique would be particularly useful on mobile devices where screen area is limited and it is relatively difficult to browse a large amount of text.

5.6 Conclusions

We have found that the experience-infused browser is useful to let users efficiently browse textual content by highlighting personally relevant named entities in web pages. Our approach of using personal archives as a way to capture a user's experiences and interests appears to be effective, and the technique of matching named entities on web pages is a useful heuristic. From our user studies we see that deploying these archives in the context of web browsing can be a valuable tool for improved personalization and web browsing using purely client-side mechanisms. This is a significant departure from personalization based on user profiling by specific services.

The prototype we have developed appears to strike a good balance between surfacing interesting information and not being obtrusive. Users may not find interesting highlights on every page, but every so often the browser augments the user experience in interesting ways. Users find material that they may have missed, notice friends where they did not expect and recall people that they have long forgotten. This notion of infusing everyday tools with our digital life experiences can potentially be used in many other settings.

Serendipity is, by definition, rare [5], so how does one build a tool that surfaces potential serendipitous facts without inundating users with irrelevant or obvious information? Instead of just listing all possible connections, our tool uses the information in the archive to highlight terms familiar to a user, thus providing the primary benefit of more efficient browsing while imposing little cognitive overhead. Users can click on highlighted terms only if they are curious about the connections to those terms. The general idea of supporting serendipitous discovery in everyday tools can be applied to other contexts.

In the next chapter, we will describe the last contribution of this dissertation, which is about providing personal, experience-infused search engines for everyone.

Chapter 6

Experience-Infused Search

We now turn to another example of experience-infused applications – the idea of experience-infused web search. Web search is one of the most commonly used applications on the Internet. While it is a marvel that search engines like Google and Bing can quickly look up and rank hundreds of billions of pages in response to a query, we hypothesize that users can often get their answers from a relatively narrow and uniquely personal slice of the web. In this chapter, we explore ways to automatically generate this slice, and thereby slant web search results towards sites of the user's interest. We show that a surprisingly small slice goes a long way towards satisfying the user's information needs.

In the early days of the Internet, users often created web pages with pointers to other pages that they liked. The original Yahoo directory is a prominent example. Algorithms like Pagerank were invented to take advantage of such curation and use the link structure of the web to rank search results [17]. However, the commercial importance of search engines has led to artificial manipulation of result rankings. A well-publicized example was J. C. Penney's success in influencing Google search to place its sites at the top for queries about a wide range of products [117].

In this context, there is value to bringing back some elements of human curation to the search problem. We use the observation that personalized and implicitly curated social recommendations of web content already exist in the user's personal archives (such as email) and social feeds (such as Twitter). Many Internet users already use social media for exchanging pointers to web pages that are of interest and relevance to them. For example, Twitter is extensively used for sharing links; it was reported in 2010 that 25% of the tweets per day contained links [111], and the total tweet volume in 2012 is above 400 million per day [43]. Similarly, users frequently recommend links to each other over email. A key idea in this chapter is to personalize search by *indexing only those domains mentioned in a user's online chatter.* We aim to let the user mine this ambient social chatter to infer sites that are likely to be relevant, and to use them while performing web search. Note that our approach is different from just searching the textual content in Twitter, email or the pages referenced therein because we use the *domains* referenced as the basis for creating a customized search index.

While personalization is currently being performed by the commercial search engines, it suffers from the same problems that we have seen in the previous chapter. Most of all, it disregards privacy issues – 73% of users in a 2012 Pew Internet poll said they did not approve of a search engine keeping track of their searches and using it to personalize future results because they felt it was an invasion of their privacy [106]. Our approach of experience-infused web search offers an alternate, privacy-respecting way to achieve personalization.

Our ideas are implemented in a system called Slant, so named because it attempts to bias search results towards a user's personal preferences. Before going on to a description of Slant and its evaluation, we preview the main contributions of this chapter.

- *Socially curated search.* We propose improving the quality of web search by extracting information from users emails and Twitter feeds to create personalized search indices. We designed four types of search indices that can be useful: links in email, links in tweets, friends' names, and links from top tweets.

- *Evaluation of social and curated search.* Using an experimental system that we built, we perform an exploratory study to compare the efficacy of different kinds of curated search by asking users to evaluate the performance of each search index on queries from their organic search history.

- *Empirical results.* We report quantitative results from our study, and discuss examples that provide qualitative insight. Briefly, we discovered that the quality of results from the email and Twitter-based indices perform comparably to personalized Google search over the whole web. We provide insight into the types of queries that do well with different search indices, and show that user satisfaction can potentially be increased using a combination of indices.

The implementation for SLANT is embedded in MUSE, but is packaged separately for ease of use and is publicly available at the URL: `http://mobisocial.stanford.edu/slant`.

6.1 Related Work

We now survey some prior work in the areas of web search, social information seeking and human curation.

6.1.1 Search Personalization

Teevan et al. studied the modeling of user interests, built from both search-related information and documents and email the user has read and created [132]. They attempt to use this information to create a profile of the user and re-rank the top 50 results returned by a search engine for each query. While their approach has better privacy guarantees than ours because it works purely on the client side, it cannot uncover results that may be buried deeper down in the search engine's ranking. It also does not utilize information embedded in social streams like Twitter. Another client-side solution that attempts to diversify results is due to Radlinski et al [110]; it does not utilize social data either. Jeh and Widom describe an approach to computing personalized Pagerank metrics starting from a seed set of interesting pages [59]. This approach requires more computation than an implementation of the domain-weighted indices that we use, which only need global page scores and weights for domains.

Prior work indicates that users frequently have a need to re-find information that they have encountered previously [1]. In some sense, SLANT generalizes this observation to hypothesize that sites previously encountered by users have a high chance of satisfying a user's information needs. Recently, there has been some work on identifying domain bias in search results, which is the idea that users may believe a page is more relevant because it comes from a domain that they are familiar with [58]. SLANT can potentially use this bias to increase user satisfaction by returning results from preferred domains.

6.1.2 Commercial Search Engines

Commercial search engines such as Google, Bing and Blekko have attempted to incorporate social elements into their search results. While the specific ranking algorithms are not publicly available, we summarize some of their features. Until recently, signed-in Google users who have connected their Google and Twitter accounts could see which of their Twitter followees have tweeted about a specific web page [22]. However, Google promoted the result only when the specific page in a search result had been tweeted (and not other pages on the site containing the page.) This would be useful when the tweeted page has a direct answer for the search query, but it does not help to mark the site as a trusted source. Bing has similar features, but is integrated with Facebook instead of Twitter, and therefore processes page "likes" on Facebook [84]. Blekko integrates with Facebook and allows users to specify a */likes* directive with search queries, which reports results from sites (not just pages) "liked" by their Facebook friends [16]. However, in our experience, Blekko's results tend to be noisy because when friends like any page on a very popular site, then all pages from that site are considered as endorsed by the friend, and tend to dominate search results. In addition to social data, an important source of curation used by these search engines is click through data on search results from humans, so much so that one search engine publicly accused another of stealing its clicks [127].

All these search engines are limited to the information that the user allows them to access, and tend to be fragmented because different services access different parts of the user's social network that they can pick up. They also require full access to the user's social data, and none of them take advantage of links and other information embedded in email. In contrast, the Slant approach not only gives users better control of their own data and privacy, but can also take advantage of multiple sources of chatter surrounding the user.

6.1.3 Social Q&A

There are several efforts to query social networks directly in response to an information need. Evans et al. analyze strategies employed by users when looking up information from their social network [35]. They asked users to learn about topics that were hard to directly search for, and yet found that social searching, even when including Q&A sites such as Yahoo Answers and the ability to ask questions of their social network, did not perform as well as with traditional web search. Morris et al. study scenarios of when users directly pose a query to their social networks and compare them to traditional web search [89, 88]. Smith et al. propose using social context i.e., a user's friends and the communities to which the user belongs to improve search for social media [123].

Other systems like Aardvark [57] and Collabio [15] attempt to steer user questions towards a person who is qualified to address their information need. In contrast to these systems, Slant combines the best of both worlds: automatic algorithms to harness the large pool of web content, and social curation to select high quality information sources.

6.1.4 Collaborative Searching

Search Together is a system that lets users collaborate in order to complete a specific task [86]. While links are exchanged between users, those links are used only for the duration of the task. Similarly, there has been much work in designing techniques to let groups work together on search tasks in a collaborative manner, whether or not they are co-located or working synchronously (e.g. [87]). In contrast, Slant indirectly captures implicit and long-term collaboration, not focused on any specific task. Moreover, users are not even aware that they are "collaborating" with others;

a link shared by a friend today may help improve a search result a year from now.

6.2 Social Search Indices

In this section, we describe four personalized search indices that we built in Slant to exploit various kinds of social chatter. These indices are meant to augment (not replace) the results of a regular search engine. We developed these four indices by observing the results on our own search queries over time, and tuned them by deploying early test versions with ourselves and a few lead users. We performed informal user studies along the way which helped us refine the algorithms and improve precision and recall. We describe below how we extract links and other information from social data; in the next section, we will explain how these are weighted and used to create a personalized search index for the user.

6.2.1 Email Links Index

Email archives are very personal and capture a lot of information directly relevant and meaningful to the user. As we have noted in earlier chapters, email is widely used, and users tend to have access to their long-term archives more than from any other source. Moreover, users routinely use email to send each other useful links, which constitutes an excellent form of curation. To create the email-based index, we ask users to download Slant to their computer, login to one or more email accounts and specify email folders from which links are to be extracted. Typically, we recommend that users select their sent mail folders and any other folders which they think has high-quality information, such as folders with emails from mailing lists that they trust. We parse the body of email messages in the specified folders and extract links from each message.

6.2.2 Friends' Names Index

People are generally interested in their friends' opinions and experiences. For example, when selecting a movie to watch, people will take a close friend's positive recommendation over an expert movie critic's negative review. SLANT extracts names of email correspondents from email messages (after performing entity resolution to account for multiple email addresses or name spellings for the same person) and assigns each person a closeness score based on communication volume. We use these names to perform query expansion – for each user query, we fork off multiple queries to the Google search engine by simply adding the name to the text of the query. For example, when a user with a friend named *Donald Knuth* searches for the term *pipe organ*, a query of the form *pipe organ + "Donald Knuth"* is automatically issued. We expand each query in this fashion for each of the top 150 friend names. Naturally, performing multiple searches implies a higher latency for getting search results; in practice, 150 searches works out to a latency of about 1-5 seconds per query, which is tolerated by most users for experimental purposes. We extract the top 10 results for each search query and score each link in the returned search result for each expanded query by simply accumulating the score for the user whose name was issued with the query. Thus, results containing names of people close to the user are likely to be ranked higher in this list. The results are assembled and ranked for the user, and presented using the same interface as normal Google search, including snippets. As we will describe later, users enjoy serendipitously discovering search results that involve their friends.

