Understanding Implicit Social Context in Electronic Communication

by Andrea Lyn Lockerd

B.S. Electrical and Computer Engineering
University of Texas at Austin (1999)

Submitted to the Program in Media Arts and Sciences
School of Architecture and Planning

In partial fulfillment of the requirements for the degree of
Master of Science in Media Arts and Sciences

At the Massachusetts Institute of Technology
September 2002

Signature of Author ______________________________

Program in Media Arts and Sciences
August 9, 2002

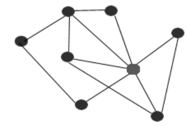
Certified by ______________________________

Ted Selker
Associate Professor
Thesis Supervisor

Accepted by ______________________________

Andrew Lippman
Chairperson
Department Committee on Graduate Students

Understanding Implicit Social Context in Electronic Communication



by Andrea Lyn Lockerd

Submitted to the Program in Media Arts and Sciences
School of Architecture and Planning

In partial fulfillment of the requirements for the degree of
Master of Science in Media Arts and Sciences

Abstract

Artificial Intelligence (AI) has shown competence in helping people with complex cognitive decisions like air traffic control and playing chess. The goal of this work is to demonstrate that AI can help people with social decisions. In this work Artificial Intelligence of Social Networks is used to improve human-human communication, recognizing the social characteristics of human relations in order to achieve a more natural online communication interface. Can a computer learn to understand the value of communication? It is shown here that a first attempt at social context classification performs with almost 70% reliability. Could a computer use this to help a person relate to other people through technology? The addition of social context to an email interface is shown to have a positive effect in a user's online communication behavior.

Email is a tool that people use practically every day, making an implicit statement about their relationships with other people, and providing an opportunity for a computer to learn about their social network. Furthermore, over the years people have come to utilize and depend on email more in their daily lives, but the tool has hardly changed to help people deal with the overwhelming amount of information. Many of the social cues that allow people to naturally function with their social network are not inherent or obvious in Computer Mediated Communication (CMC). This work offers automatic social network analysis as a means to bring these cues to CMC and to foster the user's coherent understanding of the people and resources of their communication network.

Thesis Supervisor: Ted Selker
Associate Professor, MIT Program in Media Arts and Sciences

The author thanks the Eircom Fellows Program for their support of her research.

Understanding Implicit Social Context in Electronic Communication

by Andrea Lyn Lockerd

Thesis Reader ____

Henry Lieberman
Principal Research Scientist
MIT Media Laboratory

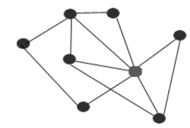
Understanding Implicit Social Context in Electronic Communication

by Andrea Lyn Lockerd

Thesis Reader ____

Ronald E. Rice
Professor, Chair of Department of Communication
School of Communication, Information & Library Studies
Rutgers University

Acknowledgements



This thesis is dedicated to my personal social network! ☺ There are so many people that I'd like to thank for helping me get this far…

Ted Selker, for giving me the opportunity to be a part of his research group, believing in my work, and simultaneously being my biggest cheerleader and my biggest critic!

Henry Lieberman, for being a role model in this academic community and for his time and thoughtful conversations that have guided this work.

Ronald Rice, for being such an enthusiastic member of my committee, and helping me learn about the Social Networks community and its exciting opportunities for computer scientists.

The graduate students of Context-Aware Computing, **Ernesto Arroyo, Win Burleson, Matt Mankins, Jorge Martinez,** and **Taly Sharon**, for the countless conversations that have helped mold my direction over the past two years, and the friendship that makes work fun! Thanks especially to Matt for his knowledge and creativity in email and networking, and Win for his newfound expertise in experimental design.

All the UROPs that work with our group, specifically **Chris Lucas**, my super-UROP, whose work on this project in particular was invaluable!

Susan Knightly, the friendly face that keeps our group together.

Network Computing Systems, necsys, especially **Elizabeth Harvey-Forsythe**, for helping me make my email system a reality in building E15.

All my new found friends in my home away from home, for the laughs and good times that make the world go round: **Rachel Bernstein, Heather Ferricks, Surj Patel, Jim McBride**, and of course who could forget the adventure team – **Byron Stigge, Jeff Huang,** and **Florian 'Floyd' Mueller**.

My friends from Texas, **Susan Kellond, Jenny Sunker, Meredith Anderson, Braxton Thomason**, for showing my over and over through the years what true friendship really means.

Edison Thomaz Junior, meu ursinho ganso bobo du corezon, for being a beautiful person that brings harmony to my life.

My family, **Clark, Jan, Scott, Patrick,** and **little Noah Lockerd**, for their love and support through the years of living, learning, and growing together as a family, I love you all!

Contents

Figures

Tables

1. Introduction

Can a computer learn to understand the value of communication? If it did, could it use this to help a person relate to other people through technology? This work is an attempt at using Artificial Intelligence (AI) about Social Networks to improve human-human communication, recognizing social characteristics of human relations in order to design a more natural online communication interface.

The medium of discussion here is email for two reasons.

1) Email is a tool that people use practically every day, and in this usage they make an implicit statement about their relationships with other people. This provides a unique opportunity for a computer to model some aspects of a user's social network.
2) The way people use email and the information that it presents hasn't changed significantly since the 1970s, even though demands have grown as it has become the most widely used internet application [Nielsen].

Computer scientists first started using a program called MAILBOX to swap messages on the Compatible Time-Sharing System at MIT in the 1960s. Then in 1971, Ray Tomlinson developed the first email application for ARPANET, SNDMSG and READMAIL. "Mail spooled out like a teletype printout". In 1975, MSG, written by John Vittal, can fairly be called the first modern email program, with a significant amount of the functionality available in email clients today. Some features include: forwarding messages, filing messages into folders, and sorting the display of messages by header information like date or sender, and automatic addressing of replies, cc, bcc [Stewart].

> "It soon became obvious that the ARPANET was becoming a human-communication medium with very important advantages over normal U.S. mail and over telephone calls." – J.C.R. [Licklider]

Years later, almost 30 billion emails are sent everyday (according to the International Data Corporation), and the tool has hardly changed in its ability to help people deal with such an overwhelming amount of information. Technology should do better than this!

This thesis is motivated in part by the following scenario: If you walk into a meeting or a party or some physical place with a number of people, you instantly scan the room to see who is there. You automatically make mental notes like “oh I haven't seen that person in a couple weeks”, “I just saw this person”, or ”there's a friend talking to someone I haven't met”. All of this helps you make an agenda of how you organize yourself to approach the event and the various people there, and is an example of how people automatically use social network analysis in face-to-face interactions.

1.1 Approach

Many of the social cues that allow people to naturally function with their social network in the above scenario are not inherent or obvious in CMC, which therefore obfuscates the maintenance and utilization of ones’ social network online. This work submits that computers should perform automatic social network analysis in order to bring these cues to CMC and to foster the user’s coherent understanding of the people and resources of their communication network.

A person’s social network consists of a set of people (nodes) with whom they have ties, connections between the nodes, and resources that are exchanged between the nodes. These resources can be information, influence, emotional support, and confidence, just to name a few. Here, the term social resources will mean any resources exchanged between two people in the social network that has some social significance (solidarity, antagonism, agreement/disagreement, etc.).

1.2 Automatic Social Network Analysis

This work does not attempt to completely analyze of all aspects of a personal social network, but rather to collect those aspects that are particularly relevant to enhancing an online communication interface.

There are a few concrete things that are easy for the computer to collect: structure (who's connected to whom from email traffic), frequency of contact, symmetry of contact, response times, time spent composing messages in the client, time spent reading messages in the client. The harder problem remains: what kinds of social resources are exchanged between the people in the user's personal social network?

AI can be the solution; a computer program that recognizes the social context of a message (i.e. *informing, inquiring, sharing, planning, intimate, etc.*) is in a better position to determine the value of that communication. It is unreasonable to expect that a machine will come to be perfect in this respect, but the stance of this research asks, given an imperfect model of social context, can this be used to enhance an online communication interface.

A number of AI techniques could attempt such a classification problem; I chose to try the supervised learning approach, using Support Vector Machines (SVMs). The reasons for doing so will be discussed in a later chapter. The steps then include: get a data corpus of email labeled with the social context classes (*informing, inquiring, intimate, planning, ...*) to use as training examples for the pattern recognition; then let the algorithm learn to discriminate between the classes of email based on various concrete features that it parses out of an email message (*length, emoticons, punctuation, ...*).

As a quick example, here is how the model for *informing* email is built:

1) For every message in the training corpus where informing = true.
2) Parse the message into a feature set (*word counts, punctuation, etc.*).
3) Give these input/output pairs to the algorithm as positive examples.
4) Repeat steps 1-3 for the negative examples.

Once the computer has a statistical model of what *informing* is in terms of email features, it can classify a new email in the following way: a new email comes in, parse it into its feature set (*word counts, punctuation, length, etc.*), give this feature set to the *informing* model, and the model returns the likelihood that this new email is *informing*.

One of the major components of this project is a Social Network Server, the SocNetServer. It is the implementation of this automatic social network analysis:

- It compiles personal social network information for a user based on email interactions (who they communicate with, frequency, symmetry, response times).
- It has statistical models of social context of email (the SVMs described above).
- It has an XML-RPC interface allowing clients to connect to it and ask for social network information about a user.

1.3 Social Context in an Email Interface

The second question of this research addresses how this automatic social network analysis, embodied in the SocNetServer, should be used to increase the user's understanding of their communication network and enhance their experience communicating online. The other major component of this work is the DriftCatcher email client, which helps the user catch the drift of what is happening with their personal communication network. It is an example of an application, built to utilize the artificial intelligence of the SocNetServer, with the goal of helping users understand and maintain their social network more naturally.

- DriftCatcher lets you see email in more than just a temporal context.
- It adds social context cues based on statistical content models and observations of the user's past behavior.
- It completes the loop by sending informing about user behavior with their network back to the SocNetServer.

1.4 Contributions

The main contributions of this thesis are the following:

1) Using AI to augment a human-human communication medium: the automatic personal network analysis of the SocNetServer informs the interface of the DriftCatcher email client to improve the way a user is able to mind their relationships.
2) Classification of social resources in email using machine learning techniques
3) The evaluation of 1 and 2.

1.5 Thesis Roadmap

Chapter 2 presents **example scenarios** of the DriftCatcher/SocNetServer system.
Chapter 3 is a brief overview of relevant **theories** and prior work.
Chapter 4 details the **design and implementation** of SocNetServer/DriftCatcher.
Chapter 5 covers the **evaluation** of the machine learning and the email client.
Chapter 6 sums up the **contributions** of this work and suggests **future work**.

2. Application Scenario

This section goes through a few examples with fictitious characters to illustrate how users interact with and benefit from the DriftCatcher email client enabled by the SocNetServer.

2.1 Meet Lori Adler

Dr. Lori Adler is a Research Staff member of the Context-Aware Computing group at the MIT Media Lab. Lori uses email regularly, and uses it to communicate with people from all facets of her life. Like many others, Lori finds that a large part of her day is spent doing “social network maintenance”: building, managing, and keeping track of various social and business relationships. Moreover, she does a large portion this maintenance over email. Recently she started using a DriftCatcher email client powered by a SocNetServer, and has found that it helps her prioritize her email tasks and have a better understanding of her personal social network.

2.2 Early for a Meeting

Traffic was light this morning, so Lori arrives 15 minutes early for her morning meeting. Having time to check her email quickly, she opens her inbox to find 10 new messages (Figure 2.1). Using the DriftCatcher CompTime feature (which shows the average time she spends composing messages to the various senders), she is able to prioritize the messages based on how much time they are likely to take her to deal with. Looking at the time bar length indicating the average time Lori takes to compose messages to the various senders (between 0 and 30 minutes), she quickly selects and responds to message numbers 6 and 9 in plenty of time for the meeting.

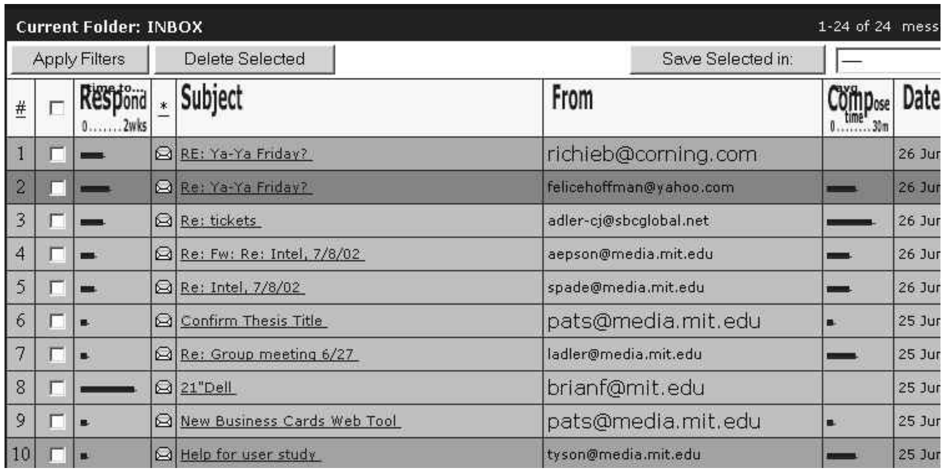


Figure 2.1: Average Compose Time feature

2.3 Reciprocating Response Time

Lori has various response patterns with people in her social network. Her friend Peter usually responds within a few days, but her colleague Andy usually responds within a few hours. She would like to reciprocate these response patterns, and the DriftCatcher client helps her do so with the ResponseTime bar. The time bar length indicates the time that Lori has left to respond to the messages (from 0 from 2 weeks). The time allotted for her reply is based on the response pattern of the sender. Figure 2.2 below shows that Lori has longer to respond to Peter (message 2) than to Andy (message 3).

