Intelligent Email: Aiding Users with AI

Mark Dredze[1], **Hanna M. Wallach**[2], **Danny Puller**[1], **Tova Brooks**[1]
Josh Carroll[1], **Joshua Magarick**[1], **John Blitzer**[1], **Fernando Pereira**[1]

[1] Department of Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104, USA
{mdredze,puller,tmbrooks,carrollk,
magarick,blitzer,pereira}@seas.upenn.edu

[2] Department of Computer Science
University of Massachusetts, Amherst
Amherst, MA 01003, USA
wallach@cs.umass.edu

Intelligent Email

Email occupies a central role in the modern workplace. This has led to a vast increase in the number of email messages that users are expected to handle daily. Furthermore, email is no longer simply a tool for asynchronous online communication—email is now used for task management, personal archiving, as well both synchronous and asynchronous online communication (Whittaker and Sidner 1996). This explosion can lead to "email overload"—many users are overwhelmed by the large quantity of information in their mailboxes. In the human–computer interaction community, there has been much research on tackling email overload. Recently, similar efforts have emerged in the artificial intelligence (AI) and machine learning communities to form an area of research known as *intelligent email*.

In this paper, we take a user-oriented approach to applying AI to email. We identify enhancements to email user interfaces and employ machine learning techniques to support these changes. We focus on three tasks—summary keyword generation, reply prediction and attachment prediction—and summarize recent work in these areas.

Keyword Summarization

Email inboxes typically display a limited amount of information about each email, usually the subject, sender and date. Users are then expected to perform email triage—the process of making decisions about how to handle these emails—based on this information. In practice, such limited information is often insufficient to perform good triage and can lead to missed messages or wasted time. Additional concise and relevant information about each message can speed up the decision-making process and reduce errors.

Muresan, Tzoukermann, and Klavans (2001) introduced the task of keyword summarization, where keywords that convey the gist of an email in just a few words are generated for each email message. The user can quickly glance at these email summary keywords when checking the subject and sender information for each message. Muresan, Tzoukermann, and Klavans used a two-stage supervised learning system to generate summary keywords. Unfortunately, supervised learning techniques rely on the availability of large numbers of user-specific annotated emails for training. In contrast, we use an unsupervised approach based on latent concept models of a user's mailbox (Dredze et al. 2008b). This requires no annotated training data and generates keywords that describe each message in the context of other related messages in the user's mailbox.

The key insight behind our approach is that a good summary keyword for an email message is not simply a word unique to that message, but a word that relates the message to other topically similar messages. Consider the following example:

> Hi John, Let's meet at 11:15am on Dec 12 to discuss the Enron budget. I sent it to you earlier as budget.xls.

The words "11:15am" and "budget.xls" may do a a good job of distinguishing this email from others in John's inbox, but they are too specific to capture the gist of the email and may confuse the user by being too obscure. In contrast, "John" and "Enron" may occur in many other messages in John's inbox. This makes them representative of John's inbox as a whole, but too general to provide any useful information regarding this particular message's content. A good summary keyword for email triage must strike a middle ground between these two extremes, and be

- specific enough to describe this message but common across many emails,
- associated with coherent user concepts, and
- representative of the gist of the email, thereby allowing the user to make informed decisions about the message.

To select keywords that satisfy all three requirements, we use two well-known latent concept models to construct a representation of the underlying topics in each user's mailbox: latent semantic analysis (LSA) (Deerwester et al. 1990) and latent Dirichlet allocation (LDA) (Blei, Ng, and Jordan 2003). These topics are then used to find summary keywords that describe each message in the context of other topically similar messages in the user's mailbox, rather than selecting keywords based on a single message in isolation.

We compare two methods for selecting keywords, each of which may be used in conjunction with either LSA or LDA. The first method, based on the query–document similarity

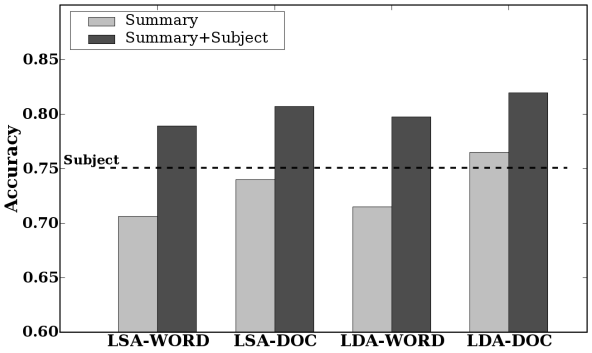

Figure 1: The accuracy averaged across 7 Enron users for automated foldering using the message subject, the four summary methods, and subjects and summaries combined.