As with experience-infused browsing, SLANT also runs into problems with common names like *Michael Scott* that can lead to noisy search results. Our interface allows users to mark off search results with such noisy names so that future results with that friend's name are ranked below other results. We typically find that users encounter two or three friends with such names, but once these names are demoted, the noise level is considerably reduced.

6.2.3 Twitter Links Index

To form the Twitter link index, we broaden the notion of personal archives to include contents that a user may have seen on her Twitter feed. To create this index, we ask users to provide Slant access to their Twitter account. We extract all the links in tweets from accounts followed by the user. We also resolve shortened URLs employing domains like bit.ly and t.co that are common on Twitter.

6.2.4 TopTweets Index

Twitter publishes a listing of top tweets based on retweeting and sharing patterns [137]. We create a search index based on these top tweets to understand the performance of content that, while being human-curated, is not personalized for individual users. These links also allow those users who do not have a Twitter account or do not use it actively to experience the effect of human-curated search.

We chose the above four indices for study as each one has different characteristics. Emails are personal, private and generally related directly to the user's online activities. On Twitter, messages are usually public and users tend to follow others they trust, whether or not they know them personally. The topmost tweets across all of Twitter capture socially curated, but non-personalized URLs. Results with friends names may be highly personal and can be highly serendipitous. They directly point to expertise or opinions of the user's friends on the query topic.

6.3 Searching with Social Indices

We implement the backend to the search indices in Slant as a layer on top of the Google search engine, though other search engines with similar features could conceivably be used as well. When the user types in a search query, we perform the search using each of the four personalized search indices described above, along with regular Google search which is itself personalized if the user is logged into Google.

For searching sites with links extracted from Twitter and email, Slant uses the Google custom search engine [46]. The custom search engine restricts search results to specified sites. Weights can be assigned for each site and up to 5,000 URL patterns can be specified, which turns out to be adequate for the users in all our experiments. The Google custom search engine is popular with publishers and is frequently used by them to offer a customized search service restricted to their own sites, using Google's search technology. Any Google user can create a custom search engine, though this feature is little used by individual consumers – SLANT appears to be the first system that helps such users build customized search engines for themselves.

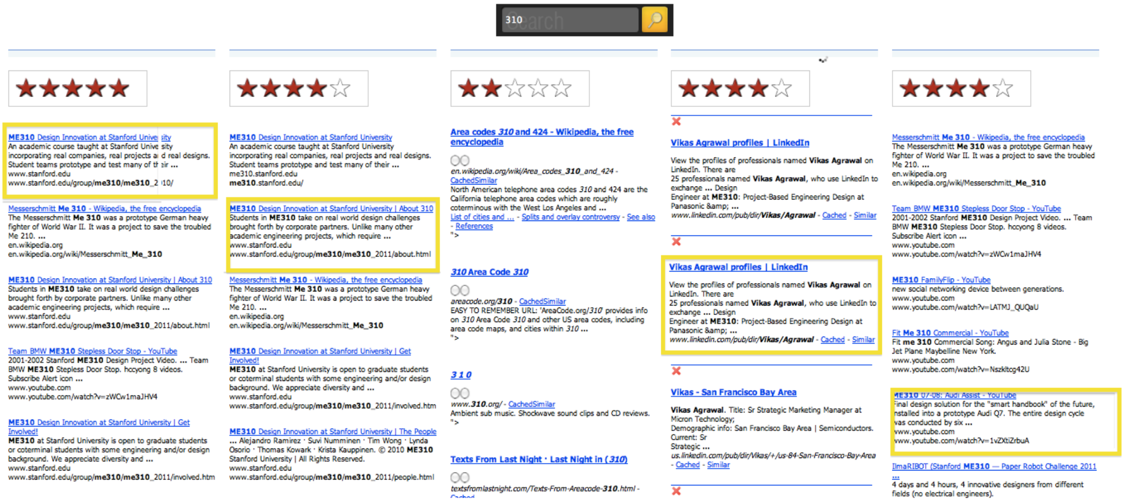
For each of the email, Twitter and TopTweets indices, SLANT instantiates a new custom search engine and populates it with the domains containing these links. For example, a page like *http:://example.com/123* would cause the domain *example.com* to be included in the search engine. We bias the weighting of a domain based on the frequency with which it appears in the extracted links. Weights for the Google custom search engines can be specified on a scale of −1 to 1. We assign weights to domains based on two factors:

- A positive bias to reflect the popularity of the domain in the corpus. The bias scales linearly from 0.1 to 1 depending on the domain's number of occurrences in the corpus.

- A negative bias to prevent major websites such as aol.com and amazon.com from overshadowing the more personalized results. We implemented this negative bias after we found them dominating search results, defeating the purpose of personalization. This bias is applied to a domain if its rank is one of the top N sites on the Internet (in our experiments, N=100). It scales linearly from −0.1 for the N-th ranked site to −1 for the topmost site.

In summary, each user's index is represented by a weighted vector of domain names that appear in the corpus. The weight for domain i is

$$w_i = \begin{cases} 0.1 + 0.9 \times c_i/c_{\text{max}} & \text{if } r_i > 100 \\ 0.9 \times c_i/c_{\text{max}} - 0.9 \times (100 - r_i)/100 & \text{otherwise} \end{cases}$$



★★★★★

ME310 Design Innovation at Stanford Unive sity
An academic course taught at Stanford Unive sity incorporating real companies, real projects a d real designs. Student teams prototype and test many of th ir ...
www.stanford.edu
www.stanford.edu/group/**me310/me310**_2 10/

Messerschmitt **Me 310** - Wikipedia, the free encyclopedia
The Messerschmitt **Me 310** was a prototype German heavy fighter of World War II. It was a project to save the troubled Me 210. ...
en.wikipedia.org
en.wikipedia.org/wiki/Messerschmitt_**Me_310**

ME310 Design Innovation at Stanford University | About 310
Students in **ME310** take on real world design challenges brought forth by corporate partners. Unlike many other academic engineering projects, which require ...
www.stanford.edu
www.stanford.edu/group/**me310/me310**_2011/about.html

Team BMW **ME310** Stepless Door Stop - YouTube
2001-2002 Stanford **ME310** Design Project Video. ... Team BMW **ME310** Stepless Door Stop. hccyong 8 videos. Subscribe Alert icon ...
www.youtube.com
www.youtube.com/watch?v=zWCw1maJHV4

ME310 Design Innovation at Stanford University | Get Involved!
ME310 at Stanford University is open to graduate students or coterminal students with some engineering and/or design background. We appreciate diversity and ...
www.stanford.edu
www.stanford.edu/group/**me310/me310**_2011/involved.htm

★★★★☆

ME310 Design Innovation at Stanford University
An academic course taught at Stanford University incorporating real companies, real projects and real designs. Student teams prototype and test many of their ...
me310.stanford.edu
me310.stanford.edu/

ME310 Design Innovation at Stanford University | About 310
Students in **ME310** take on real world design challenges brought forth by corporate partners. Unlike many other academic engineering projects, which require ...
www.stanford.edu
www.stanford.edu/group/**me310/me310**_2011/about.html

Messerschmitt **Me 310** - Wikipedia, the free encyclopedia
The Messerschmitt **Me 310** was a prototype German heavy fighter of World War II. It was a project to save the troubled Me 210. ...
en.wikipedia.org
en.wikipedia.org/wiki/Messerschmitt_**Me_310**

ME310 Design Innovation at Stanford University | Get Involved!
ME310 at Stanford University is open to graduate students or coterminal students with some engineering and/or design background. We appreciate diversity and ...
www.stanford.edu
www.stanford.edu/group/**me310/me310**_2011/involved.htm

ME310 Design Innovation at Stanford University | The People
... Alejandro Ramirez · Suvi Numminen · Tim Wong · Lynda Osorio · Thomas Kowark · Krista Kauppinen. © 2010 **ME310** Stanford University | All Rights Reserved.
www.stanford.edu
www.stanford.edu/group/**me310/me310**_2011/people.html

★★☆☆☆

Area codes *310* and 424 - Wikipedia, the free encyclopedia
*en.wikipedia.org/wiki/Area_codes_**310**_and_424* - CachedSimilar
North American telephone area codes *310* and 424 are the California telephone area codes which are roughly coterminous with the West Los Angeles and ...
List of cities and ... - Splits and overlay controversy - See also - References
">

310* Area Code *310
*areacode.org/**310*** - CachedSimilar
EASY TO REMEMBER URL: 'AreaCode.org/*310*' provides info on *310* Area Code *310* and other US area codes, including area code maps, and cities within *310* ...
">

3 1 0
*www.**310**.org/* - CachedSimilar
Ambient sub music. Shockwave sound clips and CD reviews.
">

Texts From Last Night · Last Night in (*310*)
*textsfromlastnight.com/Texts-From-Areacode-**310**.html* -
Cached

★★★★☆

✖
Vikas Agrawal profiles | LinkedIn
View the profiles of professionals named **Vikas Agrawal** on LinkedIn. There are
25 professionals named **Vikas Agrawal**, who use LinkedIn to exchange ... Design
Engineer at **ME310**: Project-Based Engineering Design at Panasonic &amp; ...
*www.linkedin.com/pub/dir/**Vikas/Agrawal*** - Cached - Similar

✖
Vikas Agrawal profiles | LinkedIn
View the profiles of professionals named **Vikas Agrawal** on LinkedIn. There are
25 professionals named **Vikas Agrawal**, who use LinkedIn to exchange ... Design
Engineer at **ME310**: Project-Based Engineering Design at Panasonic &amp; ...
*www.linkedin.com/pub/dir/**Vikas/Agrawal*** - Cached - Similar

✖
Vikas - San Francisco Bay Area
Vikas Agrawal. Title: Sr Strategic Marketing Manager at Micron Technology;
Demographic info: San Francisco Bay Area | Semiconductors.
Current: Sr
Strategic ...
us.linkedin.com/pub/dir/Vikas/+/us-84-San-Francisco-Bay-Area - Cached - Similar

✖

★★★★☆

Messerschmitt **Me 310** - Wikipedia, the free encyclopedia
The Messerschmitt **Me 310** was a prototype German heavy fighter of World War II. It was a project to save the troubled Me 210. ...
en.wikipedia.org
en.wikipedia.org/wiki/Messerschmitt_**Me_310**

Team BMW **ME310** Stepless Door Stop - YouTube
2001-2002 Stanford **ME310** Design Project Video. ... Team BMW **ME310** Stepless Door Stop. hccyong 8 videos. Subscribe Alert icon ...
www.youtube.com
www.youtube.com/watch?v=zWCw1maJHV4

ME310 FamilyFlip - YouTube
new social networking device between generations.
www.youtube.com
www.youtube.com/watch?v=LATMJ_QUQaU

Fit **Me 310** Commercial - YouTube
Fit **me 310** Commercial Song: Angus and Julia Stone - Big Jet Plane Maybelline New York.
www.youtube.com
www.youtube.com/watch?v=Nszkltcg42U

ME310 07-08: Audi Assist - YouTube
Final design solution for the "smart handbook" of the future, installed into a prototype Audi Q7. The entire design cycle was conducted by six ...
www.youtube.com
www.youtube.com/watch?v=1vZXtiZrbuA

IlmaRIBOT (Stanford **ME310** — Paper Robot Challenge 2011 ...
4 days and 4 hours, 4 innovative designers from different fields (no electrical engineers).