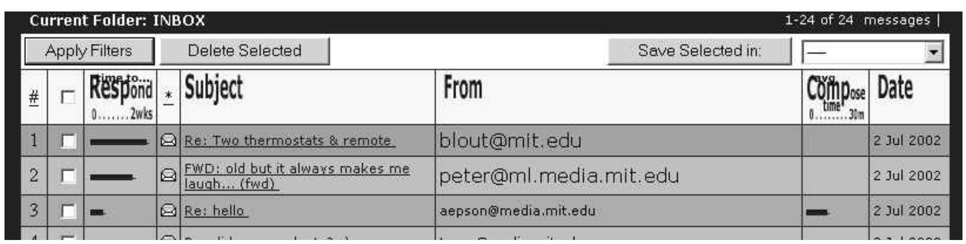


Figure 2.2: Response Time feature

2.4 Visualizing Closeness

Lori opens her email and notices that of her first six new messages only one is from someone she communicates with frequently. As shown in Figure 2.3 below, the DriftCatcher client portrays the symmetry and frequency of communication in the font size of the sender's name. This lets Lori easily distinguish frequent versus infrequent relations. In the figure below, Lori can quickly see that **teller@media** is a more frequent contact than **mres@media**.

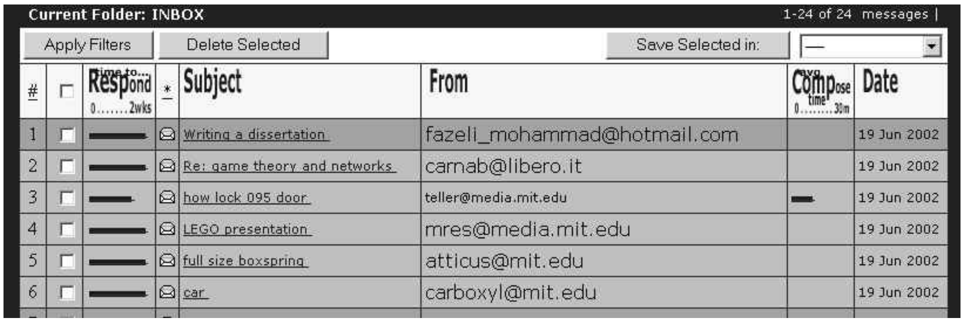


Figure 2.3: Frequency of Contact

2.5 Visualizing Context

When Lori is trying to decide which messages are most important, sometimes the subject line is not enough information to determine the social intention of the sender. The DriftCatcher client helps her by color-coding the messages according to their social context. In the figure below, Lori is able to see quickly that most of the messages are informing, but message 1 and message 12 involve planning and message 10 is an inquiry.

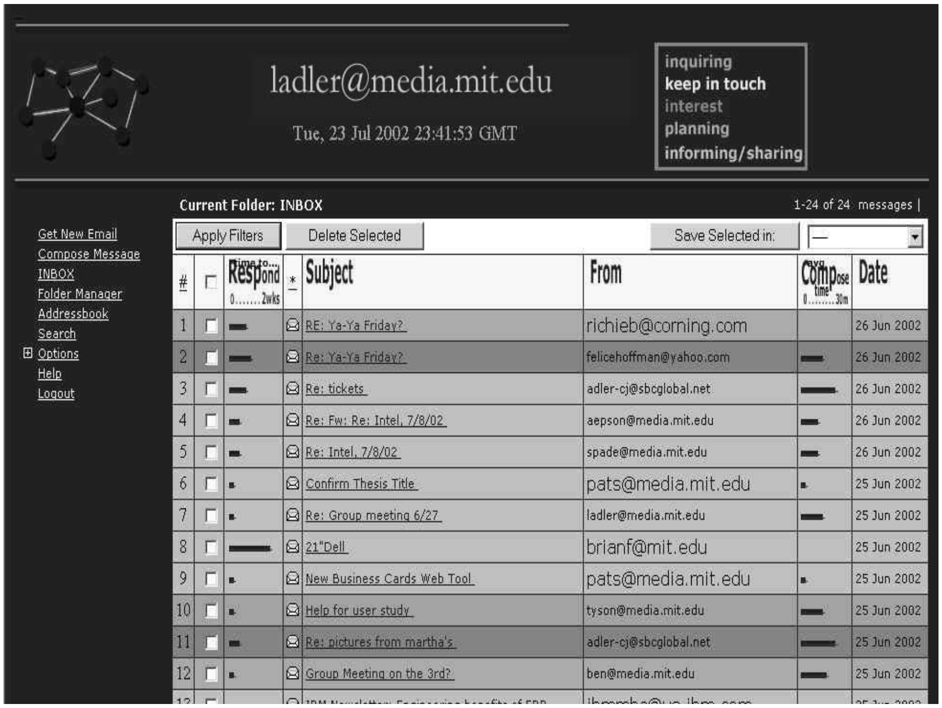


Figure 2.4: Message Context

2.6 The Administrative Assistant

David works as a temp, today is his first day on the job at the Media Lab and he is assigned to sit in for the administrative assistant of the Context-Aware Computing group. Lori Adler is having a busy day and hasn't been able to check her email, but she will have some time in a few minutes once she gets out of a meeting. David is asked to look through her inbox and find a couple of emails that she should deal with then. Viewing her email with the DriftCatcher email client makes it easier for David to step into the social context of Lori's inbox. The name sizes let David know who Lori corresponds with frequently; he looks at these first. The ResponseTime bar lets David choose messages that are likely to be more urgent than the others, and the color-coding indicates the intension of the message so he doesn't pass up a message trying to plan a meeting for later this afternoon.

3. Background

The theory and rational of this work stems mainly from three fields: Social Network Analysis, Machine Learning, and Human-Computer Interaction Design. This chapter goes through the features of these three fields that directly impact or motivate this work.

3.1 SNA and CMC

Social Network Analysis (SNA) is the study of various aspects of the structure and behavior of social networks. A person's social network consists of a set of people (nodes) with whom they have ties, connections between the nodes, and resources that are exchanged between the nodes. These resources can be information, influence, emotional support, and confidence, to name a few. This work, while not a complete social network analysis, attempts to utilize the theories and findings of social networks as means to improve an online communication interface. A couple of theories most relevant to the information collected by the SocNetServer include: *social capital* [Lin], the amount of support (of all forms) which can be called upon from the people in your social network, and *strength of weak ties* [Granovetter], a group of studies which indicate that the people most important to you in terms of access to information and resources are on the outskirts of your social network.

Computer Mediated Communication (CMC) is a field that studies and builds systems that allow people to communicate through technology; email, instant messaging, and video conferencing are a few examples of CMC. Over the past decade, social network scientists have grown interested in computer networks and to what extent CMC influences social networks. For example, computer networks are especially suited for the maintenance of relationships between people who cannot meet frequently; therefore, de-emphasizing the need for locality in both work and community structure [Wellman].

Measuring Social Resources in CMC

Interaction Process Analysis is an analysis scheme commonly used in studies of small groups [Bales]. It classifies human-human interaction related to group dynamics (in face-to-face interactions). Bales' IPA describes a socioemotional interaction as one that shows solidarity, antagonism, tension, agreement, or disagreement, and a task-oriented interaction involves giving or receiving opinions, information or orientation (see table 3.1).

SOCIOEMOTIONAL		TASK-ORIENTED	
POSITIVE	NEGATIVE	GIVING	RECEIVING
Solidarity	Antagonism	Suggestion	Suggestion
Agreement	Disagreement	Opinions	Opinions
Releasing Tension	Showing Tension	Orientation	Orientation

Table 3.1: The breakdown of Bales IPA.

In this work, the term social resources will mean any resources exchanged between two people in the social network that has some social significance, covering the whole spectrum of Bales' IPA. It was not always obvious that the whole range of Bales' IPA can be expressed in email. Some hypothesized that the text-based medium of email would be too constraining to afford the exchange of socioemotional information.

A few people addressed the extent to which socioemotional content is contained in email. In one study, over 2000 email sentences were labeled, by hand, using a slightly modified version of the Bales IPA categories. They showed that CMC does afford the exchange of socioemotional content, and in particular 30% of sentences in their dataset were of a socioemotional nature [Rice]. Another study addressed the existence of social context cues in electronic communication, and discusses how relational cues from face to face communication are translated to text based communication. They found, for example, that when communicating over email a person tends to replace a head-nod indicating agreement with a verbal phrase like 'I definitely agree…' [Walther].

Applications of SNA in CMC

The work of Bonnie Nardi strongly motivates systems, like SocNetServer and DriftCatcher, which integrate social network analysis with computer-mediated communication. The NetWORKing ethnographic study looked at how people utilize social networks in the workplace and concluded that success in today's distributed business environment is increasingly dependent on the ability to manage one's social network. They argue that "netWORKing" (the process of building, maintaining, and activating your social network) is an absolute necessity in the modern work environment [Nardi].

There have been systems with aspects of social network analysis applied to computer applications:

- The Referral Web system [Kautz], finds a path between two people in a social structure using a closeness metric based on web documents.
- Yenta [Foner] is a multi-agent system for matchmaking, based on subject matter of email messages to suggest matches between users.
- ExpertFinder [Vivacqua] is an agent system that helps people find an expert to help them in a Java Programming domain.
- [Flores] is a speech-act application that tries to identify patterns of speech in an organization related to the action that speech tends to induce.

There are two main qualities that differentiate the work here. Using a personal network approach; rather than take the point of view of a whole organization or community this work understands a social network from the point of view of a single user. Secondly, most of the current applications of social networks and online communication deal with information flow and task-oriented resources. The SocNetServer attempts to recognize all of the social resources exchanged between people in the network in order to better characterize relationships automatically.

3.2 Machine Learning

The field of Artificial Intelligence (AI) attempts to understand and build intelligent entities. There is a range of motivations for the people in this field. Some are motivated by the philosophical challenge of achieving a better understanding of human intelligence. Others are motivated by the sheer engineering challenge of building systems that behave intelligently. In this research and others, it is a practical challenge; the motivation is simply that intelligent systems will be easier for people to use [Russell].

This research concerns using artificial intelligence to augment a user's ability to make decisions and perform a task. Specifically, the challenge is that of having a machine understand the social implications of electronic communication in order to augment the user's ability to manage their relationships online.

Some AI work that is most relevant to this research is that of social intelligence, an example of which is Kismet [Breazeal], a robot, which recognizes body language and verbal tone and responds with appropriate facial expressions, to have meaningful social exchanges with humans. Essentially this work contends that computers should model and understand the implicit social context of human behavior in order to afford a more natural interaction. In the context of this work, an email system is in a better position to understand how it should behave if it has some understanding the social intensions and implications of the messages it handles.

Supervised and Unsupervised Learning

Machine Learning is the study of computer algorithms that improve automatically through experience [Mitchell]. There are two basic divisions of machine learning: supervised and unsupervised learning. Techniques that group instances without a pre-specified label are called unsupervised; for example, clustering algorithms. A technique is considered supervised when the algorithm learns the relationship between independent attributes based on a designated dependent attribute (the label). These systems, are trained by a set of examples, and learn how to behave from a set of input/output pairs. Supervised learning can be interpreted as the regression problem of approximating, from sparse data, a multivariate function.

Machine learning techniques are especially appropriate for problems that are perceptual and hard to explicitly program into a machine. For example, computer vision and speech recognition in which it is hard to explain the underlying behavior of why we behave the way we do. On this note, social relations and interpreting social context is decidedly perceptual and relatively hard to describe in certain terms; hence, the motivation for trying a machine learning approach to the problem of classifying social context over other AI techniques.

Support Vector Machines

One supervised learning technique is Support Vector Machines (SVMs), which was first introduced by [Vapnik]. Figure 3.1 shows the basic concept of SVMs: the algorithm learns a threshold value that maximally separates two classes of data in a feature space. The most basic model uses a linear threshold function, but SVMs can also be made to handle classification in which there is no linear separation of classes by specifying a different function with which to try to separate the data. Typical non-linear mappings include: a polynomial kernel, the radial basis function kernel, and the sigmoid kernel [Witten].

