# Activity-Centric Email: A Machine Learning Approach

**Nicholas Kushmerick**[1] **Tessa Lau**[2] **Mark Dredze**[3] **Rinat Khoussainov**[4]
[1]University College Dublin, nick@ucd.ie [2]IBM Almaden Research Center, tessalau@us.ibm.com
[3]University of Pennsylvania, mdredze@seas.upenn.edu [4]University of Portsmouth, rinat@ee.port.ac.uk

## Activity Management

Our use of ordinary desktop applications (such as email, Web, calendars) is often a manifestation of the activities with which we are engaged (Moran, Cozzi, & Farrell 2005). Planning a conference trip involves sending travel expense forms, and visits to airline and hotel sites. Renovating a kitchen involves sketches, product specifications, emails with the architect and spreadsheets for tracking expenses. Every enterprise has (often implicit) processes for managing customer queries, requesting maintenance, hiring a new employee, purchasing equipment, and so on.

Unfortunately, ordinary desktop applications do not know anything about these activities. Within an enterprise, many activities have been formalized into business workflows such as hiring or ordering equipment. However, the way people interact with these workflows is often through email and desktop applications. If these applications are not aware of the activity context, people bear the burden of organizing their information into activities, typically using crude techniques such as manual search, file directories, and email folders/threads.

Email has emerged as the primary tool for people to communicate about their work and manage activities. Motivated by the importance of email in conducting activities, we have recently developed several machine learning algorithms for automatically discovering and tracking activities in email. We observe that activities come in many forms, from structured workflows to informal person-to-person communication. In this paper, we summarize our efforts to provide automated assistance with two types of activities: rigid structured activities, and unstructured conversational activities.

We first discuss highly structured activities such as e-commerce transactions. A consumer purchasing an item may receive email messages confirming the order, warning of a delay and then a shipment notification. Existing email clients do not understand this structure, so users must manage their transactions by sifting through lists of messages. As a step to providing high-level support for structured activities, we consider the problem of automatically learning an activity's structure. This structure could be used to support features such as notifying the user if an item failed to ship within an expected length of time. We formalize such structured activities as finite-state automata (where states correspond to the status of the process, and transitions represent messages sent between participants), and propose several unsupervised machine learning algorithms in this context. A paper describing this work was awarded honorable mention for best paper at the Conference on Intelligent User Interfaces (Kushmerick & Lau 2005).

Second, we discuss less-structured activities such as organizing meetings or collaboratively editing documents. We describe machine learning approaches to activity discovery (i.e., grouping messages according to activities) and semantic message analysis (i.e., extracting metadata about how messages within an activity relate to one another and to the progress of the activity). One key innovation compared to related work is that we use a form of social network analysis, in addition to simply the content of the messages, to automatically categorize messages by activity. In other work, we exploit the relational structure of activity discovery and message analysis to solve the two problems simultaneously. Instead of attacking these two problems separately, in our synergistic collective classification approach, activity discovery is used to assist in semantic message analysis, and vice versa. Papers describing this work were presented at the Conference on Email and Antispam (Khoussainov & Kushmerick 2005) and the Conference on Intelligent User Interfaces (Dredze, Lau, & Kushmerick 2006).

## Intelligent User Interfaces

Several papers in the HCI community have addressed activity management in email from an interface perspective (Bellotti *et al.* 2005). To motivate our automated systems, we designed several interfaces to visualize activities. Our vision is to provide activity-centric (rather than message-centric) tools that enable users to manage their activities as first class citizens, rather than as isolated messages.

For example, Fig. 1 (left) shows an extension to the Thunderbird email client that displays an activities pane in the lower-left corner. This interface shows how an email client can be enhanced to provide activity awareness in the context of the user's existing email. As the user reads an email message, the activity pane shows activities that the user and the sender of that message are both involved in. This activity list is prioritized using the SimOverlap metric (Dredze, Lau, &

Kushmerick 2006), which compares the set of people in the message against the people in each activity to rank activities relative to a given message.

The intent of this interface is to give users the ability to manage their activities directly from their email client. A user can drag messages into an activity, send mail to all the people in an activity with a single click, and create a new activity based on the recipients of an email message.