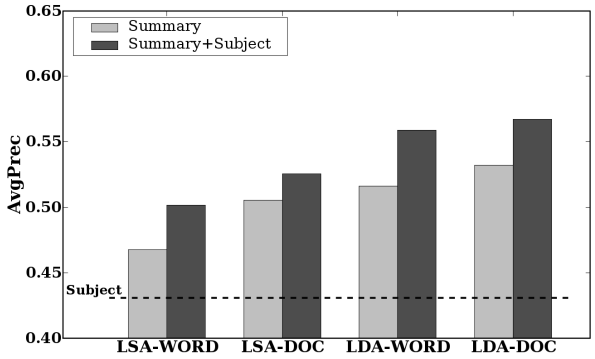

Figure 2: Average precision across 7 Enron users for recipient prediction using the message subject, the four summary methods, and subjects and summaries combined.

metric used in information retrieval, treats each candidate keyword as a one-word query. The similarity between the keyword and an email message is then computed via the latent topics. The second method is based on word association and involves choosing as keywords those words that are most closely associated with the words that occur in the message in question. Association is computed using the latent topics. A more detailed explanation of these methods can be found in Dredze et al. (2008b).

Results

The keyword generation methods described in the previous section—*LSA-doc*, *LDA-doc*, *LSA-word*, *LDA-word*—were evaluated by generating summaries for seven users selected from the Enron data set (Klimt and Yang 2004). Keyword quality was assessed using two proxy email prediction tasks: automated foldering and recipient prediction. These tasks simulate the sorts of decisions a user would make using keywords and usually rely on the entire message body. In all generation experiments, message bodies were replaced with the generated summaries. Keywords generated using term frequency–inverse document frequency (TF-IDF), along with complete message bodies, were used as lower and upper baselines. In every experiment, our keyword generation methods improved over the TF-IDF baseline, and in some cases performed better than using the entire message.

In addition to evaluating summary keywords as an approximation to message content, we also examined the extent to which keywords provide additional information over message subject lines (figures 1 and 2). Every one of our generation methods produced keywords that, when combined with subject lines, improved performance over using subject lines alone. In some cases the keywords alone resulted in better performance than the subject lines.

These results indicate that summary keywords generated using LSA- and LDA-based methods provide a good representation of email content. Furthermore, these keywords do better at summarizing messages for foldering and recipient prediction tasks than sender-written subject lines. Combining summary keywords with email subject lines significantly increases the amount of useful information available to the user when making email triage decisions.

Reply Prediction

In a large mailbox, identifying messages that need a reply can be time-consuming and error-prone. Reply management systems solve this problem by providing users with tools to manage outstanding reply commitments (Dredze et al. 2008a). We present a prototype interface that clearly indicates which messages require a reply and allows users to manage messages, marking them as *replied* or *needs reply*, as well as displaying all outstanding reply commitments. A screen shot of the interface, as implemented in the Mozilla Foundation's Thunderbird mail client, is shown in figure 3.

Underlying the interface is a reply predictor system, which automatically identifies messages that need a reply. This task is a binary classification problem, where each email is labeled by the system with either a positive label (*needs reply*) or a negative label (*does not need reply*).

The biggest challenge in creating a reply predictor is identifying the most appropriate way to represent email. Other email classification tasks typically use bag-of-words representations of message bodies and subject lines, with quoted text from previous messages removed, combined with features that indicate information such as message sender and recipients (Segal and Kephart 1999; Cohen, Carvalho, and Mitchell 2004; Carvalho and Cohen 2007). While these features are clearly important for reply prediction too, additional features are also needed. Consider a user who sends a message to Merrick and Tahlia with the subject, "Visiting client schedule." If Tahlia is only CCed on the message, then it is likely that a reply is not expected from her. However, Merrick, the primary recipient, is expected to reply. In other words, the same message can receive two different classifi-