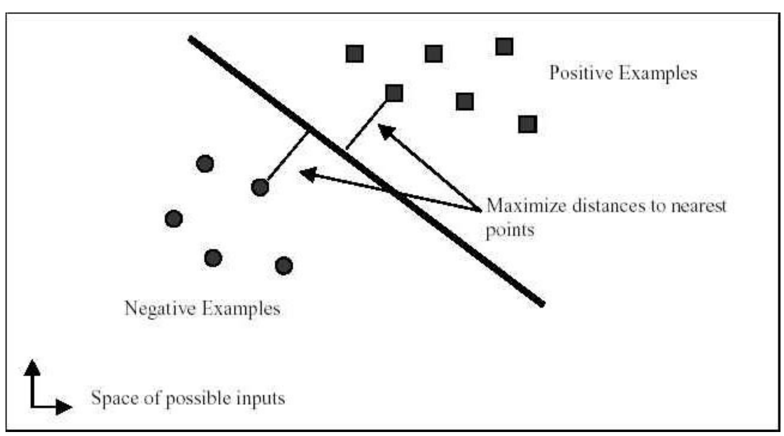


Figure 3.1: A basic linear model Support Vector Machine

It has been shown that some text classification problems are separable using SVMs. [Joachims] successfully used SVMs to classify the Ohsumed dataset and a Reuters dataset by topic categories, and was able to do so with less effort than with other classification methods. In many ways, any machine learning technique could be framed to handle this classification problem, but there are features of SVMs that make them a good candidate for the problem. SVMs work well in a high dimensional feature space; this is good for the case of email because in fact every word can be a feature. Text is generally a high-dimensional space, but when you take a specific instance of a document, its feature vector is likely to be sparse, with most feature frequency counts coming up zero.

3.3 Human-Computer Interaction Design

The two previous sections address the information that might help a CMC interface, and how a computer might model this information. This section deals with how this information can be made useful from a Human-Computer Interaction (HCI) design perspective.

There is a great deal of inspiring work in this field, especially in terms of creative interface techniques for information representation and retrieval. A couple of early prototypical works include:

- Muriel Cooper describes an ideal interface she termed "information landscapes" where a user finds information they need instantly and the experience of navigating is as useful as the information itself [Abrams].
- SemNet [Fairchild] is a three-dimensional graphical interface that explores techniques in the presentation of large amounts of data.

The field of information visualization demonstrates the possibility of improving a user's performance through a graphical interface. Presented in the right way, the right information can create an instantaneous response from the user, making a computer interface more natural and intuitive. Inspired by this field, this work attempts to provide a user with information in a communication interface that instills a natural social response making them more proficient in their communication tasks.

Some of the HCI research that is more directly applicable to this work involves the design and usability of the current desktop interface paradigm. There are many aspects of usability to consider when designing a new interface, many of which are addressed in [Neilson]. When making improvements to a current interface it is important to consider the user's habits with the old interface, and the pros and cons of changing this interface entirely [Raskin]. The new interface can have evolutionary changes compared to the old one, thus taking advantage of the user's familiarity and knowledge of the current interface and hopefully lowering the learning curve. Alternatively, a revolutionary change in an interface could be harder to get used to initially, but reap more benefits in the long term.

There are also a number interface design examples specific to electronic communication, which serve as motivating work:

- Conversation Map [Sac] is a Usenet newsgroup browser that does automatic content analysis.
- Treetables [Newman] is a tool for visualizing email threads.
- Babble [Erikson], is a communication tool for small- to medium-sized corporate groups that promotes "social translucence", providing cues about proximity and activity of other participants.

4. Design & Implementation

The goal of this research is to understand some aspects of a user's personal social network and utilize this understanding to augment their online communication experience. In fulfillment of this goal, the DriftCatcher email client displays social context information associated with a user's mail. This social context information comes from the automatic personal network analysis of the SocNetServer, which has agents that keep track of the various relationships in each user's personal network, and statistical models of the social context of email, support vector machines (SVMs), that let it recognize the social resources exchanged.

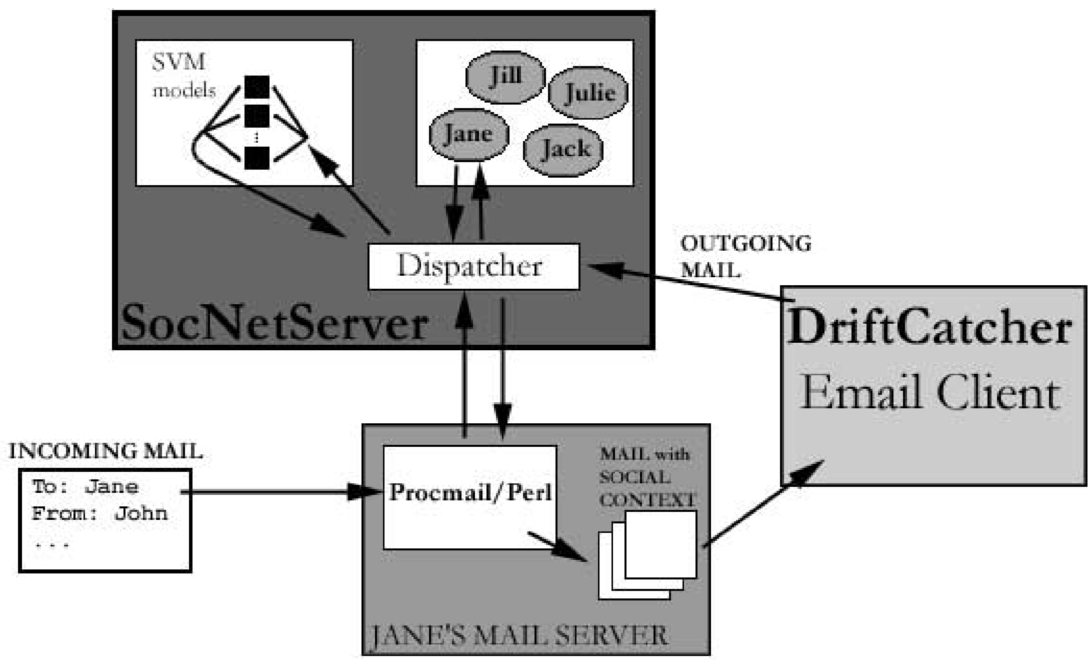
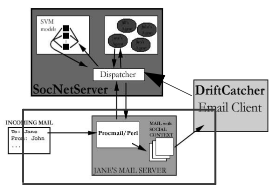


Figure 4.1: System Architecture

The system architecture has three components whose design and implementation will be detailed in this chapter

1) Processing of incoming mail by Procmail and Perl, which adds to the mail social context information (provided by the SocNetServer)
2) DriftCatcher email client that utilizes this information
3) SocNetServer, which has agents that aggregate personal network data, and models of social context, SVMs.

4.1 Incoming Mail Handling



For a user of the DriftCatcher email client, their mail has to be processed along the way to its final destination on their mail server. This interception is achieved through the use of Procmail [van den Burg], a mail processing utility that runs under Unix. Procmail is a mail-filtering program to help users filter and sort their mail (by sender, subject line, keywords, etc.). The Procmail script for this system adds information to every incoming message.

The Procmail script:

1) It parses the message into its various fields (to, from, body, etc.) and calls a local Perl program that accesses the SocNetServer to get the social context statistics about this message
2) Adds this extra info to the message header and forwards the mail to its final destination mail server

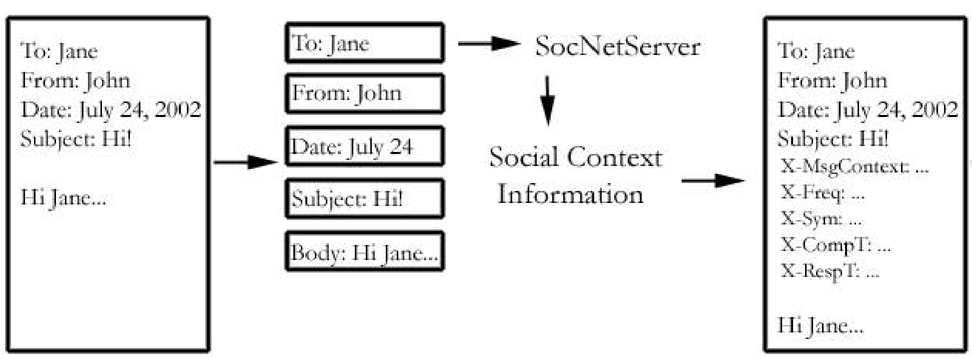


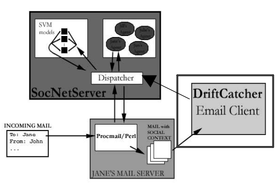
Figure 4.2: Functionality of the Procmail incoming mail script

How does this work for an example user: jane@media.mit.edu?

Jane asks the system administrator of her mail server to make sure that Procmail is used as her Mail Transfer Agent (this is commonly already the case). Then all that Jane has to do is copy the Procmail script and Perl program into her home directory on her mail server. From that point forward the system processes all of her incoming mail and the extra social context information is added to the headers of all messages. Jane can then open her mail with the DriftCatcher email client, that understands these extra header fields, and she sees the message from John in context.

The low barrier to entry was an important design point of the system. Any user that puts these scripts on their mail server allows the system to start keeping track of their personal network and marking their mail with social context information. This information is then accessible by using DriftCatcher to view their mail. If they don't use the DriftCatcher client to view their mail, the extra header information is just ignored and they see their mail, as they would have otherwise.

4.2 Social Context Mail Client: DriftCatcher



With the information that the SocNetServer provides about a user's personal network, DriftCatcher is in a position to organize and visualize mail according to social information. Its intension is to make it easier for the user to see what is happening in their personal communication network, and allow them to deal with communication in a social context rather than the current temporal context of mail browsers.

As mentioned in chapter three, familiarity is an important consideration in interface design. A number of new email interfaces were considered early in this work, some of which were a dramatic change from current email clients. The benefit of a completely new interface is the ability to experiment with the idea of completely changing the way people use email. However, email is a tool that people use every day and most have done so for years. With all of that experience, most users have a familiarity with the tool that allows them to be very proficient in spite of the tool's downfalls. Therefore, rather than throw away all of that experience and proficiency, this work makes incremental changes, adding social context to the basic email interface. The intension is to increase a user's proficiency at using a tool with which they are already familiar.

The DriftCatcher webmail client is an extension of [Emumail], an open source webmail client. It is a social context email client that is social in two dimensions. Dynamic data collection: the client is watching social aspects of the user's behavior in the application and communicating this to the SocNetServer. The display: messages with social data in the message header are recognized by the client and reflected in the way the messages are displayed in the inbox.

Dynamic data collection

An email client is a program that is used on a very regular basis; most people use email several times a day to communicate with people. This puts it in a prime position to collect information dynamically about how the user behaves with the various people in their personal network. It sees how long you spend reading messages, how long you take to compose messages, how long you take to reply to a message once it's been read, and a number of other behaviors. In the current implementation the DriftCatcher client sends information to the SocNetServer about compose time, and read time along with outgoing messages. The SocNetServer can then incorporate this into its knowledge of the user's personal network.

Display Changes

Based on the extra header information expected in the messages, DriftCatcher is able to display the inbox along social dimensions as well as the typical temporal dimension.

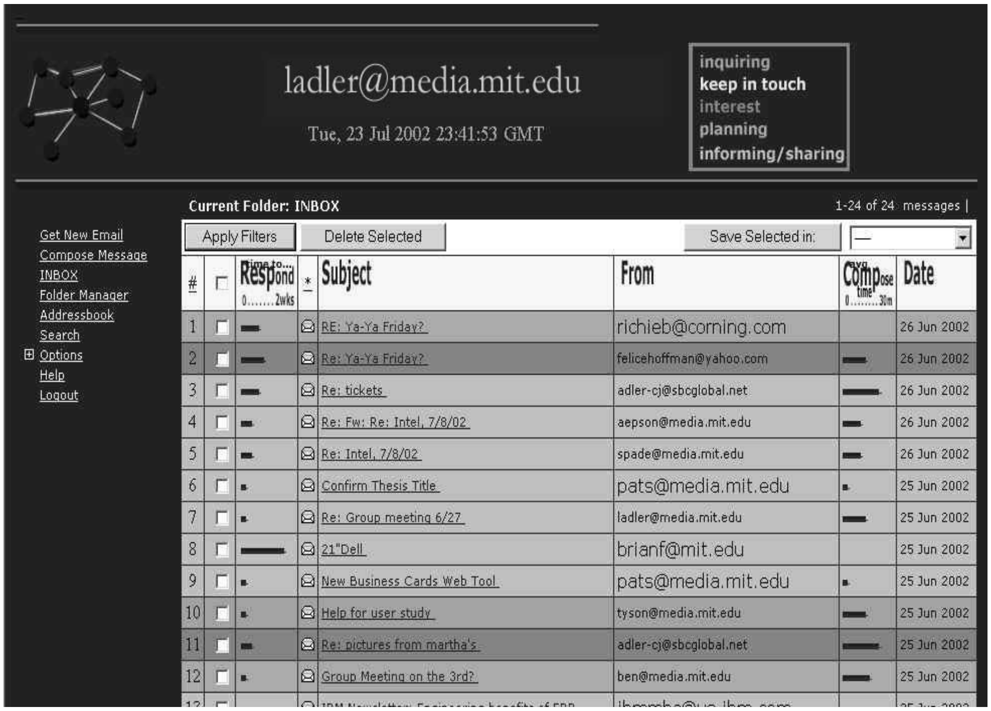


Figure 4.3: A sample DriftCatcher inbox display

Sender's names are displayed in different font sizes, based on tie strength. The weak ties are bigger than the strong ties with four resolutions. This mapping is a direct implementation of the "strength of weak ties" theory mentioned in chapter three. The theory is that weak ties are better for finding out new information and gaining access to other networks (which most likely have other resources, establishing greater social capital). However, a majority of the people in the user study found this to be counter-intuitive, so the next generation of DriftCatcher would either reverse this mapping or provide a different indication of tie strength.