However, as the number of activities increases, the list of activities in the activity pane grows, and it will be harder to find the activity the user is looking for. Our experience with this user interface led us to develop algorithms for automatically classifying email into activities, both to provide better ranking functions in this activity display, and to support automated association of email with activities.

Given a set of messages that are associated with an activity, one of our research goals is to automatically extract the structure of that activity. For example, an activity might include a formalized process, so the structure could include the steps in the process as well as the user's current state in completing that process.

Fig. 1 (middle) shows an interfacew for managing structured activities. This interface organizes messages by activity, and displays updates (in bold) for each activity, e.g. that there is action on the first "patent" activity and the second "PBC" activity. The colored-dot diagrams show a finite-state-machine representation of the structure of each activity; the pink dot shows where the new message occurred within the context of the rest of the activity. To support this user interface, we have developed algorithms (summarized below) that, given a set of messages, infer the structure of the underlying activity as a finite-state automaton, and track activities as they unfold (Kushmerick & Lau 2005).

Fig. 1 (right) shows a third activity-centric email client that builds on the ideas in the previous interface. Not all activities are instances of formalized workflows; many are less-formal, involving human-to-human communication and negotiation. This client shows how both types of activities can be represented in a single interface, and managed together. The upper-right activity browser shows all the activities a user is involved with, grouped by type (e.g., eBay or meeting scheduling). In support of this interface, we have extended our work on modelling activities as finite-state automata to handle not only structured computer-human activities (e.g., e-commerce transactions) but also informal human-human activities (e.g., meeting scheduling, collaborative document editing) (Khoussainov & Kushmerick 2005). As we describe below, our approach is to use speech act detection (Cohen, Carvalho, & Mitchell 2004) as well as inter-message relationships to infer models of these less-structured activities.

These user interface examples motivate our work on learning-based algorithms to assist with intelligent email activity management. The following sections provide an overview of the algorithms we have developed.

## Structured Activities

Structured processes comprise an important class of activities in email. For instance, an employee in an organization with a centralized hiring process receives automatically-generated messages reminding her of an upcoming interview, requesting feedback on the candidate after the interview, and notifying her of the final decision. A consumer purchasing an item from an e-commerce vendor may receive messages that confirm the order, notification of a delay or that the items have shipped.

Our goal is to provide a high-level interface to enable users to interact with their activities directly, allowing a consumer to see how many e-commerce transactions are pending, rather than searching through emails. The first step is automatically recognizing structured processes in email, and tracking the user's progress through these processes as new messages arrive.

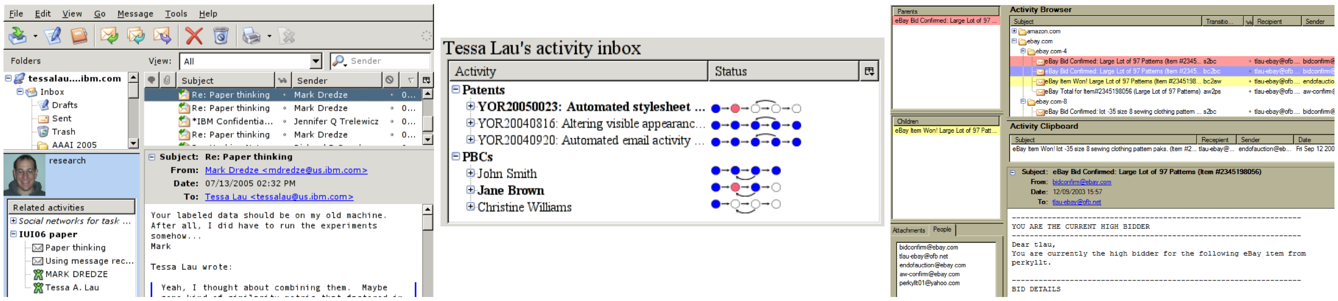
We assume that a user participates in a variety of distinct classes of activities (e.g., purchases from Amazon, auctions at eBay, recruitment activities). We formalize activities as finite-state automata called *process models*. We create a distinct process model for each type of activity (e.g., one model for Amazon, a second model for eBay, a third for the personnel department, etc). Each activity consists of the messages related to performing a single transaction with that vendor (e.g., ordering a book from amazon, completing an auction on eBay, etc.).