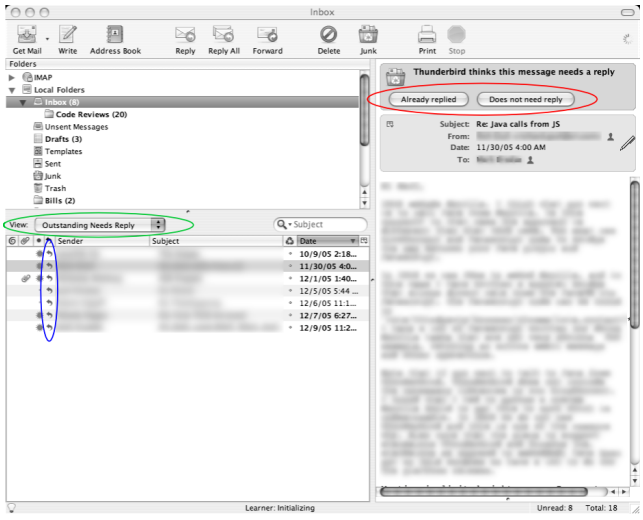

Figure 3: The reply management interface in Thunderbird has an additional column (4th from left; blue) in the bottom left pane that displays a reply arrow next to messages that need a reply. Two buttons (red) above the message contents read: "Already replied," which marks the reply as completed and "Does not need a reply," which corrects the system. Messages that still need a reply are viewed by selecting the "Outstanding Needs Reply" view (green).

cations, depending on the user's role in the message.

To address this problem, we introduce user-specific *relational features* constructed from user profiles. Each user profile includes the total number of sent and received messages for that user, the user's address book, their supervisor or boss (user-provided), as well as email address and domain. Features constructed using these profiles include *I appear in the CC list*, *I frequently reply to this user*, and *sender is in address book*. These sorts of relational features have two advantages: Firstly, general behavior patterns can be learned, such as overall tendencies towards address book contacts. Secondly, a system trained on one user can be adapted to a new user without any additional training data, since the features capture general relationships and trends, not specific email addresses or names. More information about features can be found in Dredze et al. (2008a).

Evaluation

The reply predictor was evaluated on four users' mailboxes (2391 messages total). Messages were hand-labeled as either positive (*needs reply*) or negative (*does not need reply*). Ten randomized runs were performed, each with an 80–20 training–test split. Results for each user are shown in table 1, measured using precision, recall and F_1 score. The precision scores are much higher than the recall scores, indicating that there are many messages that are difficult to identify as needing a reply. As an upper-level baseline, inter-annotator agreement was computed for Tahlia and Jaeger's email, yielding an F_1 score of 0.77. This is similar to the scores obtained using the reply predictor, demonstrating that the system achieves performance similar to that of a human.

User	*Recall*	*Precision*	F_1
Jaeger	0.42	0.55	0.47
Jarvis	0.67	0.77	0.71
Merrick	0.68	0.83	0.74
Tahlia	0.77	0.76	0.77
Average	0.64	0.73	0.67

Table 1: Reply prediction results for four email users, averaged over ten randomized trials.

Test	*Self*	*Recall*	*Precision*	F_1	Δ
Jaeger	0.47	0.56	0.72	0.63	+0.16
Jarvis	0.71	0.52	0.86	0.65	-0.06
Merrick	0.74	0.68	0.73	0.71	-0.03
Tahlia	0.77	0.77	0.59	0.67	-0.10

Table 2: A cross-user evaluation for four users averaged over ten randomized runs. Each classifier is trained on three users and tested on the fourth using relational and task specific features. Δ is the F_1 difference between this experiment and *self*, the full classifier trained on the user's data only.

The reply predictor was also evaluated in a cross-user setting to determine its effectiveness on a new user with no training data. The predictor was trained on three users while the fourth was retained for testing. Baseline features, such as bag-of-words features and address-specific features, were not effective for cross-user evaluation, so they were omitted, leaving only relational and task-specific features. Results are shown in table 2, indicating that the system does well at predicting replies even when no user-specific data is available. This is extremely useful for developing practical systems.

Attachment Prediction

Email attachments are a convenient way to transfer files and an essential part of collaborative projects. Unfortunately, it is easy for email senders to forget to attach a relevant document, creating workflow disruption, especially when the sender is offline by the time the email is received.

Email interfaces could assist users by displaying a sidebar of possible attachments, highlighting the "attach" icon, and warning the user about missing attachments before sending the message. We present an attachment prediction system for detecting whether a message should contain an attachment (Dredze et al. 2008a). This binary classification task is tackled using a similar approach to that of reply prediction.

Successful attachment prediction is dependent on both message content and other features, such as relational and task-specific features. As with reply prediction, many relational features can be generated from user profiles that include information such as the number of sent and received messages for each user, and the percentage of messages that contained an attachment. These profiles and resultant features capture high-level patterns of attachment behavior. Additional task-

Split	*System*	*Precision*	*Recall*	F_1
User	Rule Based	0.8223	0.4490	0.5808
	Learning	0.8301	0.6706	0.7419†
Cross-User	Rule Based	0.8223	0.4490	0.5808
	Learning	0.7981	0.5618	0.6594∗

Table 3: Attachment prediction results on Enron email for the rule based and learning systems. Numbers are aggregate results across ten-fold cross validation. $*$ and † indicate statistical significance at $p = .01$ and $p = .001$ respectively against the baseline using McNemar's test.

specific features include the position of the word "attach" in the email, the presence of other words in close proximity to the word "attach," and the length of the message body.