Figure 6.1: A screenshot of the interface used in our study. Results from the five different searches are shown side-by-side. From left to right: email-based, Twitter-based, personalized Google, friends names, and TopTweets. The user rates each set of results on a scale of 1 to 5. This screenshot shows the results when a user has issued the search query *310*. The results liked by the user for this query are highlighted in yellow.

where

r_i is the rank of domain i in the Alexa rating,

c_i is the number of times domain i occurs in the user's corpus, and

$c_{\max} = \max_i c_i$.

6.3.1 User Interface

As an experimental interface, SLANT presents the user with a single query box, and when the user types in a query, it fetches results from all five search indices. The results are presented side by side (see Fig. 6.1). This interface is designed only for prototyping and evaluation purposes; in a real deployment, we envision other interfaces such as a browser sidebar that combines results from all the social search indices.

6.3.2 An Example Query

To give readers a flavor of how these search indices differ, Figure 6.1 shows the results of a query organically performed by one of our users: the term *310*. The center column is normal, personalized Google search – it returns results about telephonic area code 310 as the top link. However, the user, a Stanford student, has in mind the course ME310, colloquially referred to among his friends as 310. The results from the indices seeded from email and Twitter links are biased towards the user's context and return results related to the course. The people search index in the fourth column returns a page with the name of a friend who has attended this course in the past, and happens to mention it on his resume. The TopTweets index has Youtube as one of the domains, and videos from the ME310 course appear in its results. In this particular example, each of the four social search engines in Slant was rated higher by the user than personalized Google search. Of course, different search indices do well on different types of queries, as we shall see later in this chapter.

6.4 Experimental Evaluation

To evaluate the different search indices we designed, we performed a user study with 7 subjects (2 female) in the age group 20-30, all students at our university and regular users of web search. We recruited users who self-reported themselves as having been active on Twitter for a period of at least 6 months, and to have access to an email account they had actively used for more than 1 year. On Twitter, our users followed a fairly typical assortment of friends, researchers, celebrities, journalists, university accounts, popular brands, etc.

For the email links index, users selected a subset of email folders (in addition to their sent email folders), which they judged to contain high-quality information. As mentioned earlier, this process is conducted entirely locally on users' own machines in order to assure them about privacy. We also processed the email messages to identify the top 150 people they interacted with and used these names for the friends' names index.

To work around the limits imposed by the Twitter API and gather a sizable corpus of tweets, we helped our users collect Twitter feeds over 4 rounds, at 7-10 day intervals for each user, according to their availability. This process was somewhat cumbersome and limited the size of the user pool for our study, since they had to be accessible to us at frequent intervals. But it is indicative of the general difficulties of accessing data on proprietary networks, compared to the relative ease of accessing email archives.

Users in our study had

- 5,397 to 11,398 emails over a couple of years,
- 807 to 1,442 unique domains in their email,
- 185 to 334 Twitter followees,
- 1,038 to 1,548 unique domains in their Twitter feed.

More details about these parameters are shown in Figure 6.2. The number of domains in each index is a few thousand, representing a tiny percentage of web sites that are actually present on the web and crawled by commercial search engines.

	Min	Median	Max	Mean
Number of Twitter friends	185	262	334	260
Number of Email messages	5,397	7,618	11,398	7,847
Unique domains from Twitter	1,038	1,201	1,548	1,231
Unique domains in email	807	1,161	1,442	1,150

Figure 6.2: Statistics on the personal search indices

For the TopTweets search index, we gathered links posted by the TopTweets account once a week, over 6 weeks. This process ensured that the index had a sizeable corpus to work with; in all, it consisted of 1,073 domains.

We conducted the experiment with each participant independently. All our participants used Google as their search engine by default. They were also generally logged in during their routine searches, except when using mobile devices. To lend ecological validity to our experiment, we asked them to look at their search history, (gathered by Google for signed-in users and available at http://www.google.com/history), and select about 10 queries to try out with Slant. Users were thus trying out our search indices with queries they had already organically performed on their own.

We provided users with the test interface shown in Fig. 6.1 and asked to enter their queries, while remaining signed in to Google. We asked them to carefully evaluate each column of search results (limited to the top 10 results), and rank the quality of each on a scale of 1 to 5. Participants were not told which column corresponded to which search index, in order to elicit unbiased feedback. Typically, users spent about 60 minutes performing 8 to 9 queries and evaluating and rating the results from each search index. The whole process including the mining of email and Twitter took about 6 hours per user. In all, we gathered detailed information on 59 queries across our users.

We recorded all queries and ratings for analysis. We employed a think-aloud methodology and recorded qualitative user comments to gain additional insights. While we recognize that this process may somewhat affect which queries users picked from their history, we judged it necessary; the specific queries and comments provide valuable insight into the efficacy of each search index.[1]

6.5 Results

In this section, we present a quantitative analysis of the results of our user study, noting the caveat that due to our small sample size and relatively uniform population, the results may not be very generalizable. However, we will complement the quantitative evaluation with qualitative analysis that illustrates the relative strength and weaknesses of the different indices.

We first testing the hypothesis that users tastes are very different, and that their search queries are satisfied by different sources. We then analyze how the different search indices in Slant perform and classify queries to understand the relationships between query types and evaluation results. Next, we evaluate whether different search indices provide different results and show evidence that combining these results

[1] The supplemental materials included with this dissertation contain a table listing all queries, along with their categorization, ratings and user comments on the results.

can be very effective.

6.5.1 Similarity in Content of Social Chatter

Restricting search to domains mentioned in social chatter achieves two objectives: (1) the domains are human curated and (2) they are personalized. How similar are users to each other? If most users are interested in the same set of the most popular sites, then there would be little need for personalization and we could just adopt a single human-curated index.

	User 2	User 3	User 4	User 5	User 6	User 7
User 1	0.16	0.05	0.09	0.05	0.14	0.12
User 2		0.12	0.03	0.07	0.13	0.11
User 3			0.18	0.10	0.06	0.08
User 4				0.03	0.12	0.08
User 5					0.04	0.07
User 6						0.13

Figure 6.3: Cosine similarity between weighted email indices of different users.

Fig. 6.3 shows the cosine similarity between the domain-weight vectors for the email indices of each pair of users. The vectors are normalized so all referenced domains have a positive weight. The cosine similarity ranges from 0 to 1, with 0 indicating orthogonality and 1 indicating equivalence. As we can see, the similarity between users is fairly low. This is true despite the fact that our seven users, all being students at the same university, are more likely to have similar interests than people in the general population. This indicates that different users tend to have different distributions of sites in their social chatter and therefore personalization is likely to be highly beneficial.

6.5.2 Rating the Search Indices

We first summarize the ratings users gave to the different Slant indices. Our baseline, personalized Google search, receives favorable average ratings from most users,

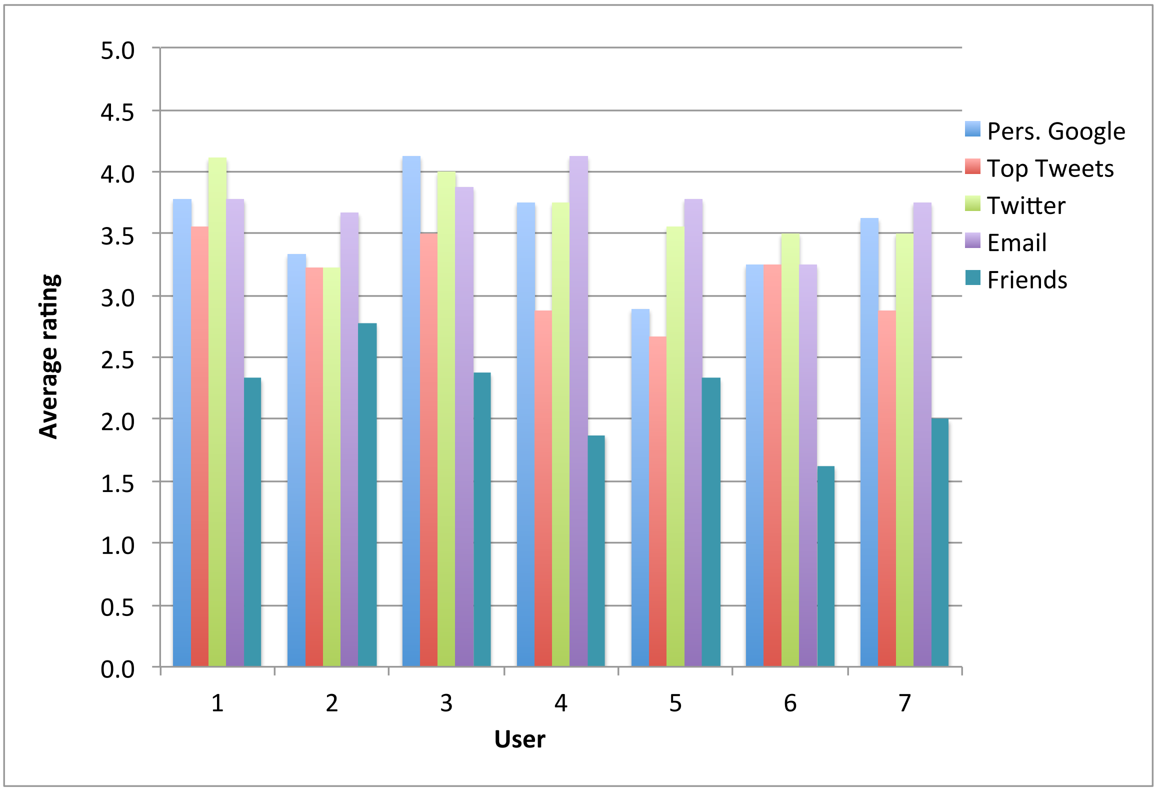


Figure 6.4: Average search quality ratings by different users for each index.

ranging from 2.9 to 4.1. Both Twitter and email-based indices also obtained similar ratings. The Twitter-based search index rates between 3.2 to 4.1, while the email-based search index rates between 3.3 and 4.1. For 4 out of 7 users, the email-based search was rated the best overall, compared to two for the Twitter-based index, and one for personalized Google search. This shows that social chatter based indices can match or outperform normal personalized search in terms of user satisfaction with results.