States in a process model correspond to the internal status of the process, and email messages correspond to *transitions* between process states. For example, an Amazon purchase might be in an "order submitted" state; when the order is shipped, the state changes to "done" and Amazon sends a message to the purchaser to indicate this transition.

Our work focuses on four distinct sub-problems. **Activity identification** is the task of partitioning a set of messages into activities. For example, in our experiments, activities corresponds to e-commerce transactions; the three messages received from freshdirect.com in Fig. 2 would be identified as relating to the same activity. The idea is to cluster messages based on automatically-detected *unique identifiers*: strings of unusual alphanumeric characters such as invoice numbers or employee serial numbers, that are common to all messages in a single activity but do not appear in other activities. Our algorithm is unsupervised: we do not rely on the user to provide labelled training data, such as a sample message from each activity, or even the total number of activities to be discovered.

**Transition identification** is the task of partitioning a set of messages according to which process model transition they correspond. For example, the algorithm would partition the freshdirect.com messages into those relating to order confirmation, order modification, etc. The key idea behind transition identification is to look for long common subsequences of text to cluster messages into the transition they represent. For example, all order confirmation messages tend to contain the same boilerplate text thanking the user for having placed an order, whereas all shipment notification messages contain a different sequence of text. As with activity identification, our algorithm does not need training data.

**Model induction** is the task of automatically generating the process model. For example, given a few activities in



| Subject | Sender |
|---|---|
| Re: Paper thinking | Mark Dredze |
| Re: Paper thinking | Mark Dredze |
| *IBM Confidentia... | Jennifer Q Trelewicz |
| Re: Paper thinking | Mark Dredze |

**Subject:** Re: Paper thinking
**From:** Mark Dredze <mdredze@us.ibm.com>
**Date:** 07/13/2005 02:32 PM
**To:** Tessa Lau <tessalau@us.ibm.com>

```
Your labeled data should be on my old machine.
After all, I did have to run the experiments
somehow...
Mark

Tessa Lau wrote:

 Yeah, I thought about combining them.  Maybe
```

Tessa Lau's activity inbox

| Activity | Status |
|---|---|
| **Patents** | |
| **YOR20050023: Automated stylesheet ...** | |
| YOR20040816: Altering visible appearanc... | |
| YOR20040920: Automated email activity ... | |
| **PBCs** | |
| John Smith | |
| **Jane Brown** | |
| Christine Williams | |

bidconfirm@ebay.com
tlau-ebay@ofb.net
endofauction@ebay.com
aw-confirm@ebay.com
perkyllt01@yahoo.com

| Subject | Transitio... | Recipient | Sender |
|---|---|---|---|
| eBay Bid Confirmed: Large Lot of 97 Patterns (Item #2345... | s2bc | tlau-ebay@ofb.... | bidconfirm@ |
| eBay Bid Confirmed: Large Lot of 97 Patterns (Item #2345... | bc2bc | tlau-ebay@ofb.... | bidconfirm@ |
| eBay Item Won! Large Lot of 97 Patterns (Item #2345198... | bc2aw | tlau-ebay@ofb.... | endofauctio |
| eBay Total for Item#2345198056 (Large Lot of 97 Patterns) | aw2ps | tlau-ebay@ofb.... | aw-confirm@ |
| eBay Bid Confirmed: lot -35 size 8 sewing clothing pattern ... | s2bc | tlau-ebay@ofb.... | bidconfirm@ |