With each message, DriftCatcher displays the average time that the user takes to compose messages to this sender (between 0 and 30 minutes). The compose time measure is based on messages composed with the DriftCatcher client, and it times out if the user stops typing for more than two minutes. This is certainly just a rough estimate since the user may use other clients from time to time, or compose a message with an external editor and copy the text over.

As shown in Figure 4.3, the left most column is an indication of how much time there is to respond to this message (between 0 and 2 weeks). The time to respond encourages reciprocation of the response pattern of the sender. The default time for a new contact is two weeks, and this changes once a response behavior is established.

The background color of each message changes to reflect the social context classification of the content by the SVM models. As indicated in Figure 4.3, Green=*Inquire,* Yellow=*KeepInTouch*, Pink=*Interest*, Orange=*Planning*, and Blue=*Inform/Share*. One issue with this is conflict resolution: what does the color do if a message is planning and informing, or interested and inquiring and supportive, etc.? The information DriftCatcher receives from the message header is a list of contexts, each with a confidence rating. Currently the client chooses to display the context with the highest confidence rating; however, it is naïve to assume that messages fit into only one context. A future goal is to experiment with the indication of multiple contexts simultaneously, and a few options are proposed in section 6.2.

4.3 Automatic Personal Network Analysis: The SocNetServer

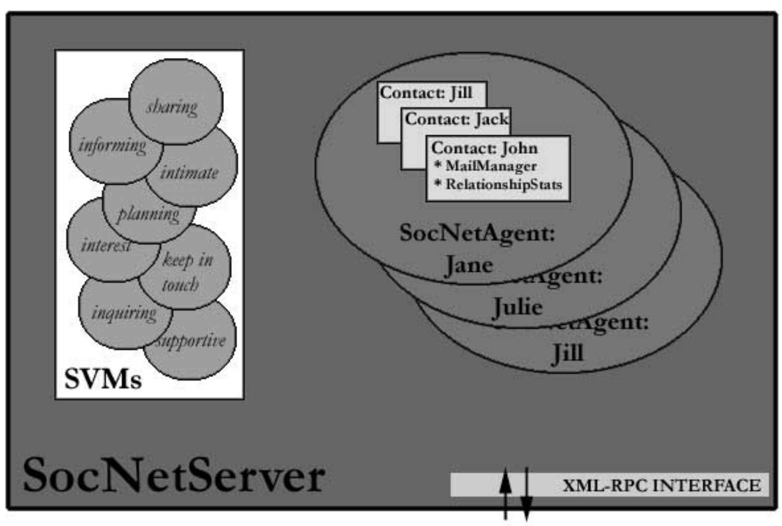


Figure 4.4: Components of the SocNetServer

The SocNetServer embodies the automatic personal network analysis that enables the DriftCatcher client described in the previous section. The SocNetServer has agents that keep track of statistics on the various contacts of each user's personal network, and statistical models of the social context of email, support vector machines (SVMs), that let it recognize the social resources exchanged. It shares intelligence with the outside world through an XML-RPC interface. This choice of interface made sense for two reasons: the rising popularity of web services in general, and the lack of dependency on a specific programming language or platform.

Functionality provided in the XML Interface

Process Incoming Message – Two things happen when a new message is received. First the social context of the message is calculated with the SVM classifiers. Then the agent for the recipient is called and alerted that there is a new message. This agent wakes up and produces information about the relationship between the recipient and the sender:

- Frequency of contact
- Symmetry of contact
- Response time – how long the user should/could take to respond to this message, based on the average time this sender takes to respond to the user, encouraging reciprocation of response time.
- Compose time – average time the user takes to compose a message to this contact based on information received over time from the DriftCatcher client.

This information is then returned to the client that made the request. Generally this function is always called from the Procmail/Perl scripts that put this information in the message headers, but other applications that are interested in the information could use this function as well. Examples of such "SocNet" applications are the Media Connector being developed by Surj Patel at the Media Lab, and inCall [Thomaz] a context-sensitive phone system.

Process Outgoing Mail – This is invoked from the DriftCatcher client. When a user sends a message, along with delivering the message, the client also sends the message and compose time to the SocNetServer. The SocNetAgents need to see both incoming and outgoing mail in order to do calculations such as frequency and response time.

The next two sections will go through the two major parts of the SocNetServer. Addressed first are the agents that aggregate knowledge about an individual user's personal network. Then second is the design and implementation of the social context models, SVMs.

4.3.1 Collecting Dynamic Network Info: SocNetAgents

The SocNetAgents comprise the automatic personal network analysis of the SocNetServer; they keep track of statistics on the dynamics of the relationships in each user's personal network. There is one agent for every user. When the SocNetServer receives a request for a user that it doesn't know, it creates a new agent for this user. The agent's main purpose is to keep track of all the people in this user's personal network. It does this by having an EmailContact structure for each person. An EmailContact keeps track of all the mail that goes between the user and the contact, and various statistics about this particular relationship (frequency, symmetry, strength, etc.).

The main concern here was privacy. There is a different agent for each user so that their information is kept and maintained in their private user space on the mail server. The system could go up another level and analyze and make information available about the whole communication network, but then the individual user loses control of the information. In this current implementation, as a user, the information about your relationship to your boss is only available to you.

4.3.2 Modeling Social Context of an Email Message: SMVs

As discussed in chapter three, Support Vector Machines are a supervised machine learning technique. The basic strategy starts with a corpus of emails for training a classifier. Each email is labeled with metadata pertaining to social intention. SVM learning techniques are applied to this corpus in order to find discriminating features and weigh the extent to which social context depends on these features. These resulting models are used to classify new messages. The remainder of this section will go through the details of this modeling building process.

Data Acquisition Alternatives

Acquiring this corpus of training email data is not a trivial task due to privacy concerns [Rogers]. Ideally it should come from more than one person, and as much data as possible is needed for best model building success. A number of alternatives were considered.

One option is to convince people to donate their inbox; a number of friends and colleagues were willing to help in this way. While this method yields a great amount of data with various real social interactions online, problems arise since it is not a closed set of people. With one-sided data, measures such as response time and symmetry cannot be calculated. Additionally, we would not have the consent of all the senders in each person's inbox.

There were two opportunities that involved corporate databases of email. One was a corporate customer relations database. This database would yield a great amount of data, but very little variance of social context: many examples of people writing in to someone they have never met before about a work related problem. The second database was a corporate mail corpus from a Media Lab sponsor, but only the header information of each message was available, so it would not work for content modeling.

The only email corpuses available publicly are a number of mailing lists and newsgroups. Since the intended application of this classification is a private email application, the training data should ideally be personal mail, not messages to a mailing list. While public mailing lists do show some variance of relationship and social context, the belief was that there is not enough variance to build a discriminating model for the domain of personal email.

The method that was used in the end, constructs a data corpus around a created social situation. A group of people is asked to volunteer to use a specified email account and email each other for a month. During this month they get to know each other both on and offline. As motivation to participate and means of getting to know each other, a party is thrown once a week for all those participants that write at least 20 emails to other participants. Theoretically, given enough participants, at the end of the month there is a large corpus of email, which is a closed set, and only contains mail from consenting participants.

Throwing Parties

Participation was solicited from students and other members of the MIT community, and over the course of a few weeks in January these people were asked to use designated email accounts to communicate with each other. The participants were notified that their email was being collected but were not told why. There is a possible bias of the data, in that people might act outside the norm when they know their email is being collected. However, while not ideal, this dataset has true socioemotional content and is therefore valid for the purpose of this research. While we had hoped to get a participation group with more than twenty people, in the end we had a group of six consistent members and four intermittent members. Their designated email accounts existed on a server in the Context-Aware Computing group, and all of the email was parsed and stored in a database for the remainder of the project.

The participation group contained people that already knew each other and people who did not. Over the course of the month we held a party every Thursday, where people who didn't already know each other were able to meet and start a relationship. Additionally the group was supplied with games and organizational tasks every week to create some diversity in the email conversations. Examples of these: riddles they were asked to solve in groups of 3 or 4, those email personality quizzes, organization of the Thursday parties (when, where, what). At the end of the month, there was a collection of approximately 550 email message ready to label for use in the statistical model-building phase.

It is important to note that the members of this constructed personal network interacted both online and face-to-face, and moreover that the face-to-face interactions (which were not measured or collected in any way) influence the content of online interactions. For the purpose of this project, this only makes the dataset more realistic. A machine attempting to classify social contexts of email is always going to be missing the knowledge of face-to-face interactions.

Observations and Anecdotal Evidence

Since the mail that these people sent back and forth is the *example* from which the machine is going to learn, it is interesting to mention some observations of the social dynamics of the group. When asked what their motivation was for coming to the event most people gave one of two answers: "I wanted to meet new people" and "I was intrigued by the advertisement for the event" (in appendix). The corpus is therefore made up of email from fairly outgoing and adventurous people who were all motivated to get to know each other and excited about interacting with people they'd never met.

Many people used the group mailing alias at first, but then broke off into personal conversations. This proves the point about how using a newsgroup corpus misses these more personal interactions.

With a specified minimum number of emails for each week, instigations may not be very natural. Additionally, the response time and symmetry are unrealistic since people had other reasons for getting back to everyone quickly.

There was evidence of in-out group behavior. The most explicit example is when people forwarded mail to each other commenting about other people in the group. At first the in-group was the two people who already knew each other and then over the month the in-group grew to about five or six. A recognizable pattern: person A and person B are of the in-group, person C is in the out-group. Person A gets an email from person C. Person A responds to person C. Then soon afterwards, as a separate interaction, forwards the mail from C to B with comments about C, etc. This behavior stops once C becomes part of the in-group. Also, some people were intentionally excluded from the in-group. Pattern: outsider makes an attempt, an inquiry or a suggestion, and the main group purposely ignores it. So much so that they even talk about the outsider to each other “did anyone answer her? Good, me either…” Additionally, many of the members were unanimously upset when a person (non in-group) sent spam.

At the end of the data acquisition phase I still had high expectations of the modeling accuracy. I was pleased with the variance of online interactions that I had seen from glancing over the data. The next step involves labeling all the messages with social context metadata.

Annotation

As mentioned in chapter three, considering how Bales’ Interaction Process Analysis best translates from physical to online interaction inspired the context labels. Table 4.1 lists the labels along with the operationalizations that were used by the human coder that annotated the data. Thirty labels were used, expecting that some might have very few examples. Labeling was revisited and labels were added after starting the annotation process once and finding that there were messages that did not quite fit into categories. Figure 4.5 shows the java annotation application built around these labels. One person used this application over four days to annotate the data corpus, labeling for less than two hours per day so as not to suffer fatigue effects.

Ideally, more than one person would annotate the dataset, so the models will be more likely to apply to the general population. There would then need to be some coding reliability analysis comparing the similarity of the coders’ coding, and training coders until a reasonable reliability is reached. Additionally after the labeling is started, there should be periodic reliability testing to maintain coding consistency, and make sure that they are continuing to behave similarly.

LABEL	OPERATION	LABEL	OPERATION	LABEL	OPERATION
Urgent	Scale of 1-5 relative to the other messages in the set	Tone	Scale 1-5, 3 being neutral, is the tone of the message positive or negative valence	Formal	Scale 1-5, formality, looked at things like: openings and closings and formatting and names used
Solicit	Did the recipient solicit this message?	Period	Solicited message received periodically	Com.	Unsolicited message, advertising
Invite	Message inviting to go somewhere or do something	Info	Telling the recipient something, providing information	Persuade	Trying to convince the recipient of something
Inquire	Sender is asking the user something	Advice	More than informing, this is giving advice, expecting to possibly change a behavior	Intro	Sender is making an introduction, of himself or herself or someone else
Keep Touch	There's no purpose to this message other than to maintain contact	Discuss	The purpose of this message is to	Motivate	The purpose of this message is to motivate the recipient to do something
Share	The sender is disclosing information, less informal than *informing*	Suggest	Less formal than advice, the sender is suggesting an action to the recipient	Plan	The purpose of this message is to plan a course of action, schedule an event
Thanks	Sender expresses appreciation	Regret	Sender expresses sorrow	Interest	Sender expresses interest in the recipient or the recipients ideas
Support	Sender shows support of or solidarity with the recipient	Intimate	The message indicates an intimate relationship, has some self-disclosure, talks about feelings	Demand	More confrontational than *persuade*, the sender demands a behavior from the recipient
Approve	Sender shows agreement with or approval of the recipient	Disagree	Sender shows disagreement with or disapproval of the recipient	Polite	Sender is going out of their way to be nice or courteous, usually a more formal message
Concern	Sender expresses concern for the recipient, usually seen with *support*	Playful	The message is playful, fun, or funny, often seen with *keepintouch*	Rude	Opposite of *courteous*, sender is being rude or crass

Table 4.1: Labels of social context used to annotate the dataset.

