MailTronome: Visualizing the rhythm of social interactions through email communication patterns

Sanby Lee
Department of Computer Science
Stanford University, Stanford, CA 94305
byslee@stanford.edu

ABSTRACT
Email provides a written record of our social interactions. In this paper, we present *MailTronome*, a tool that visualizes the patterns of email communication between two people. We identify distinct patterns of email communication that become apparent when using our visualization tool, and we discuss the implications for how our work can be extended to use these patterns in designing better email interfaces for users.

Author Keywords
Email, data visualization, social interaction

ACM Classification Keywords
H.5 Information interfaces and presentation, H.4.3.c Electronic mail

INTRODUCTION
Many of us store our lives in our email. It contains a record of who we communicated with and what we talked about, and thus becomes a written record of social interactions. By analyzing our email, we can draw insights about the nature of those social interactions. In particular, we can categorize the "rhythm" of a relationship: the speed at which two people interact with each other, the frequency of their interactions, and the length of their interactions. One can imagine distinct patterns that correspond to different social situations: A quick back-and-forth of short exchanges to set up a meeting time and place; long and lengthy responses of two people discussing an article; a short burst of messages that occurs on a seasonal basis between two friends who live in different cities. Identifying and visualizing these patterns can provide a tool for users to understand their own relationships. In addition, once these patterns have been identified, they can be used as input in designing email interfaces to better suit the user.

In this paper, we present *MailTronome*, a tool to visualize the "rhythm" of a relationship between two people, based on their email communication patterns. (The name refers to the metronome-like quality of the visualization, in that it serves the same purpose of tracking a constant rhythm over time.) We show that distinct patterns do in fact occur, and are representative of the real-life relationship between two people. We also discuss implications for how this tool can be used in designing email interfaces, limitations of the current tool, and potential avenues for future work.

RELATED WORK
There is a sizable amount of previous work that has been conducted on the topic of how to visualize email communication in general. The previous work can be categorized into four broad areas, based on the purpose of the work and the type of email data that is visualized.

Enterprise Analytics. These projects are aimed at providing insights that enable users to answer their emails efficiently, particularly in a business context. Perer and Smith created several visualizations that showed enterprise users how effective they are at managing email, including (a) a treemap that organizes a user's contacts hierarchically, based on the domain name and suffix of their email address (b) a scatter plot that plots each contact based on the number of messages received from versus the number of messages sent to that contact (c) a chart that plots the number of conversations initiated by the user versus the number of conversations responded to by the user on a weekly basis [5]. Their user studies showed that the visualizations led to insights that helped users manage their workflow more effectively, such as identifying contacts for whom email was not an efficient way of getting responses. On the commercial side, Gmail Meter provides a monthly report that includes a breakdown of emails sent and received, length of emails, and response time, visualized using bar charts and time series charts [1]. Both of these projects are focused on visualizing statistics about the email messages themselves, and do not show connections between users.

Social Network Analysis. Another group of projects are aimed specifically at showing the network connections between users. Viegas et. al have created *Social Network Fragments*, a visualization that plots all of a user's contacts and draws connections between them based on the emails exchanged. They use a spring system algorithm to position contacts with strong bonds close to each other, and contacts with weak bonds far from each other [6]. *Immersion*, from the MIT Media Lab, creates a similar visualization, grouping contacts into clusters and drawing ties between all of them [2]. *My Map* takes a slightly different approach, by

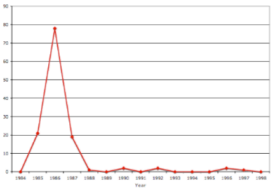

Figure 1. Visualization produced by Perer et. al to track volume of email with a contact over many years.

Figure 2. User interface of *PostHistory* created by Viegas et. al. Email volume is visualized as a heatmap in calendar view.

positioning all contacts in a ring, and drawing ties between them [3]. All of these projects rely solely on email metadata, such as the From, To, and timestamp fields, in order to determine the presence of a connection, and do not look at the contents of the message body. Viegas et. al and the *Immersion* team also note privacy concerns, which may have influenced the decision to look at metadata only.