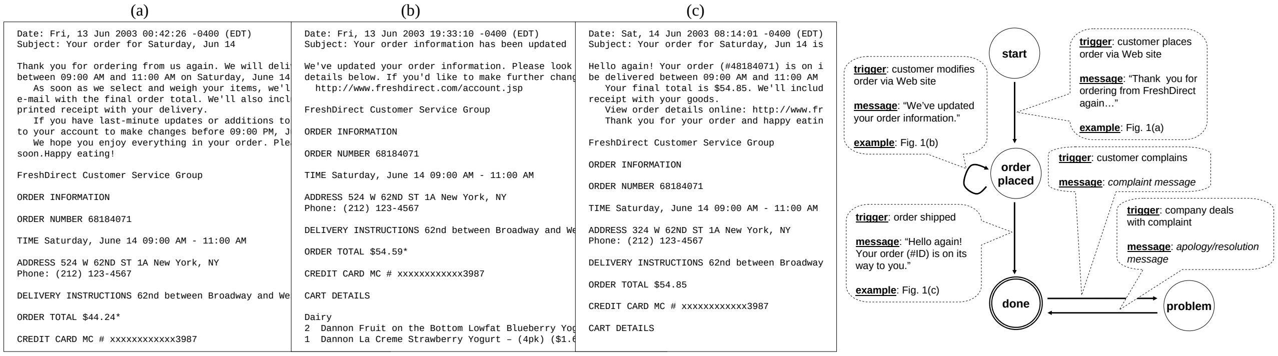
| Subject | Recepient | Sender | Date |
|---|---|---|---|
| eBay Item Won! lot -35 size 8 sewing clothing pattern paks. (Item #2... | tlau-ebay@... | endofauction@eb... | Fri Sep 12 200 |

**Subject:** eBay Bid Confirmed: Large Lot of 97 Patterns (Item #2345198056)
**From:** bidconfirm@ebay.com
**Date:** 12/09/2003 15:57
**To:** tlau-ebay@ofb.net

```
-----------------------------------------------------------------
YOU ARE THE CURRENT HIGH BIDDER
-----------------------------------------------------------------
Dear tlau,
You are currently the high bidder for the following eBay item from
perkyllt.

-----------------------------------------------------------------
BID DETAILS
```

Figure 1: Three activity-centric email clients: (left) note the contextual activity pane in the lower left; (middle) six instances of two activities, with the process structure and the current state, empty bubbles showing future steps; (right) a list of activities, associated messages, relationships between messages within the current activity, people and attachments involved.


(a)

```
Date: Fri, 13 Jun 2003 00:42:26 -0400 (EDT)
Subject: Your order for Saturday, Jun 14

Thank you for ordering from us again. We will deli
between 09:00 AM and 11:00 AM on Saturday, June 14
   As soon as we select and weigh your items, we'l
e-mail with the final order total. We'll also incl
printed receipt with your delivery.
   If you have last-minute updates or additions to
to your account to make changes before 09:00 PM, J
   We hope you enjoy everything in your order. Ple
soon.Happy eating!

FreshDirect Customer Service Group

ORDER INFORMATION

ORDER NUMBER 68184071

TIME Saturday, June 14 09:00 AM - 11:00 AM

ADDRESS 524 W 62ND ST 1A New York, NY
Phone: (212) 123-4567

DELIVERY INSTRUCTIONS 62nd between Broadway and We

ORDER TOTAL $44.24*

CREDIT CARD MC # xxxxxxxxxxxx3987
```

(b)

```
Date: Fri, 13 Jun 2003 19:33:10 -0400 (EDT)
Subject: Your order information has been updated

We've updated your order information. Please look
details below. If you'd like to make further chang
  http://www.freshdirect.com/account.jsp

FreshDirect Customer Service Group

ORDER INFORMATION

ORDER NUMBER 68184071

TIME Saturday, June 14 09:00 AM - 11:00 AM

ADDRESS 524 W 62ND ST 1A New York, NY
Phone: (212) 123-4567

DELIVERY INSTRUCTIONS 62nd between Broadway and We

ORDER TOTAL $54.59*

CREDIT CARD MC # xxxxxxxxxxxx3987

CART DETAILS

Dairy
2  Dannon Fruit on the Bottom Lowfat Blueberry Yog
1  Dannon La Creme Strawberry Yogurt – (4pk) ($1.6
```