The search indices based on TopTweets and friends' names are not directly competitive with the other two personalized indices. The TopTweets index achieved average ratings from 2.7 to 3.6, and the index with friends names was rated between 1.6 and 2.8.

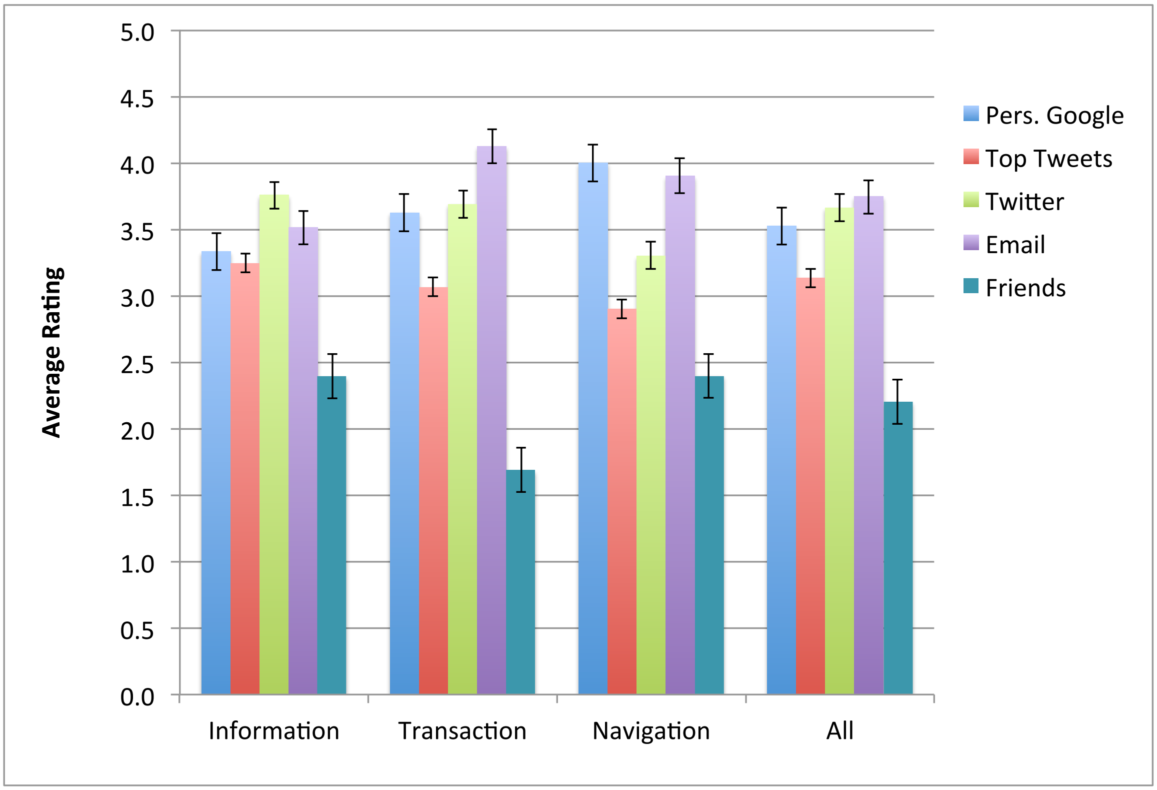


Figure 6.5: Average ratings across types of queries. Error bars indicate standard error.

6.5.3 Categories of Queries

To gain better insight into the strengths and weaknesses of the different search indices, we used Broder's web search taxonomy [18]. This taxonomy classifies queries into three categories: *informational*, where users are looking to get more information about a subject; *transactional*, where users are interested in performing a certain transaction, such as shopping or finding a file to download, and *navigational*, where users are looking for the URL of a specific site. Broder's analysis of query logs found that the distribution between informational, transactional, and navigational queries was 48%, 30%, and 20% respectively.

We manually classified the 59 query terms generated by our 7 users into one of these three categories. In some cases, if the categorization was somewhat ambiguous, we contacted the user and asked about his or her original intent. In these queries, the distribution between informational, transactional, and navigational queries is 56%, 27%, and 17% respectively, which is somewhat similar to the ratio reported by Broder.

Figure 6.5 shows the average ratings across the three types of queries, as well as the overall average. The overall averages across Google, Twitter, and email are about the same: 3.5, 3.7, and 3.8, respectively. Looking at the results of the different categories leads to some interesting observations:

- *Informational.* Both the Twitter and email based-indices fare slightly better than personalized Google search in the informational category. Informational is the most important category with over half of all queries, but it is also the hardest to do well on, compared to the other categories.
- *Transactional.* Email fares better in the transactional category, with an average rating above 4.
- *Navigational.* Both Google and email are better than Twitter in this category.

6.5.4 Qualitative Analysis

Both the Twitter and email-based search indices in Slant index a tiny fraction of the world wide web. It is therefore surprising that their results are comparable to, or even better than, regular web search which indexes every one of the billions of web pages. In fact, before the study, several of our participants indicated skepticism about our search indices due to this reason. To understand why and how these indices work well, let us discuss some actual queries and comments from users.

Informational Queries

Examples of queries where personalized Google search performs best are *health hazards of clementine fruits* and names of famous personalities like athletes and political activists. In these cases, Google provides the general information sought by the user, such as a Wikipedia entry or a news page. For example, a search on *wikileaks* also earned Google the highest score because the user was happy to see the latest news about it.

Email has the most number of unique high scores in this category. A primary characteristic is that the user typed in just a word or two and was presented with information of specific interest. As we reported earlier, a user who searched for *"310"* was looking for a course numbered ME310 and this was captured in the user's email custom search. In this case, the user exclaimed *"Wow this needed so little typing"*. Similarly, a search for *iphone5* returned an article from the technology blog TechCrunch, a favorite of the user, and a search for the single word *"dock"* returned the Bower and Wilkins dock, related to the user's interest in audio docks.

The Twitter search also returns several unique high scores. For example, the search *Simpsons Monday joke* returns the exact joke that the user was looking for. Users also tended to like blogs and articles on the topics searched.

This suggests that these search indices have different strengths. Google search tends to find general information for all queries. The Twitter links index returns results for what is trending, as well as interesting blogs and articles written by authors who the user is personally interested in. Finally, email contains a lot of links to domains of personal interest. We speculate that social chatter is useful for informational queries because users are often searching for information related to their conversations.

SLANT is also particularly useful for query terms that are commercially valuable and therefore likely to attract web spam. An interesting effect we have observed is that SLANT results tend to be more personalized and colorful than those from regular web search. One user remarked, *"I don't want results to be dull, like Wikipedia showing up on every query."*

Transactional Queries

SLANT's email-based index was clearly the best in the transactional category, and was rated 5 on half of the queries. We surmise that this is because email typically captures the user's personal context well. It often includes personal information such as bank transactions, addresses, receipts, shopping information, organizational or brand affiliations, etc. When users query for transactions related to a party they have already communicated with, this index is likely to generate good results. For example, a query like *lost cheque book* performed well with this index, without the user needing to specify details like the name of the associated bank. One user who searched for a query *disable unc check* related to compliance software was surprised to see a good result from a domain which had been emailed to him. Another user searched for *Russian roulette rihanna download* and remarked how she liked the result from a site she uses regularly.

Notably, Google search is rated the highest on all map and direction requests, particularly because it directly displays a map. It also provides the only 5-point rating on the query *matlab registry key.* Google is likely to perform well when looking for very specific information about a transaction. The Twitter-based index does better on queries like *Cauvery fishing camp* and *mac iso image mount.* It is likely that the human-curated links tweeted about these topics are of higher quality than the top Google links, and once again, avoid attracting spam.

Navigational Queries

Navigational queries are relatively unremarkable – we recorded almost no comments on the results relating to the top three indices. For these queries, all the Twitter ratings were between 2 and 4, while both Google and email score 5 on 40% of the queries. This is perhaps not surprising since Twitter is primarily used to share noteworthy news, and less for mundane navigation.

Friends Names Index

Although the search index based on friends names performed poorly overall, it has turned up a number of unexpected and serendipitous results that delighted users. For example, one user searched for *parasailing san diego* and was pleasantly surprised to find a Picasa photo album of a friend's visit to San Diego. Another searched for *Boston University* and discovered that a colleague of his was going there for his MBA. In yet another instance, a professor was looking up *Zigbee* and found that one of her own Ph.D. students had written a paper on the topic, a fact she had not known. This kind of serendipitous finding, while rare, is remarkable. A topic for future work would be explore methods to improve the chances of this happening, and to estimate the quality of results from this index, so that they can be shown only if they are likely to surprise the user.

Query Length

It is well known that average search query length has been going up over the last decade [67], and that users need to formulate more precise queries to obtain accurate results. It is customary for users to iterate on queries when the results do not turn up answers they need. A couple of our users remarked how little typing was needed to get the results they wanted. Twitter and Email take advantage of the users' context to help disambiguate information, thus minimizing typing. The *310* query mentioned earlier in this chapter is one such example. The ability to specify shorter queries is particularly useful when searching on mobile devices and other limited function input devices such as TV remotes, where typing is inconvenient.

6.5.5 Correlation Between Search Engines

Although the average ratings of the top three indices are quite similar, we observed that the ratings for each individual query vary quite widely. In addition, the qualitative results suggest that different searches perform well for different reasons. This leads us to ask if any of the indices are correlated to each other in terms of their ratings. The Pearson correlation coefficients between the ratings for each pair of search indices are shown in Figure 6.6. Except for a slight positive correlation between Google and TopTweets, there is very little correlation between any other pair of search indices. The lack of correlation suggests that the indices are complementary.

	TopTweets	Twitter	Email	Friends
Pers. Google	0.21	0.05	-0.08	-0.08
TopTweets		0.13	-0.04	0.10
Twitter			-0.10	0.05
Email				0.08

Figure 6.6: Correlation between the ratings of different search indices.