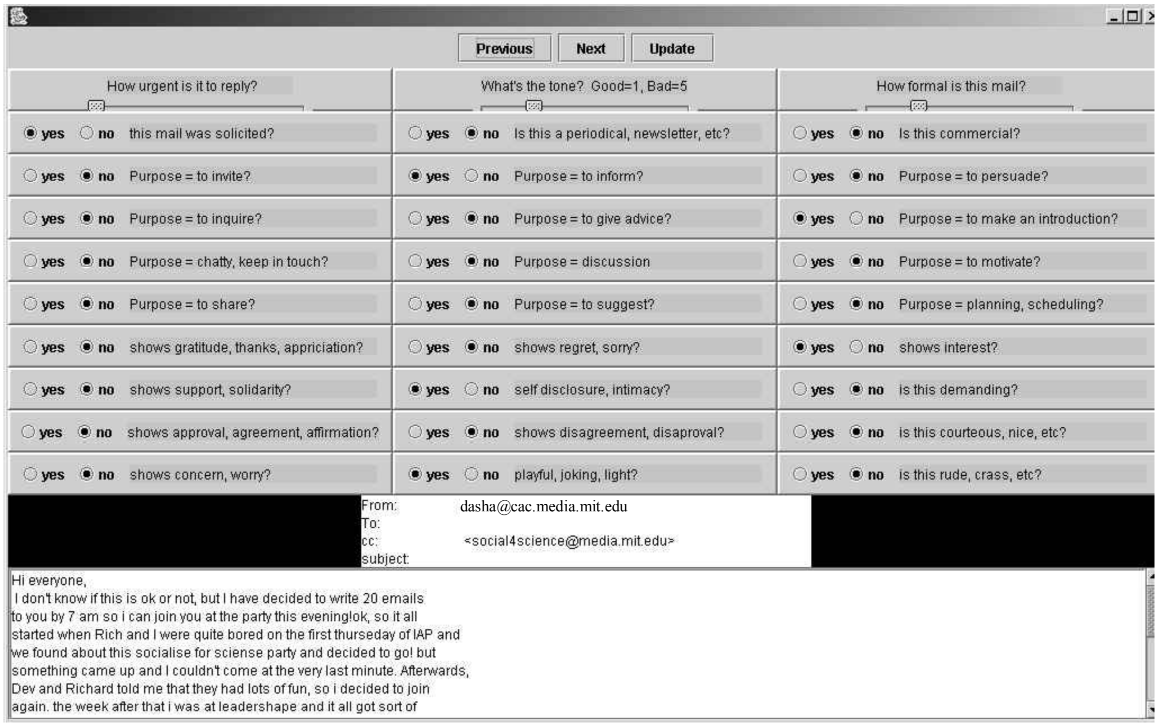


Figure 4.5: Java application used to annotate the dataset.

Expectations after the annotation phase went down a bit. I realized that there was simply very little data for a number of the classes (see breakdown in Table 4.2). There were basically 8 or 9 classes that had a reasonable amount of data for model building.

100 to 300 examples	40 to 100 examples	Fewer than 30 examples
Solicited, Informing, Inquiring, Interest, Keep in Touch, Share, Planning, Supportive, Intimate	Approve, Playful, Discuss, Suggest, Needs Response	Advice, Invitation, Gratitude, Regret, Motivate, Disagree, Introduction, Persuade, Demand, Courteous, Concerned, Commercial, Periodical, Rude

Table 4.2: Numbers of messages for each social context label

An Email Feature Set

Now that the data corpus has been collected and labeled, the next phase of the problem involves extracting feature from email that have social significance and will be used as the feature set to build a representation of the characteristics of the social connotation of email. The goal is to be able to classify a particular piece of mail as belonging to one or more of the social contexts mentioned in the previous annotation section.

There are a number of features that were believed would vary significantly depending on the social context and relationship between the sender and the receiver. These features are parsed from a message; Table 4.3 shows the feature queries implemented in this system.

Parts of the message	Numerical features	Ratios
getDate	getWordCount	getWordToSentenceRatio
getTo	getSentenceCount	getPunctuationToWordRatio
getCC	getNumberEntriesInTO	getPositiveEmoticonsRatio
getFrom	getNumberEntriesInCC	getNegativeEmoticonsRatio
getForwardedFrom	getTotalNumberRecipients	getDateRelatedWordsRatio
getSubject	getNumberOfPositiveEmoticons	
getBody	getNumberOfNegativeEmoticons	
getOldMessage	getNumberOfURLs	
wasOldMessageIntermingled	getNumberOfDateRelatedWords	
getAllEmoticons		
getAllPunctuation		

Table 4.3: Feature extraction functions of the Extractor class.

Implementation of Feature Extraction

Feature extraction is achieved through the implementation of two java classes: Extractor, a class that encapsulates all of the information about a single piece of email, and ExtractorManager, a class that contains a group of Extractors and represents most everything you could want to know about a group of messages.

Extractor has query functions listed in Table 4.3 above, allowing another application to ask about the [# of sentences], [# of positive emoticons], [# of URLs mentioned], etc., in a message. ExtractorManager has a set of extractors (messages) and acts as an interface to information about this group of messages. Another application is able to ask: “how often does person A talk to person B”, or “how long does person A generally take to respond to person B”. With this group of messages, ExtractorManager is also able to build the graph representation of a personal social network, given a root person (ego).

Statistical Modeling Approach

The email data corpus is now labeled with social context meta data, and there is the ability to extract different features about each message. Let the model building begin. The hypothesis is that there are subsets of features (*words, emoticons, punctuation, etc.*) that discriminate between the various classes of social context (*informing, inquiring, planning, etc.*).

A number of AI techniques could be used to accomplish such a classification problem. In the case of Expert Systems, or Rule-Based approaches, we would get a "social context expert" to give us rules about email and social context (i.e. when you see "Love," as the closing this is an intimate email). Then we would write a program that uses these rules to categorize incoming mail. An alternative to this is a statistical approach. The basis of such Machine Learning approaches is that maybe there isn't an expert that can list the regularities or patterns of similarity between the different classes of email. Therefore a reasonable alternative is to have a computer perform pattern recognition and learn its own rules.

Supervised learning techniques such as neural networks or support vector machines are designed to determine the extent to which various features of a dataset divide it into subclasses. The statistical modeling here uses the weka [Witten] implementation of support vector machines, which implements the sequential minimal optimization algorithm, SMO [Platt].

Building SVMs with Weka

Weka is a collection of machine learning algorithms for solving real-world data mining problems. Weka is open source software issued under the GNU General Public License. For this project, the algorithms of weka were incorporated into java code to build models for each of the social contexts from the email dataset (each of the labels in table 4.2).

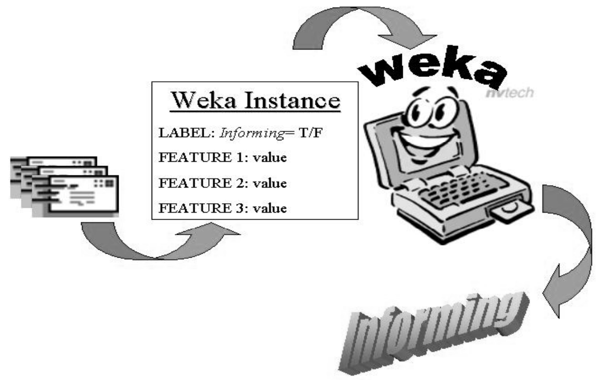


Figure 4.6: An SVM model build with weka

The Weka SMO is a java object that, once given a set of Instance objects, builds a classifier object. An Instance object consists of a feature vector and a class label. Here is an example of this process for the *informing* label (see figure):

1) Make each message in the dataset into an Instance object, using the Extractor code mentioned above to get the feature set for the message.
2) Add the label to the Instance object indicating if this particular Instance is a positive or negative example of *informing*.
3) Build the SMO classifier.
4) Test its functionality with 10-fold cross-validation testing.
5) Repeat to find a model with better results, until an acceptable model for *informing* is found.

Cross validation testing is where the use of the training set is maximized by using it all for training and testing. Training happens with 90% of the data, holding 10% of it out for testing. This process repeats ten times using a different 10% portion each time, and averaging the results.

Classifications Results

There were 8 labels that had enough data to build more than naïve models: *Informing, Inquiring, Interest, KeepInTouch, Planning, Sharing, Intimate, and Supportive*. A naïve model being a model that is built by saying "in the training data the majority of examples have this label as true, so I'm going to guess that any new example is true as well". This is bad because when the model build comes up with a naïve model it means it didn't find any significant correlations between classes and features, and it will not be generalizable. Mainly this happened when 1) there wasn't enough data for a particular class or 2) too many features were used to build the model and there weren't enough examples to converge to a solution.

The fact that this was a small dataset, made it hard to take advantage of the best qualities of SVMs. They are particularly good at learning to classify in a large feature space with sparse data, but this is dependent on there being enough examples for the algorithm to converge to a solution. A few different techniques were employed to increase accuracy, given the small dataset. The algorithm tends to converge faster, or do a better job in general, when it has relatively the same number of true and false examples. Better models resulted, when the training set had equal numbers of true and false examples rather than using all 550 email examples. Shrinking the feature space also improved the ability for the algorithm to converge, and some experimentation was done with various combinations of the feature set. The final models with the best performance (see Table 4.4) were built with the following feature set: Sentence terminating punctuation, the frequency of punctuation used, time and date related words, URLs, whether or not the old message was included in the new message, and the frequency of emoticons used.

Class Label	**Best Cross-Validation Result**
Informing	48
Inquiring	60
Interest	61
KeepInTouch	60
Share	71
Planning	62
Supportive	58
Intimate	68

Table 4.4: Results of Cross-Validation Testing

The extent to which these cross validation results will translate to actual results is dependent on the extent to which the data corpus is a reflection of the real world.

Given that the accuracy rates were around 60-70%, this doesn't instill a lot of confidence. On the other hand, let's think about how accurate people are at determining the social context of an email message. Maybe people don't do much better; at the very least, there is anecdotal evidence that people get the social intension of email wrong occasionally. Part of the evaluation phase of this project will be to compare the failure rate of these models to the failure rates (or disagreement rates) of human labelers.

It is important to note again that the size of the training data set (550 emails) indeed lowers the bar for expected accuracy. Given more data, the algorithm would have more examples from which to build models of context and would certainly yield better performance. In light of this, 60-70% accuracy should be viewed as a first attempt that shows good results in spite of sparse data and motivates future work.

Using these SVMs with the SocNetServer

The final step in the modeling process is to put these eight models of social context into the SocNetServer. This happens through a class called DriftSVM. DriftSVM is a serializable object, which means that it can be stored into a text file and read in at a later time by other programs that want to classify new email using these models. There is a separate DriftSVM for each of the eight SMO classifiers that were built. DriftSVM is an interface to the SMO classifier, which allows the SocNetServer to specify an email message and receive a probability that the message is of each particular context.

5. Evaluation

SocNetServer and DriftCatcher were evaluated on two levels.

- The first phase evaluates the extent to which the machine learning became successful in recognizing the social context of human communication. Did the system learn to categorize and label email similarly to a group of humans?
- The second evaluation phase questions whether or not an email agent that utilizes social network intelligence enhances online communication. Do people have a better sense of the value of communication when using the DriftCatcher client? This is a much more qualitative question and therefore harder to evaluate. To test this, a group of volunteers were given a motivation scenario and asked to perform email tasks with a sample inbox.

The participants, 36 overall, were all from the MIT community, and over half were students. This experiment was short term, involving people using the client for less than an hour, yielding data about the immediate effects of the social cues in the interface. The only longitudinal data is from my personal experience with the client and is mentioned at the end of this chapter.

5.1 The Message Set

In the evaluation of the DriftCatcher client, participants are asked to do a task around an email scenario, involving 24 messages, three times. Additionally, different inboxes are needed each time; therefore, a total of 72 messages are required (these will be referred to as the message set).

What messages should be in the message set? The belief was that the reality of the inboxes would be important to the generalizability of any results from this study. To accomplish this reality, three inboxes were constructed using my actual email in its original order. Since I have been using the DriftCatcher/SocNetServer system for a few months, all of my mail is in the database labeled with the SVM classification output and other social context data (*frequency, symmetry, response time*).

The 800+ messages from alockerd@media.mit.edu over the past month, were divided into 24 message blocks, such that each block is a real snapshot of my inbox from some point in the month. Three of the blocks were chosen as most appropriate for the study based on having a variety of message types (the context labels) and variety of the types of senders (in terms of frequency of contact). These three inbox snapshots were changed slightly to annonymize the data by applying the following rules:

- Andrea Lockerd (alockerd@media) changed to Lori Adler (ladler@media)
- For all personal messages, the names and emails were changed consistently across all three of the snapshots. For example, Ernesto Arroyo –> Andy Epson every time it appears in any of the 72 messages.
- For mailing list email and spam, no changes were needed.

5.2 Phase 1: Can a Machine Recognize Social Context?

This first phase evaluates the extent to which the machine learning became successful in recognizing the social context of human communication. Six participants were asked to label the message set with the eight main categories of social context that had the best cross-validation results. The precise instructions given to them can be found in the appendix. This evaluation addresses the following questions:

1) To what extent are the participants in agreement with each other about the message context labeling?
2) To what extent does the participants' labeling agree with the machine's labeling of message context?
3) To what extent does the first case correlate with the second? The hypothesis is that the machine will have higher agreement in the cases where people agree most with each other.