Evaluation

The attachment predictor was evaluated on 15,000 randomly selected Enron sent emails from 144 users (7% have attachments). A rule-based system using the stem "attach" was used as a baseline. Results (table 3) indicate that although the rule-based system has high precision, it fails to find even half of the emails with attachments. In contrast, our prediction system achieves slightly higher precision and much higher recall, finding two-thirds of the attachment messages. We also tested the system in a cross-user setting. Since attachment prediction is content dependent, system performance is worse in the cross-user setting, but is still higher than the rule-based baseline system.

Related Work

There are several systems that assist email users. Neustaedter et al. (2005) created an interface that provides social information about emails, allowing users to select messages based on social relationships. Segal and Kephart (1999) investigated systems for automated foldering that assist users with sorting and moving messages. Other work by Wan and McKeown (2004) addressed email summarization to assist with processing large mailboxes, while thread arcs can be used to position a new message in the context of a conversation (Kerr 2003). Carvalho and Cohen (2007) simplify message composition by suggesting possible recipients. Finally, many researchers have addressed the presentation of email, focusing on email as a tool for task management as well as communications (Kushmerick et al. 2006).

Conclusions

In this paper, we surveyed three ways to assist user email decisions using artificial intelligence. Some of this research is already being used in intelligent email interfaces: the reply and attachment predictors are part of IRIS (Cheyer, Park, and Giuli 2005) and the attachment predictor is also being integrated into a government email client. This work demonstrates that representations of email and user behavior play a significant role in building effective intelligent email interfaces. Furthermore, artificial intelligence enables email systems to better respond to and predict user behavior.

Acknowledgements

This work was supported in part by the Center for Intelligent Information Retrieval and in part by DoD contract #HM1582-06-1-2013. Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsor.

References

Blei, D.; Ng, A.; and Jordan, M. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research* 3:993–1022.

Carvalho, V. R., and Cohen, W. 2007. Recommending recipients in the enron email corpus. Technical Report CMU-LTI-07-005, Carnegie Mellon University, Language Technologies Institute.

Cheyer, A.; Park, J.; and Giuli, R. 2005. Iris: Integrate. relate. infer. share. In *The Semantic Desktop Workshop at the International Semantic Web Conference*.

Cohen, W. W.; Carvalho, V. R.; and Mitchell, T. M. 2004. Learning to classify email into "speech acts". In *Empirical Methods in Natural Language Processing (EMNLP)*.

Deerwester, S.; Dumais, S. T.; Furnas, G. W.; Landauer, T. K.; and Harshman, R. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science* 41(6):391–407.

Dredze, M.; Brooks, T.; Carroll, J.; Magarick, J.; Blitzer, J.; and Pereira, F. 2008a. Intelligent email: Reply and attachment prediction. In *Intellgent User Interfaces (IUI)*.

Dredze, M.; Wallach, H.; Puller, D.; and Pereira, F. 2008b. Generating summary keywords for emails using topics. In *Intellgent User Interfaces (IUI)*.

Kerr, B. 2003. Thread arcs: An email thread visualization. In *Symposium on Information Visualization (INFOVIS)*.

Klimt, B., and Yang, Y. 2004. The Enron corpus: A new dataset for email classification research. In *ECML*.

Kushmerick, N.; Lau, T.; Dredze, M.; and Khoussainov, R. 2006. Activity-centric email: A machine learning approach. In *American National Conference on Artificial Intelligence (AAAI)*.

Muresan, S.; Tzoukermann, E.; and Klavans, J. L. 2001. Combining linguistic and machine learning techniques for email summarization. In *Conference on Computational Natural Language Learning (CONLL)*.

Neustaedter, C.; Brush, A. B.; Smith, M. A.; and Fisher, D. 2005. The social network and relationship finder: Social sorting for email triage. In *Conference on Email and Anti-Spam (CEAS)*.

Segal, R., and Kephart, J. 1999. Mailcat: An intelligent assistant for organizing e-mail. In *Proceedings of the Third International Conference on Autonomous Agents*.

Wan, S., and McKeown, K. 2004. Generating overview summaries of ongoing email thread discussions. In *Conference on Computational Linguistics (COLING)*.

Whittaker, S., and Sidner, C. 1996. Email overload: exploring personal information management of email. In *Computer-Human Interaction (CHI)*.