Communication Patterns. The third group of projects focus on studying dyadic communication between the user and one other contact, rather than looking at the social network as a whole. The goal of these projects is to characterize the "rhythm" of communication between two people. Viegas et. al have created *PostHistory*, shown in Figure 2, which uses a calendar format to visualize the volume of email that a user sends or receives on any given day [6]. Days with high email traffic are represented as large squares on the calendar, while days with low email traffic are represented as smaller squares. Users can also filter the visualization by specific contacts. Perer et. al created time series plots of one user's email volume over many years, shown in Figure 1, with a separate plot for each contact [4]. They used k-means clustering to group the 76 most active relationships into 9 groups, but did not find any noticeable correlation with user-defined groups. Perer et. al also found that just 2% of the dyadic relationships accounted for 31% of all email volume, suggesting a power distribution of dyadic relationships. As in the Social Network Analysis category, both of these projects relied on email metadata to visualize communication patterns, and did not look at the content of the email.

Content Analysis. The final category of work does look at the content of emails in creating visualizations. Viegas et. al created *Themail*, which analyzes the email exchanges between the user and another contact, to find the most commonly used words by month and by year [7]. The words are plotted on a timeline by month, with the most common words for each month being listed at the top in larger and brighter font. Case studies with 16 users show that the users find these visualizations personally meaningful and start reminiscing, and mention that the visualization serves a purpose similar to that of a photo album.

In this paper, we have chosen to focus specifically on the topic of communication patterns. Previous work in this area has focused on using the volume of emails that are exchanged in order to determine the "rhythm" of the relationship between two users [4, 6]. However, volume of email is only one attribute, and it indicates only the intensity of the relationship, and may not capture other nuances. Our work aims to expand on previous work by examining more detailed attributes that categorize a chain of email communication: (a) the response time between each email (b) the length of each email. Furthermore, previous work visualized all email exchanges between two people as one unit. In my work, I plan to break out the messages originating from the user, versus the responses from the other contact, in order to analyze who is initiating and/or responding. Finally, previous work focused on the time span of years [4] or using the calendar [6]. However, this may not be the right resolution to understand relationships through email. In our work, we experiment with visualizing the data at the time level of weeks.

MAILTRONOME

Our tool creates one visualization for each relationship between the self and another contact. The visualization captures four attributes of the relationship: response time, message size, frequency of contact, and direction of contact. (See Figure 3 for reference.) Each message that is exchanged between the self and another contact is represented by one bubble. Messages that are received by the user are plotted above the line, and messages sent by the user are plotted below the line. This gives a visual indication of whether more inbound or outbound messages are occurring. The distance from the midline indicates the response time of each message. This provides a visual cue to show how quickly exchanges happen: messages that are exchanged quickly will be clustered together around the midline, whereas messages that are more delayed will have a lot of vertical distance between them. The plot area is divided into 52 vertical columns, each representing a week, and all of the messages that were exchanged within a given week are plotted within that column. This provides a visualization of how frequently contact occurs, and allows us to identify cyclical patterns. Finally, the message size is represented by the size of the bubble.

Methods

The current implementation of Mailtronome is based on one year of email messages sent to and from the author's personal Gmail account. In total, there were 5,058 messages exchanged with 462 different email addresses during the time period from January 1, 2012 to December 31, 2012. The messages were retrieved using Gmail API, which allows us to access the metadata and content for any email by querying based on message ID. The API returns message-level data which contains metadata fields such as To, From, Subject, and the byte size of the email. The data was then parsed using Python scripts. We calculated the attributes for each message using the methods detailed below.

Response Time. The metadata for each message contains a thread ID, which indicates the Gmail thread that this message belongs to. We grouped all messages into their respective threads, then calculated the response time of a given message by taking the difference between its timestamp and the timestamp of the most recent prior message. A message at the start of the thread was given a response time of 0. In order to visualize the data, the raw response time was indexed to the maximum response time, on a scale of 0 to 100%, with 100% being equivalent to the maximum response time. Indexing the response time allows us to easily compare response times between different contacts and automate the plotting of the visualization in d3. Preliminary analysis of the data revealed that email response times follow a power law distribution: 99% of all messages had a response time of within a week, but the remaining 1% of messages had response times ranging from several weeks to 3 months. As a result, because of a few outliers at the high end of response times, indexing directly off the raw maximum response time would visually compress the vast majority of the data, making it difficult to draw insight from the visualization. Therefore, we indexed response time based on the adjusted maximum response time, which is defined as one standard deviation above the mean. For this particular dataset, the adjusted maximum response time was 2.7 days, which covers 97% of all messages exchanged. Raw response times that were above the adjusted maximum response time were indexed at 100%. We chose not to normalize the response times further, since the power law distribution of response times is an important aspect of the data that we wanted to represent visually. For a given visualization, often there were many points that were close together in response time. Therefore, we set the opacity of each data point to 0.5, so that a darker region would provide a visual cue to indicate many data points in that region.