5.3 Phase 1: Results

This classification problem is a little different than others in the sense that the output is indefinite. When two people are talking about the social intension of an email message, this might involve some discussion and the two may not end up agreeing with each other. In this sense, it is unreasonable to expect a machine to always agree with everyone. This phase of the evaluation illustrates two points around this topic of similarity and agreement: the level of agreement in message labeling between different people, and the level of machine-human agreement in the message labeling. There are 72 messages and 8 labels available for each message, a total of 576 labeling opportunities.

There were six people that participated in this phase of the evaluation and labeled, given the same label definitions (these can be found in appendix A), the entire message set. There was a consensus, either positive or negative, in 473 of the 576 of the labeling opportunities. Therefore these six people agreed with each other about the message labeling in 82.1 % of the cases.

Now, the machine performance can be measured against this consensus. The machine models were used to label the 473 instances in which there was human consensus. In general, the machine tended to be more generous in giving a message a particular label, yielding a large quantity of false positives. The machine labeling agreed with 230 of the cases, 48.6 %, and 89.7 % of the disagreements were false positives.

According to this study, the machine is "wrong", according to the consensus of these six people, about half the time. While this does not sound particularly good, a second way to view the machine-human agreement is whether or not, for all of the 576 labeling opportunities, any of the six participants agree with the machine label. In this view the machine does much better. In 67.5 % of the labels, at least one person agreed with the label given by the machine. This shows that even though the machine does not always choose the majority answer, there is often at least one person that would argue that the machine labeling is correct. This second result is similar to the cross-validation results that indicated we should expect the models to be about 60-70% accurate.

In light of the small dataset used to build the machine models, these levels of agreement, while not great, are still promising. Future work that uses a larger base of examples to train a machine in recognizing social context could certainly expect to achieve even better results.

5.4 Phase 2: Does Social Network Intelligence Improve a Communication Interface?

The second evaluation phase questions whether or not an email agent that utilizes social network intelligence enhances online communication. Do people have a better sense of the value of communication when using the DriftCatcher client? To test the DriftCatcher interface and the extent to which the information provided by the SocNetServer augments an online communication experience, a group of volunteers were given a motivation scenario and asked to perform email tasks with sample inboxes from the message set. Participants are given the following scenario:

> *You work as a temp; today is your first day as the administrative assistant for the Context-Aware Computing group at the Media Lab. One of the people you support, a research staff member, Dr. Lori Adler, is going to be back from her meetings in 5 minutes. Go through her inbox, which has 24 new messages, and choose the three messages she should deal with first. Here are some things that Lori Adler would consider email priorities (in no particular order):*
>
> - *People trying to make plans or things that affect her schedule*
> - *People asking her for something or for advice*
> - *Making timely responses in general, and especially to people with whom she has a close relationship*

This scenario is appropriate for a number of reasons. Having each participant use their own email or a personalized inbox would be more realistic for them, but was unreasonable in terms of preparation time, privacy, and comparability of results. It was decided that the task would involve looking at another person's inbox and trying to step into the social context of that person. Additionally, five minutes is not long enough to read all 24 messages, so the participants are required to browse the inbox and use what is given by the interface to decide what is important enough to read. This is where the social client is expected to prove most useful, by giving more context information than just the date, sender, and subject line.

Each participant does this task of finding the most important/urgent messages three times, each time with a different inbox from the message set, with the following variations of the client:

Task 1: use the social mail client; with message context from the human labelers in evaluation phase one.

Task 2: use the social mail client; with message context from the machine labels.

Task 3: use the normal mail client; no extra context information.

Order effects are counter balanced by changing the order in which users do the three tasks. Ten people did each of the three task ordering variations (123, 312, 231), thirty participants total. The precise instructions given to the participants can be found in the appendix.

The following measures are used to examine quantifiable differences between using the client with and without social context.

- Total number of messages read.
- Percentage of close relation messages read.
- Percentage of messages read that required a quick response.
- Percentage of messages read of each of the contexts: informing/sharing, inquiring, keep in touch, planning, and interest/support.

Hypotheses:

1) The client type would affect the total number of messages a participant needed to read in order to complete the task.
2) When given the social context information in the client, the participants will attend to more messages that are related to the task.

5.5 Phase 2: Results

In considering the generalizability of this portion of the evaluation it is important keep in mind the task that participants were given. Essentially they were given too much information to deal with in a short period of time, and this study evaluates the extent to which the social context cues of the DriftCatcher client help a person make decisions in this situation. Hypothesis 1 was not supported by the data, and hypothesis 2 was supported. Therefore, while it does not change the number of messages a user is able to attend to in five minutes, the social context email client improves a user's ability to make judgments about which messages are most important to the task at hand.

Measures

The measures that were tested for significant differences between the three client versions were:

1) totalread: Total number of messages read
2) p_close: % of messages read that were a close relation
3) p_quick: % of messages read that required a quick response
4) p_inquire: % of messages read that were inquiring context
5) p_kit: % of messages read that were keep in touch context
6) p_interest: % of messages read that were interest/support context
7) p_plan: % of messages read that were planning context
8) p_info: % of messages read that were informing context

1-Way ANOVA with Repeated Measures, The Inbox Equivalency Problem

Since each person did each version of the task, the first type of test applied was a 1-way ANOVA with repeated measures. There were no significant differences found with any of these measures. It was thought this might be due to inequivalence of the composition of messages in the three different inboxes. For example Inbox 3 had three times as many messages of the planning context, and the relative number of message from close ties was imbalanced between the three as well. So, there were factors other than the client that are affecting the measures. Figure 5.1 shows a characterization of the inboxes and their composition equivalence.

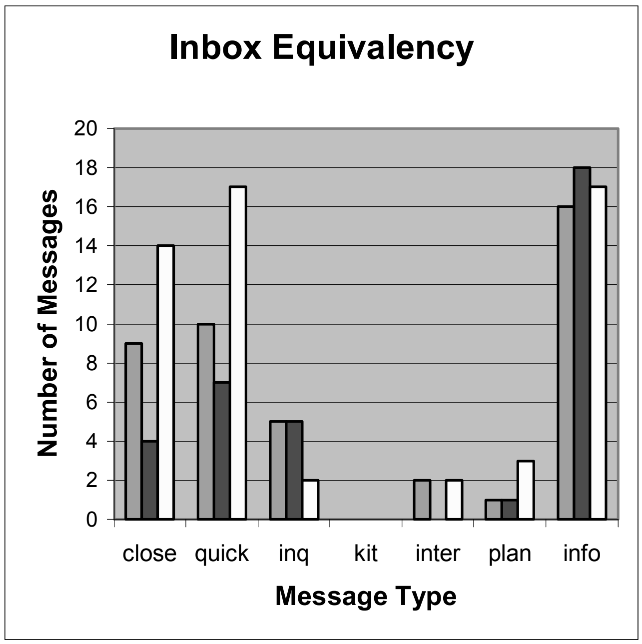


Figure 5.1: Composition of the three inboxes.

Further testing was done on each inbox as a separate case, to examine the differences that the client caused between people using the same set of messages (this is possible since each inbox has equal numbers of examples with task 1, 2, and 3; and order effects have already been counter balanced). A 1-way ANOVA was performed for each inbox for each measure with the client type as the factor, 24 tests in all.

1-Way ANOVA Results with Separate Inboxes

	totalread		% close		% quick		% inq		% plan		% inform	
box	f(2,27)	p >	f(2,27)	p >	f(2,27)	p >	f(2,27)	p >	f(2,27)	p >	f(2,27)	p >
1	1.71	.20	2.71	.09	5.89	.01	81.38	.00	3.22	.06	2.24	.13

2	.32	.73	6.1	.01	3.57	.04	.45	.64	23.88	.00	8.71	.00
3	.18	.84	.53	.6	.52	.6	7.1	.00	29.91	.00	18.51	.00

Table 5.1: Results of significant measures

There was no significant difference found in the total number of messages that were read in completing the task. Therefore having the new interface did not increase, but also did not decrease the number of messages a person can scan through and deal with in five minutes.

In two of the three inboxes, having the social client caused there to be a significant increase in the percentage of messages read that were from a close relation. With two of the three inboxes, the percentage of messages read that required a quick response increased significantly when a participant had the social client. In both of these last two measures, close and quick messages, inbox 3 was the one message set in which the social client did not have a significant effect. A possible explanation for this is the fact that in this inbox there were a relatively large number of close and quick messages (see figure 5.1), thus the likelihood of reading a large number of them was significantly higher than with the other two.

The remaining measures all concern the percentages of messages read of a particular context. Measure 5, the keep in touch label, is not relevant since there weren't actually any examples of this (see figure 5.1), and measure 6, the interest label, did not produce any significant effect. With the social client, both the percentage of inquiring messages read and percentage of planning messages read went up significantly, in at least 2 of the inboxes. The percentage of messages with the informing context was significantly less with the social client than without in 2 out of the 3 inboxes (and the difference was almost significant in the third).

These results support that in fact having the social context mail client helped people with their task; not by increasing the number of messages they could attend to, but by increasing the value of the messages they did attend to. In the instructions (refer to scenario in 5.4), people were asked to pay attention to scheduling, inquiries, close relations, and timely responses. The data shows that with the social email client, people read a greater percentage of messages that:

- were from close relations
- needed a quick response
- involved planning
- involved an inquiry

Questionnaire Results

In addition to measuring the participant's behavior with the client, they were also asked to answer open-ended questions on a printed survey, which adds some personal qualitative perspective to the statistical findings. A few of the more interesting answers and answers that were common across a large number of people are mentioned here.

While many people noted that the correlation of font size and frequency was very useful, they found the size correlation was counter-intuitive. This is interesting because we were able to show that even though the *user interface technique* was undesirable, the *information* produced significant effects, increasing performance. Additionally, one would expect that a more intuitive interface technique would only improve the effects even more.

It was mentioned that the color mapping was not intuitive and would take more time to get used to, and in general a number of people mentioned that there was a learning curve; given more time to get used to the interface they may have found it even more useful.

Some people mentioned rules that they used to pick out messages related to the task (times, dates, planning, inquiring, close relations), showing that the task prompting was effectively consistent.

Generally speaking almost everyone mentioned something about the social interface of DriftCatcher that changed their behavior and helped them step into the social context of Lori Adler's inbox.

Recommendations

These results mostly reflect positively on the DriftCatcher client. Two things will change based on this study related to the user interface: the font size correlation with the sender's name will be reversed, and the color mapping will be revisited to consider something more intuitive.

In future studies like this one there are a couple of things that should be done differently. First, it was found that when multiple inboxes are necessary the equivalence of the composition becomes a factor. If this exact study were to be repeated, the three inboxes should be constructed with the same relative number of each message type, and possibly even with the various types occurring in the same order. Second, since a large number of people mentioned the learning curve of the interface, future studies should have a practice round in the beginning where people are able to become familiar with the interface.

5.6 Personal Observations

I have been using the DriftCatcher/SocNetServer system for all of my email interactions for over three months. While the designer and builder of a system are always going to be biased, it's worthwhile to mention some observations of my own experiences since this is the only longitudinal data about the usability of this system.

I agree with people from the user study that the context coloring is useful in spite of the fact that the color mapping is not intuitive. But it really only took about a week before seeing orange made me think planning, etc.

And surprisingly enough, I actually like the font size mapping. The smaller font size feels more intimate like a close relation so it is rather intuitive to me. It might also be that more people would agree with me if they saw their *own* inbox arranged this way with sender names that they recognized. But given the overwhelming majority of people that have objected to this mapping, I think it is important for it to change in some way.

The most exciting part of using the real system was to get a feeling for how often I agreed with the machine classification of the context labels. The cross-validation results reported in chapter four suggested that the classification should only be about 60% accurate; however, I think the machine classification is probably right more than 60% of the time. There are a number of reasons this could be the case. Most messages could be one of many things, so it may be that even though the message doesn't get classified in the "best" way; it still gets classified in an "acceptable" way. I may also be attributing a particular context to a message based on the machine labels. Regardless of what makes the classification seem more accurate than 60%, it is wrong from time to time which is relatively annoying and reduces confidence in the system as a whole. The ability to correct the system would help a great deal. This is addressed in section 6.2, future work.

6. Conclusion

Can a computer learn to understand the value of communication? This work has shown that, while not exceptional, as a first attempt the social context classification did perform with about 68% reliability (see section 5.3). Could a computer use this to help a person relate to other people through technology? This work found that the addition of social context to an email interface had a positive effect (see section 5.5). In this work Artificial Intelligence (AI) of Social Networks is used to improve human-human communication, recognizing the social characteristics of human relations in order to achieve a more natural online communication interface.

Email is a tool that people use practically every day, making an implicit statement about their relationships with other people, and providing an opportunity for a computer to learn about their social network. Furthermore, over the years people have come to utilize and depend on email more in their daily lives, but the tool has hardly changed to help people deal with the overwhelming amount of information. Many of the social cues that allow people to naturally function with their social network are not inherent or obvious in CMC. This work offers automatic social network analysis as a means to bring these cues to CMC and to foster the user's coherent understanding of the people and resources of their communication network.

6.1 Contributions

The main contributions of this thesis are the following:

1) Classification of social resources in email using machine learning techniques.
2) Using AI to augment a human-human communication medium with:
   a. Automatic personal network analysis of the SocNetServer .
   b. DriftCatcher email client (informed by the SocNetServer) which improves the way a user is able to mind their relationships.
3) The evaluation of both 1 and 2.