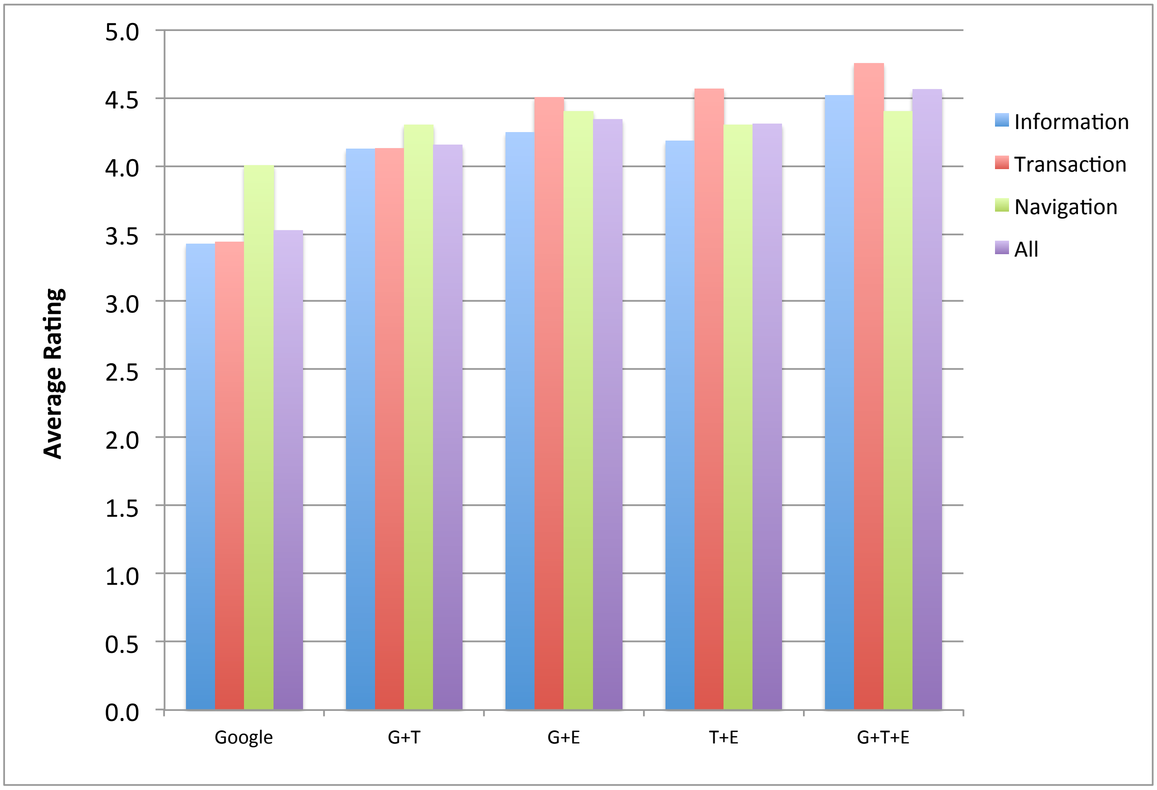


Figure 6.7: Combining the best of Google, Twitter and Email. G+T: Google+Twitter; G+E: Google+Email; E+T: Email+Twitter; G+T+E:Google+Twitter+Email

6.5.6 Combining Results from Multiple Indices

If the indices are complementary, it seems like we could combine the best of all worlds and switch between engines as appropriate, if only we could tell which index to use when. While designing such a system is beyond the scope of our work, we can ask: How good would such a system be? Our results can be used to compute the theoretical upper bound of a combined search with the maximum rating of its components.

Figure 6.7 shows the possible ratings for four combinations, compared to the baseline of personalized Google search. The baseline obtains an average of about 3.5 for all categories, except for scoring a 4.0 average on the navigational category. Adding Twitter-based search boosts the average to above 4.1 in all categories; similarly, adding email-based search to Google boosts the average about 4.2, with transactional queries scoring a high 4.5 average. Interestingly, we can combine just Twitter and email-based results to achieve a similarly high rating. In other words, combining any two of the top three search indices will provide a significantly improved result, bumping up the average from 3.5 by 0.7 points, a gain of 20%. While users in our study are active on both email and Twitter, this result suggests that users who have a good corpus on only one of email or Twitter can benefit by adding their medium of choice to augment Google search.

Finally, we can combine results from all the top three engines. In this case, the average rating improves from 3.5 to 4.6, an impressive gain of about 31%. Specifically, out of the 59 queries, in 37 queries at least one index gets the highest possible score, in 20 queries the highest score awarded is 4 and in only 2 queries the highest score assigned to any index is a 3. With over 60% of the queries receiving the best possible score, combining the three search indices can produce an exceptionally powerful search engine.

In our current study, each search index returns ten results. A straightforward way of creating a combined engine is to simply return ten results from each search. Combining two searches will generate 20 results; combining three will generate 30. Another option is to return the top 3 results from each. While it is preferable to return satisfactory results in the fewest number of search hits, further investigation of ways to combine the indices is desirable, given the likely increase in user satisfaction.

6.5.7 Summary

We now summarize the major findings from our evaluation.

1. The personalized SLANT indices have little similarity between individuals. This suggests that personalization is important.

2. Each user found the quality of email and Twitter-based searches in SLANT similar to personalized Google, even though the results returned are very different, and from a small subset of domains.

3. Google is better at returning general results for terms and helping navigate to specific web sites when users type in its distinguishing characteristics. The

Twitter-based index is better at returning trending topics and opinions, blogs and articles of interest to users. The email-based index, being personal, helps in disambiguating terms, thus minimizing the need for typing long queries and search refinement. Email-based search is useful for transactional queries because the domains that users routine interact with are often embedded in email.

4. Combining the best results from different search indices can significantly improve user satisfaction. Combining results from any pair of Google, Twitter-based, and email-based indices boosts the average rating from 3.5 to above 4, and combining all three indices increases it to 4.6.

5. Although the friends names index generates generally poor results, it occasionally returns highly serendipitous results that surprise the user.

6.6 Discussion

Our results are evidence of the somewhat surprising fact that custom search engines that index a tiny fraction of the entire world wide web can perform comparably with traditional search engines for general-purpose web search. There could very well be some effect of domain bias that we discussed earlier [58], but we believe there is value to giving users the results that they trust and that satisfy them.

A slightly intangible benefit of using Slant is that search results tend to be more entertaining than the uniform and somewhat monotonous results of a traditional search engine. Commentators have said that each search engine has its own search voice, a unique set of results it provides based on its collection of documents and its particular method of ranking, and there is some concern that the number of these voices is limited to a very small number of large search providers [127]. Personalized indices appear to have their own voice and may be one way of alleviating this problem.

In the future, we envisage that users may see multifaceted search results. They can get authoritative and definitive results from a global search engine, along with the more personal and entertaining results from personal search engines that are automatically assembled from their social chatter.

Of course, social search indices also have limitations. We believe that the approach of generating socially curated results complements traditional search engines, but does not replace them. The quality of the indices in Slant is dependent on the extent and quality of email archives, or links tweeted by the user's followees.

6.6.1 Implicit Consumption of Information

There are multiple levels of social chatter surrounding a user. The chatter may involve the user's inner social circles, outer social circles, friends of friends, people the user trusts but does not know personally, or members of a wider community. Each of these sources provides information valuable in different settings, though it is difficult for the user to process each nugget of information manually. Following too many sources takes too much time, while following too few represents a potential loss of opportunity.

Active Twitter users report being inundated with too much information, and are only able to "dip a toe" in the torrent of tweets [14] meaning that they rarely read everything in their stream. Similarly, many people may want to be on mailing lists related to their broader interests, but have no time to read even more email than they already get, and hence limit their subscriptions to lists of direct and current interest. Slant's approach of funneling social chatter into a personal search engine benefits users by enabling them to indirectly consume streams of information they may otherwise have missed – not by reading them explicitly, but by capturing recommendations embedded in them. Even for messages they have already read, it is difficult for people to remember all the recommendations and links they contain, and it is therefore valuable to pipe them automatically into a personalized search engine.

We envisage that such implicit consumption (and extensive collection) of social information will become commonplace in the future, resulting in users being able to follow more people on Twitter or subscribing to more mailing lists. We envisage that a user may have two categories of followees or mailing lists: one that he actually reads, and one that is used simply to gather recommendations to pipe into a search index.

6.6.2 Privacy Considerations

Today's search engines collect detailed user information such as search history, location, profile, and social data in order to personalize web search. Slant is a mostly client-side approach that tackles some of the privacy issues inherent in this model by allowing the user to analyze her communications under her own control. Only the results of the analysis, such as preferred sites need to be part of the user's search index; even if this index is then given to a search engine, users have better privacy since the search engine is not listening in on their detailed communications. While search engines have been personalizing search results for users, there is a sense of discomfort among users in giving out detailed information such as their email and clickthrough patterns to "Big Brother" portals. One of our users commented: *"I would not like to share my personal information with Google, or for that matter any company, as I always feel that it is a definite risk of loss of privacy. There could also be accidental information leak by a company with whom I might share my details, like Blippy accidentally leaked credit card information."*

A further benefit is that since only the user has access to all of his or her social data, the personalization can be relatively accurate and complete. In contrast, commercial search engines tap into social information only from specific sources of social media, based on commercial agreements between them and the source. For example, in July 2011, Google ended the commercial arrangement with Twitter that let a user's friends tweets appear in search results [128]. Similarly, Google search has no visibility into a user's Facebook information.

6.6.3 Spam Elimination

An important benefit of using social media is the elimination of many kinds of spam and artificial gaming of search results. Users choose who to follow in Twitter; and while one can receive email spam, Slant uses only messages in email folders that the user has chosen. The problem of spam filtering in email is also independent of link-indexing and search, and is addressed by an independent body of work.

6.7 Conclusions

Using Slant, we have shown that social chatter from email and social media can be leveraged to improve web search. Social data is personalized, free of spam and can be analyzed directly by users on the client side, alleviating privacy concerns.

User studies with Slant reveal that search results derived from socially curated indices can be as good if not better than the results of commercial search engines, even when they use personalized search. We envision that combining results from these different indices will generate superior results that satisfy users for many types of queries.

Chapter 7

Conclusions and Future Work

We now make some concluding observations and describe avenues for future research. Focusing primarily on email archives to enable experimentation, we have built and released systems in several areas and tried to characterize at an early stage the value that people can derive from their personal archives. We have provided some initial ideas, but we believe that even more exciting work remains to be done to extend these ideas.

It is clear that billions of personal histories are being captured in the digital tools being used by consumers. We believe that this can lead to a new class of assets for consumers. Far from being useless, old documents, archives can be used in creative ways to revive memories and build personal profiles that can be used by many applications.

7.1 Working with Archives

As discussed in Chapter 2, our experiences with Muse have uncovered several possible applications for email archives. For example, these archives may reflect valued relationships that are languishing and need to be renewed, or make users feel a sense of confidence by looking back at their past accomplishments, or help people reminisce and write memoirs about important phases of their lives. It may be worth following up on some of these observations and designing experimental systems to conduct deeper research.

Our experience with Muse indicates that the detailed information captured in personal archives can be useful to help understand the nature of autobiographical memory and the kinds of information that users find useful, memorable, or indeed, useful but hard to remember. This understanding in turn can be used to inform the design of a wide array of life-logging tools and data collection methods that are meant to capture data that may be useful over a lifetime. With the example of the crossword puzzle, we have also begun to see that gamification can be a useful way of engaging people with their archives. This is an area that is ripe for research.