Message Size. The metadata for each message contains a field indicating the overall byte size of the message. We used this metadata to plot the size of each message. One issue to account for is that when replying to an email, often the writer will automatically include the previous email in the response, thereby artificially inflating the message size. We experimented with adjusting for this effect by calculating message size by taking the difference between the size of a given message and subtracting the size of the previous message in the thread. However, the message sizes resulting from this adjustment were unreliable, since many messages had a byte size smaller than the previous message, and this adjustment tended to overweight message sizes at the beginning of a thread. Therefore, we discarded this adjustment. As with the response times, we indexed the message size off the maximum message size, on a scale of 0 to 100%. The message sizes also followed a power law distribution, and therefore we also chose to index the message size off an adjusted maximum message size, rather than the raw maximum message size. We experimented with setting the adjusted maximum message size at one standard deviation above the mean. However, due to the extremely skewed distribution of message sizes, this was still too large to effectively visualize message sizes (~458 KB compared to 90% of messages that were under 25 KB). Therefore, we manually set the adjusted maximum message size at 50 KB, which covers 95% of all email messages.

Contact and Direction. For each message, we examined the To and From fields to determine whether the message was sent to the user or from the user, and which contact the message was exchanged with. We also manually de-duped contacts that used several email addresses. For example, messages exchanged with the same contact from both niki@rogue.com and niki@division.com would be grouped together. The names were then anonymized to protect contacts' identities or email addresses from being displayed. One nuance in parsing the To/From fields is that roughly 10% of the messages were sent to a group of contacts. For example, the user might email several contacts about an upcoming event, and one of the contacts would then respond to the user, and include all of the other contacts in the email. In these cases, a message sent to several contacts would be plotted on the respective visualization graphs of all contacts. In future iterations of our visualization tool, we could include a feature that allows users to distinguish between group messages versus one-on-one messages.

Week Number. Our tool visualizes email flow on a weekly basis. In order to plot each message, we mapped the timestamp of the message to a week number (1 through 52) corresponding to the week in 2012 in which it was sent. For example, an email received on January 12, 2012 would be plotted in week 2.

Out of 5,058 messages exchanged during a one-year period, we discarded 90 messages that were missing a timestamp, and 64 messages that were missing a message size. We restricted our final dataset to those contacts with whom at least 25 messages had been exchanged over the course of a year. Those contacts accounted for roughly 80% of all email traffic. Once again, we found that the distribution of email traffic followed a power law distribution: A small number of contacts were responsible for the majority of email traffic, and the vast majority of contacts only had 1 or 2 email exchanges. Our final dataset included 17 contacts and 4,025 messages.

The final dataset was then exported to a JSON file and plotted using d3 and JavaScript to produce the interactive visualization.

Since the entire visualization process is automated using Python scripts and d3, a personalized visualization can be created for any end user, given a set of email credentials that is authorized through the Gmail API. Due to issues of privacy and security, the current implementation of *MailTronome* does not include an option to run the visualization with datasets from other users. We discuss these implementation issues further at the end of this paper.

User Interface

The user interface consists of two panes. On the left, contacts are listed in descending order of volume of email exchanges. On the right, the visualization for a given contact is shown. The user can flip through the visualization for different contacts by clicking the name of the contact in the left panel.

In order to keep the interface minimal, no labels are shown on the initial page load. This gives the user a chance to explore the different visualizations, interpreting the different patterns based purely on their visual character, before assigning meaning to a pattern. The interface includes a button to "Show Labels." By clicking this button, the user can turn on axes labels and visual guides to help interpret the data. In this way, we give the user a choice of when and how to engage with the visualization as a data analysis tool.

On initial page load, before any particular contact is selected, all of the email messages across all contacts are shown. Mousing over a contact name highlights all of the bubbles corresponding to that contact. This allows the user to compare and contrast patterns from different contacts, and to put the pattern of a contact in overall context.

Once the user is in the visualization for a specific contact, mousing over any particular bubble displays detailed info for that message.