Modeling Social Context in Email

What kinds of social resources are exchanged between the people in the user's personal social network? A computer program that recognizes the social context of a message (i.e., *informing, inquiring, sharing, planning, intimate, etc.*) is in a better position to determine the value of that communication.

This work used SVMs to classify the social context of email messages, with the following steps: collection of an email data corpus, annotation of social context (*informing, inquiring, intimate, planning, ...*) , pattern recognition to discriminate between the classes of email based on various concrete features of an email message (*length, emoticons, punctuation, ...*).

This work introduced a technique for acquiring a corpus of personal email. By creating a social situation, throwing parties, natural personal email was collected from a group of volunteers over the course of one month. SVMs were then built around these examples.

The accuracy of the SVMs models was tested with human labelers. The experiment also allowed us to look at the level of agreement between people, and found that they agreed with each other 82% of the time. Considering the messages in which there was a consensus about the social context among the human labelers, the machine agrees with that consensus 49% of the time (with most of its disagreement being false positive). Alternatively, in 68% of all messages, there is at least one person that would argue that the machine labeling of social context was correct.

Automatic Social Network Analysis

The SocNetServer is introduced in this work, and is the social network intelligence of the system. It compiles personal social network information for a user based on their email interactions (who they communicate with, frequency of contact, symmetry of contact, response times, time spent composing/reading messages). It has statistical models of social context of email (the SVMs described earlier). It also has an XML-RPC interface allowing clients to connect to it and exchange social network information about a user.

Email with Social Context

The DriftCatcher email client serves as an example of a SocNet application, built to utilize the artificial intelligence of the SocNetServer, with the goal of helping users understand and maintain their social network more naturally. The DriftCatcher email client helps a user catch the drift of what is happening with their personal communication network.

- DriftCatcher displays email in more than just a temporal context, adding social context cues based on information from the SocNetServer.
- It completes the loop by sending informing about user behavior back to the SocNetServer.

An experiment was conducted to measure the extent to which the social context of the DriftCatcher enhances the email experience. The results of this study show that having the social context mail client helped people with an email task that involved stepping into the social context of another person's inbox. In the task instructions (refer to scenario, in section 5.4), people were asked to pay attention to scheduling, inquiries, close relations, and timely responses. The data shows that with the social email client, people read a greater percentage of messages that:

- were from close relations
- needed a quick response
- involved planning
- involved an inquiry

6.2 Future Work

This work has been challenging and fulfilling and has a number of future directions. This section recommends work in three areas of this research: the DriftCatcher email client, the artificial intelligence techniques, and some general future directions for email research.

DriftCatcher Email Client

Two aspects of the current implementation that warrant further exploration are the indications of context and tie strength. Currently the message contexts are indicated with the coloring of messages, and allow only one context to be depicted at a time. A future goal is to have the flexibility to show that a single message is more than one context. Some possible solutions here might be to use more channels of display: color, transparency, patterns, texture. Another idea is for every context to be displayed as a color bar with each message, and the relative sizes of these color bars indicate the relative intensity of the various contexts. Additionally, the tie strength between the user and a particular sender is currently correlated to the font size of the sender's name. In the future this correlation should be reversed. Instead of weak ties being larger they should be smaller. Alternatively, there could be a different indication of tie strength altogether. Font type might be less drastic and more acceptable indication than font size.

Another suggestion for a future implementation of the client is a summary section of the current social context of the inbox. This would be placed in the top right corner of the screen and example would be:

> *"Hi, you have 20 unread messages. 5 of them are inquiring, 2 are planning, 4 are supportive, etc. Most are from people you don't talk to very much. 3 are ones that you need to get back to today."*

Recommendations for the AI

The most significant way in which the modeling of social context could be improved using the current SVM scheme: more training data. There simply were not enough examples of some of the contexts we would have liked to model. In the future, if this exact model building technique were to be used, the data collection event would need at least twice as many volunteers, and the resulting data corpus should have on the order of thousands of messages. Additionally, the annotation should happen with multiple people, much like what was done in the evaluation of this work, where the context label comes from a consensus of many human coders.

While this would be the way to get better models using the current technique, this is a very tedious process. Practically, it is hard to find that many volunteers for the data collection. It is even harder to find enough people who are willing to tediously label on the order of a thousand messages, never mind that they would need to be periodically assessed for consistency. Therefore, it is beneficial to consider other alternatives to the current model-building technique.

One promising direction is to look for opportunities to get the examples and the annotation from a user's interaction with the client. The direction in which I would like to take this research is to allow the user to explicitly train the system by "showing" it different examples of different email contexts. This could be implemented on top of the current interface by adding a correction module. The expectation is that this would be the most natural form of training. The system would make classifications, and if the user feels strongly enough that it is wrong they will "complain" and thus train the system through correction. Additionally, in an interface that is changing based on user input it is important for the user to be able to *see* that their input is causing a positive change. In the implementation of the correction interface, there should be a mode where the system shows the user its new classification of some past examples that were misclassified. This would allow the user to see the difference their teaching is imposing in the system.

Another aspect of letting users train the system is to allow for user-defined rules. While I still believe that a pattern recognition approach is the most promising in most aspects of the challenge of social context modeling, there are also cases in which users feel strongly that they know exactly what behavior they want from the system given a particular situation. Therefore, a good filter-maker would be enough to make some users happy. It should be complex enough to let the more advanced user specify regular expressions, but also have the ability to train by example with a more natural interface of the form: "with messages like these"…"do this".

Email in General

A common behavior among users seems to be treating the Inbox as a "To Do" list. A large number of people mentioned this on the user study post questionnaire, and I've had a number of conversations about email with friends who mention "oh, you know what I end up doing…my inbox is sorta like a to do list". Currently, I have only anecdotal evidence, but it seems that a large population of people exhibit a particular behavior that could be more explicitly supported in an email client. This is currently my biggest interest. I would like to find out more formally if this is a significant behavior, and experiment with ways in which this could be supported in an interface to reduce the cognitive load of having to keep track of everything there is "to do" in your inbox.

7. References

J. [Abrams], “Muriel Cooper's Visible Wisdom”, I.D. Sept./Oct. 1994.

R.F. [Bales], “Interaction process analysis: A method for the study of small groups”, Reading MA: Addison-Wesley, 1950.

C. [Breazeal], “Designing Sociable Robots”*,* Cambridge MA: MIT Press, 2002.

[Emumail], www.emumail.org.

T. [Erickson], D.N. Smith, W.A. Kellogg, M.R. Laff, J.T. Richards, and E. Bradner, “Socially translucent systems: Social proxies, persistent conversation, and the design of Babble”, *Proc of ACM Computer Human Interface, 1999*.

K. [Fairchild], S. Poltrock, G. Furnas, “SemNet: Three-Dimensional Graphic Representations of Large Knowledge Bases”, Readings *In Information Visualization,* Morgan Kaufmann Publishers, Inc. 1999.

F. [Flores], M. Graves, B. Hartfield & T. Winograd, “Computer System and the Design of Organizational Interaction”, *ACM Trans. on Information Systems*, Vol. 6. No. 2, 1988, (153-172).

L. [Foner], “Yenta: Multi-Agent Referral System for Matchmaking”, *First International Conference on Autonomous Agents*, 1997.

M. [Granovetter], “The Strength of Weak Ties”, *Social Structure and Network Analysis*, ed by Marsden, P. and Lin, N. Beverly Hills: Sage, 1982, (105-130).

T. [Joachims], “Text Categorization with Support Vector Machines: Learning with Many Relevant Features”, *Proc of the European Conf on Machine Learning*, Springer, 1998.

H. [Kautz], B. Selman, M. Shah, “Referral Web: Combining Social Networks and Collaborative Filtering”, *Comm. of the ACM*, v40, 3, Mar 1997, 63-65.

J. [Licklider], A. Vezza, “Applications of Information Networks”, *Proc of the IEEE,* 66(11), Nov 1978.

N. [Lin], “Building a Network Theory of Social Capital”, *Social Capital: Theory and Research*, ed by Lin, N., Cook, K., and Burt, R.S., New York: Aldine de Gruyter, 2001, (3-30).

T. [Mitchell], “Machine Learning”, McGraw Hill, 1997.

B. [Nardi], S. Whittaker, S. Heinrich, “It’s Not What You Know, It’s Who You Know: Work in the Information Age”, *First Monday*, May 2000.

P.S. [Newman], “Treetables and Other Visualizations for Email Threads”, *Tech Report in the Human-Document Interaction Group at Xerox PARC,*2001.

[Nielsen-Netratings], Global Internet Trends, 1st Quarter 2002.

J. [Nielson], “Usability Engineering”, Academic Press, Inc. 1993.

J.C. [Platt], “Fast Training of Support Vector Machines using Sequential Minimal Optimization”, *Advances in Kernel Methods - Support Vector Learning*, B. Schölkopf, C. Burges, and A. Smola, eds, MIT Press, 1998.

J. [Raskin], “The Humane Interface”, Adison-Wesley, 2000.

R. [Rice], and G. Love, “Electronic emotion: Socioemotional content in a computer-mediated communication”, *Communication Research* 14 (1), Feb 1987, (85-108).

S. [Russell], P. Norvig, “Artificial Intelligence: A Modern Approach”, Upper Saddle River NJ: Prentice Hall, 1995.

W. [Sac], “Conversation Map: a Content-based Usenet Newsgroup Browser”, *Proc. of IUI 2000*, (233-240).

W. [Stewart], “How Email Was Invented”, www.livinginternet.com, 2002.

E. [Thomaz], “Developing Distributed Contextualized Communication Services”, *MIT Master's Thesis, Program in Media Arts and Sciences*, 2002.

V. N. [Vapnik], “The Nature of Statistical Learning Theory”, Springer, 1995.

S.R. [van den Burg], Procmail, www.procmail.org.

A. [Vivacqua], H. Lieberman, “Agents to Assist in Finding Help”, *Proc of the ACM Conference on Computer Human Interaction*, April 2000.

J.B. [Walther], “Interpersonal effects in computer-mediated interaction: A relational perspective”, *Communication Research* 19 (1), Feb 1992, (52-90).

B. [Wellman], “For a Social Network Analysis of Computer Networks”, *Proc of SIGCPR/SIGMIS*, Denver CO, 1996.

I. [Witten], E. Frank, “Data Mining”, Morgan-Kaufmann, 2000.

Appendix A: Evaluation Phase 1 (Instructions and Questionnaire)

User Study of the Social Context of Email

July 2002

Participation in this activity is voluntary and you are free to withdraw your consent, and discontinue participation in this activity at any time without prejudice.

We are conducting research concerning computer-mediated communication (CMC). The field of CMC is interested in how people interact with and communicate with each other when there is some technology involved. In the case of this study we are looking at how people interpret the social context of email messages, and how well people agree about whether or not an email message is informing, inquiring, planning, sharing, etc.

In this study you will be asked to read through a collection of email, and give each message various social context labels. The study should take less than one hour to complete, but you are given no time limit.

You will receive a copy of this consent form, and any inquiries concerning the procedures should be directed to:

Andrea Lockerd -- alockerd@media.mit.edu -- 617.253.0597

In the unlikely event of physical injury resulting from participation in this research, I understand that medical treatment will be available from the M.I.T. Medical Department, including first aid emergency treatment and follow-up care as needed, and that my insurance carrier may be billed for the cost of such treatment. However, no compensation can be provided for medical care apart from the foregoing. I further understand that making such medical treatment available; or providing it does not imply that such injury is the Investigator's fault. I also understand that by my participation in this study I am not waiving any of my legal rights*. I understand that I may also contact the Chairman of the Committee on the Use of Humans as Experimental Subjects, M.I.T. 253-6787, if I feel I have been treated unfairly as a subject.

*Further information may be obtained by calling the Institute's Insurance and Legal Affairs Office at 253-2822.

I agree to the procedures of this activity ________________________ Date: ____________

Principal Investigator ____ Date : ____________

~ Pre-Study Questionnaire ~

1. When you are reading through your email what would be some characteristics that would make you call a message each of the following:
   a. Informing____
   b. Sharing____
   c. Inquiring____
   d. Interested____
   e. Supportive____
   f. Planning____
   g. Intimate____
   h. Keep in touch ____

2. How often do **people in general** get social connotation in email wrong?

   Never Seldom Frequently Often

3. How often do **you** get the social connotation of an email message wrong?

   Never Seldom Frequently Often

~ INSTRUCTIONS ~

There is no time limit. You will be giving each email messages up to 8 labels. Here is a brief explanation of each label.

Informing – the message is telling the user something, providing some information.

Sharing – the purpose of the message is to disclose something; in general this is more personal than informing or involves some type of self-disclosure.

Inquiring – the message asks something of the recipient.

Interest – the message shows interest in the recipient; gives attention to the recipient's ideas, or the topic of conversation in general.

Keep in Touch – the purpose of the message is just to maintain contact.

Planning – the purpose of the message is to organize something. In general this is a message that requires the recipient to look at or change their schedule.

Supportive – the message shows support, supporting an idea of the recipient or being supportive of them as a person in general.