Tools that help in sense-making with archives are a pre-requisite to working with them. In this dissertation, we have proposed cues that appear to be useful enough that a few thousand people have downloaded Muse and broadly enjoyed using it (and about 50 of them even “liked” it on Facebook!). The cues that we have found useful are: groups, sentiments, name summaries over time and image attachments. We have learned that it is important to focus not just on cues and visualizations, but also on interfaces that allow rapid navigation of the actual messages.

The data mining and visualization techniques we have used are fairly lightweight, and chosen with an eye towards building a usable tool that can uncover valuable applications of archives. Many synergies with more sophisticated text mining and visualization techniques are possible, including those we discussed in section 2.1. Several types of cues other than the ones we suggested are possible. Our initial attempts to identify places and visualize them on a map did not work well due to the ambiguity in place names; however, this problem can be addressed with better implementation techniques. Similarly, our experience with trying to use topic models over email archives was not very effective, but tuning these models to achieve meaningful results on personal data is an interesting area of research. In general, there is a lot of text analysis work applied to public corpora which can also be tried on personal archives. While evaluation of research systems on private data is not very easy, personal data is very meaningful to users and these systems can have a large impact. The IBM Watson system has demonstrated recent advances in open-domain question and answering over public knowledge [38]; it is interesting to think what this may mean for natural language understanding from personal archives.

Going beyond textual data, it is likely that other data types may need entirely new techniques for the purposes of sense-making over archives. For example, data based on media or locations poses new challenges.

Our work with archival organizations has made progress towards making the processing and use of email archives easier than it used to be. However, there are many other forms of "Born Digital" materials that will have to be assimilated into the archives in the future, throwing up new and interesting challenges.

7.2 Experience-Infused Applications

Our ideas of experience-infusion envision a new style of personalization that lets a user customize an application based on the detailed profile already captured in her archives. This style of personalization allows consumers to regain control of their personal data, and yet obtain the benefits of personalization.

We have demonstrated the efficacy of this approach in two widely used applications, web browsing and web search. The experience-infused browser brings users the benefit of inline recall from their archives, and experience-infused search allows people to get information from sites that they are likely to know and trust. These implementations are meant to illustrate the idea and inspire application designers to consider this style of personalization; even with our simple techniques of matching names and using curated links from archives, the results are promising. We expect that more sophisticated analysis techniques such as natural language understanding will provide better accuracy for the implementations that we have described. So far, we have discussed the scenario of consumers using their own archives, but social sharing of information in these archives is an interesting area for research.

Personalization is especially useful on mobile devices due to the small form factor, but rich user context available. Mobile devices can capture all kinds of useful information such as location data than can go into a personal archive. We have not explored this area in this dissertation, but it is likely to prove interesting for research.

7.3 Active Management of Archives

The results of experience-infused browsing and search depend on the quality of data present in the personal archive. For initial experimentation, we have focused on the email domain because it is widely used and contains not just communications, but also reflects transactions such as shopping records, tickets booked, etc. Instant messaging logs are also useful as they reflect real-time discussions among people and encompass a wide range of information, from restaurant reviews to discussions about buying a car.

We envision that in the future, as users get used to experience-infused applications, they will actively enhance and manage their digital archives. For example, a student interested in his classmates can simply import the class directory into the archive. A professor could do the same for all the students she has taught over the years. People can save business cards or LinkedIn profiles of their contacts in their digital archives. Now, whenever the name of any of these people is on a web page being browsed, the user will not miss noticing it. Such lookups often turn out to be useful when looking at a page containing a lot of names – for example, the list of attendees at a large event or conference. We envision easy connectors being built to enable such imports, but for now, users can simply email themselves the information they wish to be looked up from their archives.

In the future, it is possible that people may wish to subscribe to a large but still curated amount of chatter on Twitter, mailing lists or blogs. They may read this chatter occasionally, but more importantly, experience-infused applications can automatically imbibe and process the recommendations embedded in them. The user will not have to remember all the information that has flashed by, yet it can be made instantly available when needed.

Similarly, students may import the class materials for all their courses into their personal archive, so that when they visit a related web page at any time in the future, the browser can insert a link back to their original class material. Researchers can import their own papers into the personal index, so they can spot related material when reading other papers. Needless to say, the larger the archive, the more noisy it can become, so easy-to-use filtering controls over the archive are essential.

These scenarios indicate that the ability to store and recall information from the personal archive can be useful in a wide range of applications.

Of course, not all information should necessarily be archived – for example, some memories are better forgotten or are too sensitive to be preserved. As Mayer-Schonberger has argued, there is some virtue to forgetting in the digital age [82]. Our research techniques offer ways for users to tap into the power of archives if desired and useful; but we would also like to see users retain control over the archive, so that unwanted parts of them may be deleted or disabled.

7.4 How will the Historical Record Change?

Personal digital archives have the potential to change how history is recorded and studied. Certainly, digital archives of eminent individuals will be studied intensively by scholars, much the way historical letters have been in the past. Systems such as ours will ease the task of scholars by making a large set of archives conveniently accessible.

More broadly, histories can now be recorded and studied for ordinary users, not just a chosen few. Families will likely broaden their archives to include a lot of material, not just photographs. Children growing up in the digital age will have detailed records of everything they do, and many other things their families wish them to have access to. It will be easy for people to look up their histories to record their memoirs, and tell the story of their lives.

Over time, this may affect the process of collection itself, as people realize the long-term value of their archives, and see some of their current actions from the perspective of long-term recall by themselves or perhaps others. It remains to be seen how social norms for what should or should not be archived, and who should have access to the data, will evolve over the next few decades.

Bibliography

[1] Eytan Adar, Jaime Teevan, and Susan T. Dumais. Large scale analysis of web revisitation patterns. In *Proceedings of CHI '08*. ACM, 2008.

[2] Eytan Adar, Jaime Teevan, and Susan T. Dumais. Resonance on the web: web dynamics and revisitation patterns. In *Proceedings of CHI '09*. ACM, 2009.

[3] Rakesh Agrawal, Tomasz Imielinski, and Arun N. Swami. Mining Association Rules between Sets of Items in Large Databases. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pages 207–216, 1993.

[4] AIMS Work Group. AIMS Born-Digital Collections: An Inter-Institutional Model for Stewardship. *White paper*, October 2011.

[5] Paul André, M. C. Schraefel, Jaime Teevan, and Susan T. Dumais. Discovery Is Never by Chance: Designing for (Un)Serendipity. In *Proceedings of the seventh ACM Conference on Creativity and Cognition*. ACM, 2009.

[6] Sebastian Anthony. Relive and analyze your entire email archive. *Extreme Tech*, Oct. 17, 2011.

[7] The Internet Archive. http://archive.org.

[8] Sitaram Asur and Bernardo A. Huberman. Predicting the Future with Social Media. volume abs/1003.5699. Arxiv.org, 2010.

[9] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In *Proceedings of Language Resources and Evaluation (LREC'10)*. European Language Resources Association, May 2010.

[10] A. Banerjee, C. Krumpelman, J. Ghosh, S. Basu, and R.J. Mooney. Model-Based Overlapping Clustering. In *Proceedings of KDD '05*. ACM, 2005.

[11] Jeff Barr. Amazon Glacier: Archival Storage for One Penny Per GB Per Month. In *Amazon Web Services blog*. Amazon.com, Aug. 21, 2012.

[12] R Beale. Supporting serendipity: Using ambient intelligence to augment user exploration for data mining and web browsing. *International Journal of Human – Computer Studies*, 65(5):421–433, 2007.

[13] Gordon Bell and Jim Gemmell. *Total Recall: How the E-Memory Revolution Will Change Everything*. Dutton Adult, 2009.

[14] Michael S. Bernstein, Bongwon Suh, Lichan Hong, Jilin Chen, Sanjay Kairam, and Ed H. Chi. Eddi: interactive topic-based browsing of social status streams. In *Proceedings of UIST '10*. ACM, 2010.

[15] Michael S. Bernstein, Desney Tan, Greg Smith, Mary Czerwinski, and Eric Horvitz. Personalization via friendsourcing. *ACM TOCHI*, 17:6:1–6:28, May 2008.

[16] Blekko. Friends make search better!, Retrieved Aug. 29, 2011.

[17] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. In *Proceedings of WWW '98*. Elsevier, 1998.

[18] Andrei Broder. A taxonomy of web search. *SIGIR Forum*, 36:3–10, September 2002.

[19] Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl. *The Adaptive Web*, volume 4321 of *Lecture Notes in Computer Science*. Springer, 2007.

[20] Vannevar Bush. As We May Think. *Atlantic Monthly*, 176, 1945.

[21] Evan Carroll and John Romano. *Your Digital Afterlife: When Facebook, Flickr and Twitter Are Your Estate, What's Your Legacy?* New Riders, 2010.

[22] Mike Cassidy. An update to Google social search. *The Official Google Blog*, Feb. 17, 2011.

[23] George Chin, Jr., Olga A. Kuchar, and Katherine E. Wolf. Exploring the analytical processes of intelligence analysts. In *Proceedings of CHI-2009*. ACM, 2009.

[24] A. Clauset, M.E.J. Newman, and C. Moore. Finding Community Structure in Very Large Networks. *Physical Review E*, 70(6):66111, 2004.

[25] Christopher Collins, Martin Watternberg, and Fernanda Viegas. Parallel Tag Clouds to Explore and Analyze Faceted Text Corpora. In *Proceedings of VAST '09*. IEEE, 2009.

[26] Aron Culotta, Ron Bekkerman, and Andrew Mccallum. Extracting Social Networks and Contact Information from Email and the Web. In *Proceedings of the Conference on Email and Anti-Spam '04*, 2004.

[27] Edward Cutrell, Daniel Robbins, Susan Dumais, and Raman Sarin. Fast, flexible filtering with Phlat. In *Proceedings of CHI '06*. ACM, 2006.

[28] David S. Ferriero. A New Presidential Library. *AOTUS blog*, 2012.

[29] Department of Special Collections. Email Archives in the Robert Creeley Papers, M0662. Stanford University Libraries, Stanford, CA., Retrieved Dec. 1, 2012.

[30] Mark Dredze, Hanna M. Wallach, Danny Puller, and Fernando Pereira. Generating summary keywords for emails using topics. In *Proceedings of IUI '08*. ACM, 2008.

[31] Susan Dumais, Edward Cutrell, JJ Cadiz, Gavin Jancke, Raman Sarin, and Daniel C. Robbins. Stuff I've Seen: a system for personal information retrieval and re-use. In *Proceedings of SIGIR '03*. ACM, 2003.

[32] Martin Dzbor, John Domingue, and Enrico Motta. Magpie: Towards a Semantic Web Browser. In Dieter Fensel, Katia Sycara, and John Mylopoulos, editors, *The Semantic Web - ISWC 2003*, volume 2870 of *Lecture Notes in Computer Science*, pages 690–705. Springer Berlin / Heidelberg, 2003.