RESULTS

The visualizations produced by our tool show that there are in fact distinct patterns of communication that can be identified when comparing and contrasting the "rhythm" of relationships with different contacts. Furthermore, these patterns do in fact correspond to the real-life relationships between the user and the contact.

Figure 3 shows email communication between the user and "Mindy." We see a pattern of constant communication, with multiple instances of contact occurring almost every week. In addition, many of the messages are relatively large. In real life, Mindy is in fact a close friend of the user. For comparison, Figure 4 shows email communication between the user and "Alex." The frequency of communication is also constant and weekly, but the size of the messages is much smaller, and the messages are much closer together in response time. Alex is also a close friend, but the style of communication is markedly different than that of Mindy's.

A third pattern occurs with "Ann" (Figure 5). The user and Ann exchange a number of messages back-and-forth, but contact occurs cyclically, at the frequency of roughly 2-3 months. In real life, Ann is a close friend who lives in a different city. Therefore, one hypothesis is that contact is centered around events that occur seasonally, such as the winter holidays, Thanksgiving, spring break, and summer holidays. Note that this pattern of contact is visually apparent only by plotting data at the week level as we have done here; the cyclical pattern would be hidden in a

Figure 3. User interface of *MailTronome* and visualization of email communication with "Mindy," a close friend.

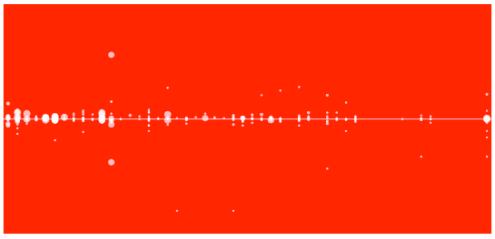

Figure 4. Communication with "Alex," another close friend.

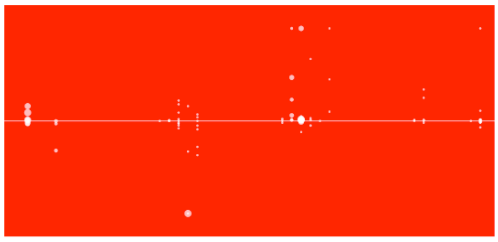

Figure 5. Communication with "Ann," an old friend who lives in a different city.

Figure 6. Email sent across a mailing list.

visualization at the year level. In addition, plotting response time as an attribute enables us to see the back-and-forth responses and visualize the pattern of communication, rather than looking only at the volume of communication.

Figure 6 shows the visualization for "hb-list," which is a mailing list that the user is subscribed to. We observe that the mailing list has a very distinct pattern: there are a much greater number of inbound messages, and the user rarely

sends an outbound message. Furthermore, responses are heavily clustered toward the midline, indicating that many responses to a thread occur quickly, and peter out quickly, with very few responses coming in after about half a day. In addition, the darkened opacity of the initial cluster of responses indicates that many threads follow this pattern.

Based on the sample in our current implementation of *MailTronome*, we can loosely categorize communication patterns into several types: (a) Mailing list (b) Frequent contact with mixed response times (c) Frequent contact with quick, short exchanges (d) Cyclical contact with bursts of messages (e) Sporadic contact with short messages and long response times. In our conclusion, we discuss ways to further refine how we define these categories.

DISCUSSION

Our results show that visualizing email communication at the weekly level, rather than at the yearly or day level, produces meaningful patterns that can be used to draw insight about the "rhythm" of a relationship between two people. In addition, expanding the dimensions of the visualization beyond just the volume of email, to include additional attributes of each message such as response time, message size, and message direction, further illuminates these patterns. We have visually identified distinct clusters of patterns that can be used to categorize different types of relationships. If automated, the identification of these clusters can be used to optimize the email interface for individual users.

One limitation of our work is that it shows a relatively static relationship between the user and each contact. In other words, while distinct patterns of communication can be identified, those patterns do not meaningfully change over the course of a year. This is important to note because one of the original motivations for our work was to identify patterns that predict a decline or increase in the volume of communication. For example, the user exchanged over 300 messages with "Grace" in 2012, but none in 2016. What happened in those intervening years? Our current visualization does not provide data to answer this question. Expanding the dataset would help illuminate this further.