(c)

```
Date: Sat, 14 Jun 2003 08:14:01 -0400 (EDT)
Subject: Your order for Saturday, Jun 14 is

Hello again! Your order (#48184071) is on i
be delivered between 09:00 AM and 11:00 AM
   Your final total is $54.85. We'll includ
receipt with your goods.
   View order details online: http://www.fr
   Thank you for your order and happy eatin

FreshDirect Customer Service Group

ORDER INFORMATION

ORDER NUMBER 68184071

TIME Saturday, June 14 09:00 AM - 11:00 AM

ADDRESS 324 W 62ND ST 1A New York, NY
Phone: (212) 123-4567

DELIVERY INSTRUCTIONS 62nd between Broadway

ORDER TOTAL $54.85

CREDIT CARD MC # xxxxxxxxxxxx3987

CART DETAILS
```

start
trigger: customer places order via Web site message: "Thank you for ordering from FreshDirect again..." example: Fig. 1(a)
trigger: customer modifies order via Web site message: "We've updated your order information." example: Fig. 1(b)
order placed
trigger: customer complains message: complaint message
trigger: order shipped message: "Hello again! Your order (#ID) is on its way to you." example: Fig. 1(c)
trigger: company deals with complaint message: apology/resolution message
done
problem

Figure 2: An example of a highly structured activity. (left) Purchases from freshdirect.com result in messages that confirm (a) the initial order, (b) an order change and (c) delivery. (right) A finite-state model of freshdirect.com activities.

Fig. 2(a-c), the task is to derive the model in Fig. 2(d). We formalize this problem as that of inducing a regular grammar from positive examples of the strings generated by that grammar, and apply the MDI algorithm (Thollard, Dupont, & de le Higuera 2000) to solve it.

Finally, **message classification** is the task of assigning incoming messages to their transitions. For example, given the model in Fig. 2(d), the task is to assign message Fig. 2(a) to the edge from "start" to "order placed", etc. We have treated this as a text classification problem, and used a standard SVM classifier trained on the set of messages labelled by transition to predict which transition each new message is most similar to.

Kushmerick & Lau (2005) describe these algorithms in more detail, and demonstrates empirically that, even when the activities, transitions and process models are not learned exactly, the approach can accurately generate useful predictions, such as determining whether an activity is complete, or explaining what will happen next.

## Informal Activities

Most activities involve noisy and ambiguous messages sent between people rather than messages arising from highly structured activities generated by a computer. We therefore extend our finite-state approach to represent many informal activities using the idea of speech acts. *Speech acts* (Cohen, Carvalho, & Mitchell 2004) identify the core semantic intent of messages (e.g. making a request, delivering information, committing to some future action) and can be viewed as the analogs of transitions for informal activities. Our work focuses on three sub-problems: *activity classification* involves assigning incoming messages to existing activities; *activity discovery* means clustering past messages into activities; and *message analysis* involves assigning the speech acts. Dredze, Lau, & Kushmerick (2006) concentrate on activity classification; Khoussainov & Kushmerick (2005) address activity discovery and message analysis.

**Activity classification.** Given a set of existing activities $\mathcal{A}$, the null activity $\epsilon$, and a message $M$, the goal of activity classification is to output a probability distribution $P$ over activities such that $\sum_{i \in \mathcal{A} \cup \{\epsilon\}} P(i|M) = 1$. Note that this is an incremental learning problem: the set of class labels changes over time as new activities are created. However, at a particular point in time, the classifier is only expected to be able to predict activities that have previously been created. The intent is that $\epsilon$ is a distinguished activity label meaning the message is not associated with any activity.

The goal is to automatically populate activities with the emails related to them as seen in our interfaces described above. Our approach leverages two characteristics of activities: activities connect groups of people together, and activity-related messages tend to center around particular topics. We have defined two similarity metrics, SimOverlap and SimSubset, that compare the set of people in a message against the set of people in an activity, in order to determine how similar a message is to an activity. For content, we have used a variant of LSI known as iterative residual rescaling (Ando & Lee 2001) to create another similarity metric, SimContent, that determines how similar a message is to an activity based on the words in the message and the words in the activity. Based on these metrics, we have developed algorithms that learn from previously-seen messages and predict, for a new incoming message, which activities are most relevant to the new message. We found that a combined approach, which votes together the predictions of the base models, performs better than each individual model alone.