[33] Hermann Ebbinghaus. *Memory: A Contribution to Experimental Psychology*. Teachers College, Columbia University, 1885.

[34] David Elsweiler, Mark Baillie, and Ian Ruthven. Exploring memory in email refinding. *ACM Transactions on Information Systems*, 26(4):1–36, 2008.

[35] Brynn M. Evans, Sanjay Kairam, and Peter Pirolli. Do your friends make you smarter?: An analysis of social strategies in online information seeking. *Information Processing and Management*, 46(6):679 – 692, 2010.

[36] Facebook. Facebook Groups: How do I chat with a group?, Retrieved Dec. 1, 2012.

[37] Facebook. Facebook Featured Friends: How do I feature specific friends on my profile?, Retrieved Dec. 1, 2012.

[38] D. A. Ferrucci. Introduction to "This is Watson". *IBM Journal of Research and Development*, 56(3.4):1:1 –1:15, May-June 2012.

[39] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2005.

[40] Joshua Foer. *Moonwalking with Einstein*. The Penguin Press, 2011.

[41] Susan Gauch, Mirco Speretta, Aravind Chandramouli, and Alessandro Micarelli. User Profiles for Personalized Information Access. In *The Adaptive Web*, volume 4321 of *Lecture Notes in Computer Science*, pages 54–89. Springer, 2007.

[42] Jim Gemmell, Gordon Bell, and Roger Lueder. MyLifeBits: a personal database for everything. *Communications of the ACM*, 49(1), 2006.

[43] Tomio Geron. Twitter's Dick Costolo: Twitter Mobile Ad Revenue Beats Desktop On Some Days. *Forbes*, June 6, 2012.

[44] Eric Gilbert and Karrie Karahalios. Predicting tie strength with social media. In *Proceedings of CHI '09*. ACM, 2009.

[45] Andrea Goethals and Wendy Gogel. Reshaping the Repository: The Challenge of Email Archiving. In *Proceedings of the 7th International Conference on Preservation of Digital Objects*, 2010.

[46] Google. http://google.com/cse, Retrieved Dec. 1, 2012.

[47] Jim Gritton. Of serendipity, free association and aimless browsing: do they lead to serendipitous learning? Unpublished M.Sc. Dissertation, The University of Edinburgh, 2007.

[48] The Guardian. The Sarah Palin emails, 2011.

[49] Philip J. Guo. *The Ph.D. Grind. A Ph.D. Student Memoir*. http://phdgrind.com, 2012.

[50] Joshua Hailpern, Nicholas Jitkoff, Andrew Warr, Karrie Karahalios, Robert Sesek, and Nik Shkrob. YouPivot: Improving recall with contextual search. In *Proceedings of CHI '11*. ACM, 2011.

[51] Sudheendra Hangal, Peter Chan, Monica S. Lam, and Jeffrey Heer. Processing Email Archives in Special Collections. In *Digital Humanities 2012 Conference Abstracts*. Hamburg University Press, 2012.

[52] Sudheendra Hangal, Monica S. Lam, and Jeffrey Heer. MUSE: Reviving Memories Using Email Archives. In *Proceedings of UIST-2011*. ACM, 2011.

[53] Sudheendra Hangal, Abhinay Nagpal, and Monica Lam. Effective browsing and serendipitous discovery with an experience-infused browser. In *Proceedings of IUI '12*. ACM, 2012.

[54] S Havre, E Hetzler, P Whitney, and L Nowell. ThemeRiver: visualizing thematic changes in large document collections. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):9–20, 2002.

[55] Jeffrey Heer and Danah Boyd. Vizster: Visualizing Online Social Networks. In *Proceedings of INFOVIS '05*. IEEE Computer Society, 2005.

[56] Lichan Hong, Ed H. Chi, Raluca Budiu, Peter Pirolli, and Les Nelson. SparTag.us: A low cost tagging system for foraging of web content. In *Proceedings of the working conference on Advanced Visual Interfaces*, AVI '08. ACM, 2008.

[57] Damon Horowitz and Sepandar D. Kamvar. The anatomy of a large-scale social search engine. In *Proceedings of WWW '10*. ACM, 2010.

[58] Samuel Ieong, Nina Mishra, Eldar Sadikov, and Li Zhang. Domain bias in web search. In *Proceedings of WSDM '12*. ACM, 2012.

[59] Glen Jeh and Jennifer Widom. Scaling personalized web search. In *Proceedings of WWW '03*. ACM, 2003.

[60] Vaiva Kalnikaite and Steve Whittaker. A saunter down memory lane: Digital reflection on personal mementos. *International Journal of Human – Computer Studies*, 69(5):298 – 310, 2011.

[61] Sepandar D. Kamvar and Jonathan Harris. We Feel Fine and searching the emotional web. In *Proceedings of WSDM-2011*. ACM, 2011.

[62] Joseph 'Jofish' Kaye, Janet Vertesi, Shari Avery, Allan Dafoe, Shay David, Lisa Onaga, Ivan Rosero, and Trevor Pinch. To have and to hold: exploring the personal archive. In *Proceedings of CHI '06*. ACM, 2006.

[63] Sarah Kim. *Personal Digital Archives: Preservation of Documents, Preservation of Self*. Ph.D. dissertation, The University of Texas at Austin., 2012.

[64] David S. Kirk and Abigail Sellen. On human remains: Values and practice in the home archiving of cherished objects. *ACM TOCHI*, 17(3):10:1–10:43, 2010.

[65] G. Kossinets and D.J. Watts. Empirical Analysis of an Evolving Social Network. *Science*, 311(5757):88, 2006.

[66] A. Lancichinetti, S. Fortunato, and J. Kertész. Detecting the Overlapping and Hierarchical Community Structure in Complex Networks. *New Journal of Physics*, 11:033015, 2009.

[67] Frederic Lardinois. Hitwise: Search queries are getting longer. *ReadWriteWeb*, Feb. 24, 2009.

[68] James Lawley and Penny Tompkins. Maximising Serendipity: The art of recognising and fostering unexpected potential - A Systemic Approach to Change. *cleanlanguage.co.uk*.

[69] Jure Leskovec, Kevin J. Lang, Anirban Dasgupta, and Michael W. Mahoney. Statistical properties of community structure in large social and information networks. In *Proceedings of WWW '08*. ACM, 2008.

[70] Henry Lieberman, Christopher Fry, and Louis Weitzman. Exploring the Web with reconnaissance agents. *Communications of the ACM*, 44:69–75, August 2001.

[71] Sin E. Lindley. Before I Forget: From Personal Memory to Family History. *Human - Computer Interaction*, 27(1-2):13–36, 2012.

[72] LinkedIn Maps.

[73] Jiahui Liu, Peter Dolan, and Elin Rønby Pedersen. Personalized news recommendation based on click behavior. In *Proceedings of IUI '10*. ACM, 2010.

[74] Shixia Liu, Michelle X. Zhou, Shimei Pan, Yangqiu Song, Weihong Qian, Weijia Cai, and Xiaoxiao Lian. TIARA: Interactive, Topic-Based Visual Text Summarization and Analysis. *ACM Transactions on Intelligent Systems and Technology*, 3(2):25:1–25:28, February 2012.

[75] LIWC Inc. Linguistic Inquiry and Word Count. `http://www.liwc.net`.

[76] Diana MacLean, Sudheendra Hangal, Seng Keat Teh, Monica S. Lam, and Jeffrey Heer. Groups without tears: mining social topologies from email. In *Proceedings of IUI-2011*. ACM, 2011.

[77] Sylvain Malacria, Eric Lecolinet, and Yves Guiard. Clutch-free panning and integrated pan-zoom control on touch-sensitive surfaces: the Cyclostar approach. In *Proceedings of CHI '10*. ACM, 2010.

[78] Mirko Mandic and Andruid Kerne. Using intimacy, chronology and zooming to visualize rhythms in email experience. In *Proceedings of CHI '05 (extended abstracts)*. ACM, 2005.

[79] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schutze. *Introduction to Information Retrieval*, page 127. Cambridge University Press, 2008.

[80] Catherine C. Marshall. Rethinking Personal Digital Archiving parts 1 and 2. *D-Lib Magazine*, 4(3-4), 2008.

[81] Michael Massimi and Ronald M. Baecker. Dealing with death in design: developing systems for the bereaved. In *Proceedings of CHI '11*, pages 1001–1010. ACM, 2011.

[82] Viktor Mayer-Schnberger. *Delete: The Virtue of Forgetting in the Digital Age*. Princeton University Press, 2011.

[83] A. McCallum, X. Wang, and A. Corrada-Emmanuel. Topic and Role Discovery in Social Networks with Experiments on Enron and Academic Email. *Journal of Artificial Intelligence Research*, 30(1):249–272, 2007.

[84] Yusuf Mehdi. Facebook friends now fueling faster decisions on Bing. *Microsoft Bing Search Blog*, May 16, 2011.

[85] Malena Mesarina, Jhilmil Jain, Craig Sayers, Tyler Close, and John Recker. Evaluating a Personal Communication Tool: Sidebar. In *Proceedings of the 13th International Conference on Human – Computer Interaction. Part I: New Trends*, pages 490–499, Berlin, Heidelberg, 2009. Springer-Verlag.

[86] Meredith Ringel Morris and Eric Horvitz. SearchTogether: an interface for collaborative web search. In *Proceedings of UIST '07*. ACM, 2007.

[87] Meredith Ringel Morris, Jaime Teevan, and Steve Bush. Enhancing collaborative web search with personalization: groupization, smart splitting, and group hit-highlighting. In *Proceedings of CSCW '08*. ACM, 2008.

[88] Meredith Ringel Morris, Jaime Teevan, and Katrina Panovich. What do people ask their social networks, and why? A survey study of status message Q&A behavior. In *Proceedings of CHI '10*. ACM, 2010.

[89] Meredith Ringel Morris, Jaime Teevan, and Katrina Panovich. A Comparison of Information Seeking Using Search Engines and Social Networks. In *Proceedings of ICWSM '10*. AAAI, 2010.

[90] Abhinay Nagpal, Sudheendra Hangal, Rifat Reza Joyee, and Monica S. Lam. Friends, Romans, Countrymen: Lend me your URLs. Using social chatter to personalize web search. In *Proceedings of CSCW '12*. ACM, 2012.

[91] Carman Neustaedter, AJ Brush, Marc A. Smith, and Danyel Fisher. The social network and relationship finder: Social sorting for email triage. In *Proceedings of the Conference on Email and Anti-Spam '05*, 2005.

[92] The New York Times. The Palin E-Mails, 2011.

[93] M. E. J. Newman. Fast algorithm for detecting community structure in networks. Arxiv.org, September 2003.

[94] Nielsen. Social Media Report: Q3 2011.