In addition, one hypothesis was that increasing response times over time would predict the decline of a relationship. One would be able to visually observe the increasing distance between two people, as the closest bubbles get farther and farther away from the midline. However, the current format of the visualization does not support this hypothesis. Even in relationships with relatively low volume of contact, the response times still cluster quite closely around the midline. Therefore, response time may not be the best indicator of increasing distance in a relationship. A revised hypothesis is that frequency may be a better indicator, as message initiation becomes farther and farther apart, visually represented as horizontal spaces which are empty in weeks with no contact. Given the importance of frequency, there may be other visualization formats which visually prioritize frequency in order to facilitate better analysis.

A final consideration is that originally, the visualization was created using one year of email data from May 1, 2015 to April 30, 2016. However, this dataset did not yield enough data points to observe meaningful patterns. One hypothesis for the decrease in email communication is that communication has moved to other platforms. Over time, as more digital communication platforms proliferate and gain adoption, the communication that used to take place over email is now becoming distributed through multiple platforms, such as SMS, WhatsApp, Facebook Messenger. Email still occupies an important place in the user's range of communication platforms, as there are certain types of conversations that are unwieldy to have over SMS or messaging platforms, and the user interfaces for current mail clients are optimized for archival and efficient search/retrieval. However, visualizing a given user's communication over other platforms in addition to email would provide a fuller picture of the user's relationships, possibly leading to further insight.

FUTURE WORK

We identify two potential avenues for future work. First, we can extend our system to automatically assign contacts to a category, based on the visual pattern of communication. In the research done by Perer et. al, the authors first plotted email relationships as a time series graph, then used k-means clustering to group patterns into distinct clusters. They also created the ability for users to search for matches to a visually drawn time series line [4]. We can envision using the same approach with our visualization tool and dataset. We can use a clustering algorithm to systematically identify the patterns we found visually in our results, then automatically assign any given contact to a cluster. Given the ability to classify any given contact into a communication cluster based on their past email patterns, there are many real-world implications for how email interfaces can be designed. For example, the interface might prioritize a frequent contact in deciding when to surface a message from that contact. Alternatively, the interface might choose to prioritize a message from someone who the user previously had very frequent contact with, but has since fallen out of touch with, in order to bring an element of surprise to the user interaction. As another example, we might envision an inbox where incoming mail is sorted by communication pattern, e.g. Frequent, Cyclical, Short Exchanges, rather than Important, Default, Spam, Commercial as they often are now.

A second avenue for extending our system is automating the visualization process for all users. In our current implementation, the visualization is produced through several steps: (a) Authorize the Gmail API to make requests to the user's inbox (b) Make requests and download data (c) Parse the data (d) Manually link contacts who email from several different addresses (e) Anonymize the names based

on a list provided by the user (f) Export data to JSON and plot using d3. The majority of the process is automated; however, (d) and (e) require user input, and are currently done by requiring the user to manually create a CSV that maps email addresses to anonymized names. Extending the system would involve building a GUI to allow the user to easily link contacts, and assign them names or auto-assign randomly generated names. In addition, a simpler authorization flow would be built which allows end users to authorize the Gmail API through the web interface, rather than running a Python script. Finally, we would address issues of privacy concerns, by building the system so that email data is processed online and only stored temporarily, and is never stored on any external servers.

CONCLUSION

We have created a tool that analyzes email data to visualize the "rhythm" of relationships between two people. These visualizations demonstrate distinct patterns of communication that can be grouped into categories. We envision our future work as focusing on ways to automate this visualization process for all users and automatically assign contacts to communication clusters, with the potential to help optimize the design of email interfaces.

REFERENCES

1. Gmail Meter. http://www.gmailmeter.com/
2. *Immersion*. MIT Media Lab. https://immersion.media.mit.edu/
3. *My Map* (A Self-Portrait). http://christopherbaker.net/projects/mymap/
4. Perer, A., Shneiderman, B., & Oard, D. Using rhythms of relationships to understand email archives. Paper presented at the 22nd Annual Symposium of the Human-Computer Interaction Laboratory, University of Maryland, College Park, MD (2005).
5. Perer, A., & Smith, M. Contrasting portraits of email practices: visual approaches to reflection and analysis. In *AVI*, ACM Press (2006).
6. Viegas, F., boyd, d., Nguyen, D., Potter, J., & Donath, J. Digital artifacts for remembering and storytelling: *PostHistory* and *Social Network Fragments*. In Proceedings of the 37th Hawaii International Conference on System Sciences (2004).
7. Viegas, F., Golder, S., & Donath, J. Visualizing email content: Portraying relationships from conversational histories. In *CHI*, ACM Press (2006).