**Activity discovery & message analysis.** The goal of *activity discovery* is to group email messages into activities and to establish conversational links between messages within an activity. Note that activities need not correspond to threads: an activity can have multiple conversation threads, and a thread can be related to several activities. Likewise, organizing emails into folders can be orthogonal to activities. For instance, users may have a single folder for urgent messages (often the Inbox), while these messages can be from different activities. Moreover, activities represent ongoing work, while folders are more typically used for archival purposes.

*Message analysis* involves generating speech act metadata for individual messages in an activity that provides a link between the messages and the changes in the status of the underlying process, or the actions of the user in the underlying workflow. For example, a message described as a meeting confirmation can change the state of an activity from “Awaiting response” to “Meeting confirmed”. Of course, such semantic analysis requires making assumptions regarding what actions are available to the user in an activity, what state transitions are possible in a process, etc. Note that such assumptions do not rstrict the user, but merely allow the client to provide the advanced activity-centric capabilities.

Our approach is based on the idea that related messages in an activity provide a valuable context that can be used for semantic message analysis. Similarly, the speech act metadata in separate messages can provide relational clues that can be used to establish links between messages and subsequently group them into activities. Instead of treating these two problems separately, we propose a *synergistic iterative* approach: identifying related messages is used to assist semantic message analysis, and vice versa. Our key innovation compared to related work is that we exploit the *relational structure* of these two tasks.

In more detail, we investigate several methods for identifying relations between messages and grouping emails into activities. We use pair-wise message similarity to find potentially related messages, and hierarchical agglomerative clustering is used to group messages into activities. We extend the message similarity function to take into account not only the textual similarity between messages, but also the available structured information in email, such as send dates and message subjects. We then propose a relational learning approach to email activity management that uses relation identification for semantic message analysis and vice versa. In particular, we investigate how (a) features of related messages in the same activity can assist with classification of email speech acts, and how (b) information about message speech acts can assist with finding related messages and grouping them into activities. Combining these two methods yields an iterative relational algorithm for speech act classification and relation identification.

## Discussion

Many structured activities are managed by email. Existing email clients have no understanding of this structure, forcing users to manage their activities by manually sifting through lists of messages. As an alternative, we envision email clients that provide high-level support for activity management. The key idea is that activities should be identified and managed as entities in their own right. Examples include reply prediction, detection of completed activities, message prioritization based on the activity status, identifying dependencies, version control over attached data, and automated to-do/reminder updating. To this end, we have developed algorithms for automatically identifying and tracking activities in email, and investigated several intelligent user interfaces driven by these algorithms.

The current research leaves plenty of scope for future work. We are extending our ideas to other desktop applications (e.g., Web browsing). We are also improving the activity-tracking algorithms, and developing better models of human-to-human email. For example, we have developed an algorithm for modeling collaborative form-filling tasks as an automata whose states corespond to the subset of fields that have been filled. Finally, we will conduct a user studies to quantify the benefit of activity-centric email for real-world knowledge workers.

## References

Ando, R., and Lee, L. 2001. Iterative residual rescaling. In *Proc. Int. Conf. Information Retrieval*.

Bellotti, V.; Ducheneaut, N.; Howard, M.; Smith, I.; and Grinter, R. 2005. Quality versus quantity: e-mail-centric task management and its relation with overload. *Human-Computer Interaction* 20:89 – 138.

Cohen, W.; Carvalho, V.; and Mitchell, T. 2004. Learning to classify email into speech acts. In *Proc. Conf. Empirical Methods in Natural Language Processing*.

Dredze, M.; Lau, T.; and Kushmerick, N. 2006. Automatically classifying emails into activities. In *Proc. Conf. Intelligent User Interfaces*.

Khoussainov, R., and Kushmerick, N. 2005. Email task management: An iterative relational learning approach. In *Proc. Conf. Email and Anti-Spam*.

Kushmerick, N., and Lau, T. 2005. Automated email activity management: An unsupervised learning approach. In *Proc. Conf. Intelligent User Interfaces*.

Moran, T. P.; Cozzi, A.; and Farrell, S. P. 2005. Unified activity management: supporting people in e-business. *Commun. ACM* 48(12):67–70.

Thollard, F.; Dupont, P.; and de le Higuera, C. 2000. Probabilistic DFA inference using Kullback-Leibler divergence and minimality. In *Proc. Int. Conf. Machine Learning*.