Intimate – the message is intimate, any message that in some way indicates a close relationship. For example, an inside joke, or a shared language that is different from messages with others (particularly in the closings: "Love, Dad")

If you have questions about the labels or need more examples please ask now.

You will be logged into a webmail client that has an inbox full of messages. Please follow these steps exactly so your answers get logged in the system.

1. Click on the first message in the inbox.
2. Read the message.
3. Scroll down to the bottom and check any context labels that apply (Yes, you can choose more than one for a single piece of mail, or none if you believe none apply)
4. Click on the NextMessage button (Note: this is the only way to properly exit one message and go on to the next. If you accidentally press something else, notify the experiment administrator to get back on track)
5. Repeat steps 2-4 until you reach the end of the inbox.

DON'T TURN THE PAGE

~ Post-Study Questionnaire ~

1. When you are reading through your email what would be some characteristics that would make you call a message each of the following:
    a. Informing____
    b. Sharing____
    c. Inquiring____
    d. Interested____
    e. Supportive____
    f. Planning____
    g. Intimate____
    h. Keep in touch ____

2. Do you think that other people would agree with your labeling?
   No   Probably Not   Maybe   Probably   Yes

3. Why? ____

4. Are there any labels that you felt should be available but were not?

Appendix B: Evaluation Phase 2 (Poster, Instructions and Questionnaires)

Read Email for my user study

Help me finish my thesis

Get a FREE ticket to Kendall Cinema!

I need subjects for my user study. It takes less than an hour. Come over to the Media Lab, read some email using a client that I developed, tell me what you think, and I'll give you a free movie ticket good for 2 years!

Interested? Send email to…

userstudy@media.mit.edu

Social Context Email Client User Study

July 2002

Participation in this activity is voluntary and you are free to withdraw your consent, and discontinue participation in this activity at any time without prejudice.

We are conducting research concerning computer-mediated communication (CMC). The field of CMC is interested in how people interact with and communicate with each other when there is some technology involved. This study is designed to evaluate a new email client, DriftCatcher, which has been developed at the Context-Aware Computing at the MIT Media Lab.

You will be asked to use this email client to perform three short tasks that will take no longer than 5 minutes each. There will then be questions for you to answer about your experience.

You will receive a copy of this consent form, and any inquiries concerning the procedures should be directed to:

Andrea Lockerd -- alockerd@media.mit.edu -- 617.253.0597

In the unlikely event of physical injury resulting from participation in this research, I understand that medical treatment will be available from the M.I.T. Medical Department, including first aid emergency treatment and follow-up care as needed, and that my insurance carrier may be billed for the cost of such treatment. However, no compensation can be provided for medical care apart from the foregoing. I further understand that making such medical treatment available; or providing it does not imply that such injury is the Investigator's fault. I also understand that by my participation in this study I am not waiving any of my legal rights*. I understand that I may also contact the Chairman of the Committee on the Use of Humans as Experimental Subjects, M.I.T. 253-6787, if I feel I have been treated unfairly as a subject.

*Further information may be obtained by calling the Institute's Insurance and Legal Affairs Office at 253-2822.

I agree to the procedures of this activity ________________________ Date: ____________

Principal Investigator ____ Date : ____________

~ Pre-Study Questionnaire ~

1. How many times do you check your email in a day?

2. How big is your inbox?

3. What order do you usually read your messages if you open your inbox and have more than just a few? By date, by sender, something else?

4. When you walk into a room full of people, pretty quickly you get a sense of who's there, how you know them, etc. – the social context. When you open up your email, do you have a good sense of the social context of your inbox? What are your expectations for a tool that helps with this?

5. How organized is your email?
   (Do you folder it? how many folders do you keep? Do you delete mail?)

6. If you want to organize a dinner with a few friends what method of communication would you be most likely to use?

   Email Face to face Phone Other _____________

7. Why?

~ INSTRUCTIONS ~

SCENARIO:

You work as a temp. You just got to your first day on the job at the Media Lab and you're filling in for an administrative assistant of the Context-Aware Computing group (cac@media). One of the people you support in the group is the new research staff member, Lori Adler.

TASK: (this will be done 3 times with two different variations of the email client)

Ms. Adler is very busy and hasn't been able to get to her email in the last day or two, so there are 24 new messages in her inbox. She is currently in a meeting and will be stopping by in 5 minutes. Spend the next 5 minutes looking through her email and pick out 3 messages she should deal with between meetings. Here are some things that Lori Adler would consider email priorities (**in no particular order**):

- People trying to make plans or things that affect her schedule
- People asking her for something or for advice
- Making timely responses in general, and especially to people with whom she has a close relationship

Each of the three times that you complete this task, space will be provided for you to jot down the message number of each email as you read it, if you go back and re-read a message please write down its number again. There is also a space to write down the message numbers of the 3 emails you choose for Lori to deal with first.

DETAILS ABOUT THE VARIATIONS of the CLIENT:

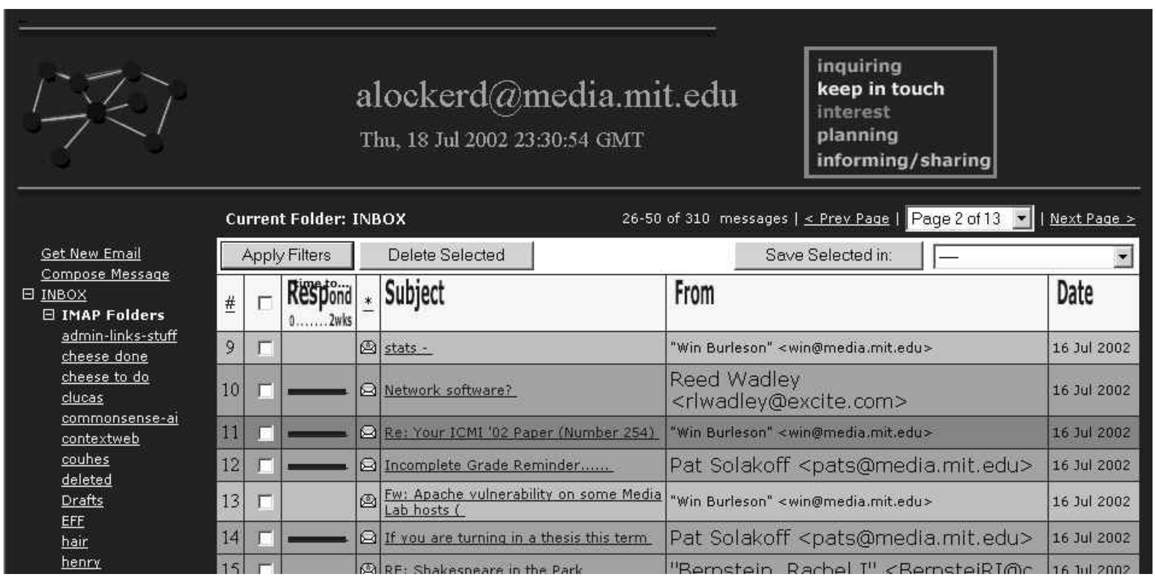
Normal – this is just a normal webmail client with a standard interface (subject, sender, date)

Social – this is an enhanced webmail client with the following additional features that you may find useful in your task. Here is a snapshot of my inbox with the social client.

1. The frequency of contact is noted in the font size of the sender. More frequent is smaller. (Win Burleson is a closer contact of mine than Reed Wadley)

2. There is a timer bar on the left hand side, indicating how long I have to respond to this email (from 0 to 2weeks). This is based on how long the sender takes to respond to my emails, thus encouraging the reciprocation of response time.

3. The different colors indicate the social context of the message. (White indicates that it didn't meet any of the categories) The categories are listed in their color in the top right corner.



If you have any questions about the instructions ask now…

WAIT FOR ADMINISTRATOR BEFORE GOING ON

TASK = Login as: drift using "normal"

Ms. Adler is very busy and hasn't been able to get to her email in the last day or two, so there are 24 new messages in her inbox. She is currently in a meeting and will be stopping by in 5 minutes. Spend the next 5 minutes looking through her email and pick out 3 messages she should deal with between meetings. Here are some things that Lori Adler would consider email priorities (in no particular order):

- People trying to make plans or things that affect her schedule
- People asking her for something or for advice
- Making timely responses in general, and especially to people with whom she has a close relationship

Write down the message numbers of emails as you read them

____
____
____
____

Write down the message numbers of the 3 messages you choose

1. ____________
2. ____________
3. ____________

WAIT FOR ADMINISTRATOR BEFORE GOING ON

TASK = Login as: drift   using "social"

Ms. Adler is very busy and hasn't been able to get to her email in the last day or two, so there are 24 new messages in her inbox. She is currently in a meeting and will be stopping by in 5 minutes. Spend the next 5 minutes looking through her email and pick out 3 messages she should deal with between meetings. Here are some things that Lori Adler would consider email priorities (in no particular order):

- People trying to make plans or things that affect her schedule
- People asking her for something or for advice
- Making timely responses in general, and especially to people with whom she has a close relationship

Write down the message numbers of emails as you read them

____
____
____
____

Write down the message numbers of the 3 messages you choose

1. ____________
2. ____________
3. ____________

WAIT FOR ADMINISTRATOR BEFORE GOING ON

TASK = Login as: drift using "social"

Ms. Adler is very busy and hasn't been able to get to her email in the last day or two, so there are 24 new messages in her inbox. She is currently in a meeting and will be stopping by in 5 minutes. Spend the next 5 minutes looking through her email and pick out 3 messages she should deal with between meetings. Here are some things that Lori Adler would consider email priorities (in no particular order):

- People trying to make plans or things that affect her schedule
- People asking her for something or for advice
- Making timely responses in general, and especially to people with whom she has a close relationship

Write down the message numbers of emails as you read them

____

____

____

____

Write down the message numbers of the 3 messages you choose

1. _____________
2. _____________
3. _____________

WAIT FOR ADMINISTRATOR BEFORE GOING ON

~ Post-Study Questionnaire ~

1. When you first open up your email, do you have a good sense of the social context of your inbox? What are your expectations for a tool that helps with this?

2. Did the additional information of the social webmail client change the way you dealt with the email? If yes, in what way?

3. What did you like or not like about the social webmail interface?

4. How confident do you feel about your choices of what email what most important? What things did you look for in making the decision?

Appendix C: Message Set Composition

STATS ON EACH DATASET

freq	1=bigfont	2=med	3=small			
context	1=inquire	2=keepit	3=interest	4=plan	5=info/share	
ttr	1=least	2= <1wk	3= <1wk	4= <2wks	5= < 2wks	6=2weeks

inbox	msg num	freq	M context	H context	ttr
3	**1**	1	5	1	6
3	**2**	1	5	5	4
3	**3**	3	5	4	2
3	**4**	3	5	5	2
3	**5**	3	5	5	2
3	**6**	3	5	5	2
3	**7**	1	5	1	6
3	**8**	3	5	5	2
3	**9**	3	5	5	2
3	**10**	3	5	5	2
3	**11**	3	5	1	2
3	**12**	1	5	5	6
3	**13**	1	5	1	6
3	**14**	1	1	5	6
3	**15**	1	5	1	6
3	**16**	1	5	5	6
3	**17**	1	5	5	6
3	**18**	1	5	5	6
3	**19**	3	5	3	2
3	**20**	2	5	3	2
3	**21**	1	2	5	6
3	**22**	1	1	5	6
3	**23**	1	5	5	5
3	**24**	1	5	5	5
1	**1**	1	4	1	6
1	**2**	1	2	5	6
1	**3**	3	5	5	4
1	**4**	1	2	5	6
1	**5**	1	5	5	6
1	**6**	1	5	5	6
1	**7**	3	5	1	1
1	**8**	2	3	1	2
1	**9**	1	5	5	6
1	**10**	1	5	5	6

inbox	msg num	freq	M context	H context	ttr
1	**11**	1	2	5	6
1	**12**	1	5	5	6
1	**13**	1	5	5	6
1	**14**	1	5	5	2
1	**15**	3	5	5	1
1	**16**	1	5	5	2
1	**17**	1	5	5	6
1	**18**	1	5	5	6
1	**19**	1	5	4	2
1	**20**	1	5	5	6
1	**21**	1	5	1	6
1	**22**	2	5	5	6
1	**23**	3	3	1	1
1	**24**	1	5	5	6
2	**1**	1	5	4	3
2	**2**	3	5	3	4
2	**3**	3	3	5	3
2	**4**	3	5	5	2
2	**5**	3	3	5	2
2	**6**	1	1	5	1
2	**7**	3	5	5	1
2	**8**	3	5	5	1
2	**9**	1	5	5	6
2	**10**	1	5	1	1
2	**11**	3	3	3	2
2	**12**	1	5	4	6
2	**13**	3	5	5	1
2	**14**	3	5	4	1
2	**15**	1	5	5	6
2	**16**	1	5	5	6
2	**17**	3	5	5	1
2	**18**	3	5	5	1
2	**19**	2	5	5	2
2	**20**	3	5	5	2
2	**21**	3	5	5	1
2	**22**	3	5	5	1
2	**23**	2	5	5	2
2	**24**	2	3	1	2

SUMMARY OF EACH INBOX

Box	close	quick	inq	kit	inter	plan	info
3	9	10	5	0	2	1	16
1	4	7	5	0	0	1	18
2	14	17	2	0	2	3	17