[95] William Odom, Richard Banks, David Kirk, Richard Harper, Siân Lindley, and Abigail Sellen. Technology heirlooms? Considerations for passing down and inheriting digital materials. In *Proceedings of CHI '12*. ACM, 2012.

[96] William Odom, Abi Sellen, Richard Harper, and Eno Thereska. Lost in translation: understanding the possession of digital things in the cloud. In *Proceedings of CHI '12*. ACM, 2012.

[97] Gergely Palla, Imre Derenyi, Illes Farkas, and Tamas Vicsek. Uncovering the Overlapping Community Structure of Complex Networks in Nature and Society. *Nature*, 435(7043):814–818, June 2005.

[98] Eli Pariser. Beware online "filter bubbles". TED talks, March, 2011.

[99] S. Tejaswi Peesapati, Victoria Schwanda, Johnathon Schultz, Matt Lepage, Soyae Jeong, and Dan Cosley. Pensieve: supporting everyday reminiscence. In *Proceedings of CHI '10*. ACM, 2010.

[100] Adam Perer, Ben Shneiderman, and Douglas W. Oard. Using rhythms of relationships to understand e-mail archives. *Journal of the American Society for Information Science and Technology*, 57(14):1936–1948, 2006.

[101] Daniela Petrelli, Nicolas Villar, Vaiva Kalnikaite, Lina Dib, and Steve Whittaker. FM radio: family interplay with sonic mementos. In *Proceedings of CHI '10*. ACM, 2010.

[102] Daniela Petrelli and Steve Whittaker. Family memories in the home: contrasting physical and digital mementos. *Personal Ubiquitous Computing*, 14(2):153–169, February 2010.

[103] Piclens. http://cooliris.com.

[104] Chris Prom. Preserving email. *DPC Technology Watch Reports*, Dec. 1, 2011.

[105] Kristen Purcell. *Search and email still top the list of most popular online activities*. Pew Internet and American Life Project, 2011.

[106] Kristen Purcell, Joanna Brenner, and Lee Rainie. *Search Engine Use 2012*. Pew Internet and American Life Project, 2012.

[107] T. J. Purtell, Diana MacLean, Seng Keat Teh, Sudheendra Hangal, Monica S. Lam, and Jeffrey Heer. An Algorithm and Analysis of Social Topologies from Email and Photo Tags. In *Proceedings of the Fifth ACM Workshop on Social Network Mining and Analysis (SNAKDD)*. ACM, 2011.

[108] Radicati Group Inc. Email Statistics Report, 2012-2016, 2012.

[109] Radicati Group Inc. Email Market, 2012-2016, 2012.

[110] Filip Radlinski and Susan Dumais. Improving personalized web search using result diversification. In *Proceedings of SIGIR '06*. ACM, 2006.

[111] Leena Rao. Twitter Seeing 90 Million Tweets Per Day, 25 Percent Contain Links. *TechCrunch*, Sept. 14, 2010.

[112] Daniel Reisberg and Paula Hertel. *Memory and Emotion*. Oxford University Press, 2003.

[113] Bradley J. Rhodes. Margin notes: building a contextually aware associative memory. In *Proceedings of IUI '00*. ACM, 2000.

[114] David S.H. Rosenthal, Daniel C. Rosenthal, Ethan L. Miller, Ian F. Adams, Mark W. Storer, and Erez Zadok. The Economics of Long-Term Digital Storage. In *"Memory of the World in the Digital Age"*. UNESCO, Sept. 2012.

[115] Maayan Roth, Assaf Ben-David, David Deutscher, Guy Flysher, Ilan Horn, Ari Leichtberg, Naty Leiser, Yossi Matias, and Ron Merom. Suggesting friends using the implicit social graph. In *Proceedings of KDD '10*. ACM, 2010.

[116] Julia Schwarz, Jennifer Mankoff, and H. Scott Matthews. Reflections of everyday activities in spending data. In *Proceedings of CHI '09*. ACM, 2009.

[117] David Segal. The dirty little secrets of search. *The New York Times*, Feb. 13, 2011.

[118] David Segal. Muse to sift the emails of yesteryear. *New Scientist*, Oct. 15, 2011.

[119] Abigail J. Sellen and Steve Whittaker. Beyond total capture: a constructive critique of lifelogging. *Communications of the ACM*, 53, May 2010.

[120] M. G. Siegler. Zuckerberg: "Guess What? Nobody Wants To Make Lists", Aug. 26, 2010.

[121] Tom Simonite. Facebook App Reveals Your Social Cliques. MIT Technology Review, February 2011.

[122] Tom Simonite. Microsoft builds a Browser for your Past. March 15, 2012.

[123] Marc Smith, Vladimir Barash, Lise Getoor, and Hady W. Lauw. Leveraging social context for searching social media. In *Proceedings of the 2008 ACM workshop on search in social media*, SSM '08. ACM, 2008.

[124] Betsy Sparrow, Jenny Liu, and Daniel M. Wegner. Google Effects on Memory: Cognitive Consequences of Having Information at Our Fingertips. *Science*, 333(6043):776–778, 2011.

[125] Stanford NLP group. The Stanford Named Entity Recognizer. Stanford University web site.

[126] John Stasko, Carsten Gorg, and Zhicheng Liu. Jigsaw: supporting investigative analysis through interactive visualization. *Information Visualization*, 7:118–132, 2008.

[127] Danny Sullivan. Google: Bing Is Cheating, Copying Our Search Results. *Search Engine Land (blog)*, Feb. 1, 2011.

[128] Danny Sullivan. As deal with Twitter expires, Google realtime search goes offline. *Search Engine Land (blog)*, Jul. 4, 2011.

[129] Arun C. Surendran, John C. Platt, and Erin Renshaw. Automatic Discovery of Personal Topics To Organize Email. In *Proceedings of the Conference on Email and Anti-Spam '05*, 2005.

[130] Susan Manus. Why should we save our email? *The Signal, Library of Congress Digital Preservation Blog*, Sept. 8, 2011.

[131] Susan Thomas. Paradigm Academic Advisory Board Report. *John Rylands University Library, Manchester*, Dec. 12, 2005.

[132] Jaime Teevan, Susan T. Dumais, and Eric Horvitz. Personalizing search via automated analysis of interests and activities. In *Proceedings of SIGIR '05*. ACM, 2005.

[133] Jaime Teevan, Susan T. Dumais, and Daniel J. Liebling. A longitudinal study of how highlighting web content change affects people's web interactions. In *Proceedings of CHI '10*. ACM, 2010.

[134] Jaime Teevan, Susan T. Dumais, Daniel J. Liebling, and Richard L. Hughes. Changing how people view changes on the web. In *Proceedings of UIST '09*. ACM, 2009.

[135] Ray Tomlinson. The First Network Email? *bbn.com*, Date unknown.

[136] Endel Tulving. *Elements of Episodic Memory*. Oxford University Press, 1983.

[137] Twitter. `http://www.twitter.com/TopTweets`.

[138] Joshua R. Tyler and John C. Tang. When can I expect an email response? A study of rhythms in email usage. In *Proceedings of the European Conference on Computer-Supported Cooperative Work '03*. Kluwer Academic Publishers, 2003.

[139] J.R. Tyler, D.M. Wilkinson, and B.A. Huberman. E-mail as Spectroscopy: Automated Discovery of Community Structure within Organizations. *The Information Society*, 21(2):143–153, 2005.

[140] Emory University. Salman Rushdie Papers. 2010-2012.

[141] Elise van den Hoven, Corina Sas, and Steve Whittaker. Introduction to this Special Issue on Designing for Personal Memories: Past, Present, and Future. *Human Computer Interaction*, 27(1-2):1–12, 2012.

[142] Elise van der Hoven and Berry Eggen. Informing augmented memory system design through autobiographical memory theory. *Personal and Ubiquitous Computing*, 12(6):433–443, August 2008.

[143] Jessica E. Vascellaro. Why Email No Longer Rules... *The Wall Street Journal*, October 12, 2009.

[144] Fernanda B. Viégas, Danah Boyd, David H. Nguyen, Jeffrey Potter, and Judith Donath. Digital Artifacts for Remembering and Storytelling: PostHistory and Social Network Fragments. In *Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04) - Track 4 - Volume 4*. IEEE Computer Society, 2004.

[145] Fernanda B. Viégas, Scott Golder, and Judith Donath. Visualizing email content: portraying relationships from conversational histories. In *Proceedings of CHI '06*. ACM, 2006.

[146] Daniel M. Wegner. Transactive memory: A contemporary analysis of the group mind. In *B. Mullen and G. R. Goethals (Eds.), Theories of group behavior*, pages 185–208. Springer-Verlag, 1986.

[147] Steve Whittaker, Quentin Jones, Bonnie A. Nardi, Mike Creech, Loren Terveen, Ellen Isaacs, and John Hainsworth. ContactMap: Organizing Communication in a Social Desktop. *ACM TOCHI*, 11(4):445–471, 2004.

[148] Steve Whittaker, Vaiva Kalnikait, Daniela Petrelli, Abigail Sellen, Nicolas Villar, Ofer Bergman, Bar Ilan, Paul Clough, and Jens Brockmeier. Socio-technical Lifelogging: Deriving design principles for a future proof digital past. *Human-Computer Interaction*, 27(1-2):37–62, 2012.

[149] World Wide Web Consortium. W3C Contacts API Working Draft.

[150] Mic Wright. Why the British Library archived 40,000 emails from poet Wendy Cope. *Wired*, May 10, 2011.

[151] Ka-Ping Yee, Kirsten Swearingen, Kevin Li, and Marti Hearst. Faceted metadata for image search and browsing. In *Proceedings of CHI '03*. ACM, 2003.

[152] J. Zalinger, N. Freier, M. Freire, and B. Shneiderman. Reading Ben Shneiderman's Email: Identifying Narrative Elements in Email Archives. In *UMD Tech. report HCIL 2009-31*. University of Maryland, 2009.

[153] Jason Zalinger. *Gmail as storyworld: How technology shapes your life narrative*. Ph.D. dissertation, Rensselaer Polytechnic Institute, 2011.

[154] Kathryn Zickuhr. *Generations 2010*. Pew Internet and American Life Project, 2010.

[155] Sonia Zjawinski. British Library Puts Public's Emails on The Shelves. *Wired*, May 29, 2007.

Abbreviations used in this bibliography:

CHI	ACM SIGCHI Conference on Human Factors in Computing Systems
CSCW	ACM Conference on Computer Supported Cooperative Work
IUI	ACM International Conference on Intelligent User Interfaces
KDD	ACM SIGKDD International Conference on Knowledge Discovery in Data Mining
SIGIR	ACM SIGIR Conference on Research and Development in Information Retrieval
TOCHI	ACM Transactions on Computer-Human Interaction
UIST	ACM Symposium on User Interface Software and Technology
WSDM	ACM International Conference on Web Search and Data Mining
WWW	International Conference on World Wide Web