Extracting Knowledge about Users' Activities from Raw Workstation Contents

Tom M. Mitchell
Carnegie Mellon University
Pittsburgh PA 15213
Tom.Mitchell@cs.cmu.edu

Sophie H. Wang
Carnegie Mellon University
Pittsburgh PA 15213
Sophie.Wang@cs.cmu.edu

Yifen Huang
Carnegie Mellon University
Pittsburgh PA 15213
hyifen@cs.cmu.edu

Adam Cheyer
SRI International
Melon Park, CA 94025
Adam.Cheyer@sri.com

Abstract

A long-standing goal of AI is the development of intelligent workstation-based personal agents to assist users in their daily lives. A key impediment to this goal is the unrealistic cost of developing and maintaining a detailed knowledge base describing the user's different activities, and which people, meetings, emails, etc. are affiliated with each such activity. This paper presents a clustering approach to automatically acquiring such a knowledge base by analyzing the raw contents of the workstation, including emails, contact person names, and online calendar meetings. Our approach analyzes the distribution of email words, the social network of email senders and recipients, and the results of Google Desktop Search queried with text from online calendar entries and person contact names. For each cluster it constructs, the program outputs a frame-based representation of the corresponding user activity. This paper describes our approach and experimentally assesses its performance over the workstations of three different users.

Introduction

The research reported here is motivated by our goal to build intelligent workstation assistants to help users organize their workstation contents, to automate routine tasks such as email processing, and to provide other active assistance such as anticipating the user's information needs in real time.

One essential component to providing such intelligent assistance is to have a machine-understandable description of the ongoing *activities* in which the user is involved (e.g., courses they are teaching, committees in which they participate). Workstation users rarely maintain an explicit list of these activities in a form interpretable by the workstation, yet it is often these activities that primarily determine the implicit connections between different meetings, emails, files, people, etc.

The goal of the research reported here is to automatically infer computer-understandable descriptions of a workstation user's activities, to support two uses:

1. To index the workstation contents by these inferred activities, thereby allowing the user to browse their emails, contacts, meetings, and files by activity.
2. To use these activity descriptions as the basis for knowledge-based assistance to the user. For example, given knowledge of the users' activities, a computer assistant may reason that when a new unseen email arrives and is associated with the same activity as a meeting the user is about to attend, then it is useful to interrupt the user to alert them to this new email.

This paper focuses on algorithms for automatically inferring activity descriptions from user workstation contents. Whereas much previous research has been done on clustering and social network analysis of email, we believe ours is the first attempt to automatically extract structured, logical representations of user activities directly from workstation contents. Below we briefly summarize related research, then describe our algorithms for inferring activity descriptions, and report on experiments applying these algorithms to contents from several workstations.

Related Work

A variety of researchers have explored automated analysis of email and other workstation contents. For example, machine learning approaches to automated email classification have been applied to tasks such as detecting spam (Sahami 1998), predicting where to store a message (Segal 2000), classifying the intent of the email sender (Cohen 2004), and grouping similar messages (Mock 2001). McCallum analyzed email collections with an algorithm that forms "author-recipient-topic" models, which characterize the distribution of words sent by specific authors to specific recipients, later extending this to model multiple roles in the role-author-recipient-topic model (McCallum 2005). Our earlier work on ActivityExtractor (Huang 2004) applies bag-of-words clustering to the user's email corpus to identify user activities. Kushmerick's activity management system (Kushmerick 2004) learns workflow models from example execution logs, using text classification and clustering to attach labels. Some researchers have focused on learning activity-centric collaboration through Peer-to-Peer shared objects (Geyer, 2003), and automatically classifying emails into activities (Dredze, 2006). Others have worked on discovery of personal topics to organize email (Surendran 2005).

One significant cluster of related research explores social network analysis of email collections. The most popular approach is to construct a graph where vertices represent senders or recipients of email messages, and links denote a direct email between the nodes they connect. Graph algorithms are employed to find the communities embedded in the graph (e.g., (Tyler 2003)). One end-to-end system (Culotta 2004) extracts a user's social network and then extracts its members' contact information by searching the web. Another system, EmailNet (EmailNet 2003) automatically mines email traffic across the entire organization to generate organization-wide social networks. The ReMail system (Rohall 2004), explores visualization techniques for displaying message threads and extracts important dates and message summaries.

This paper focuses, in contrast, on the automatic extraction of logical, frame-based representations of user activities, based on analyzing the user's email, calendar, and indirectly using the entire workstation contents accessible via Google Desktop Search. Our algorithm employs bag-of-words representations of each email, person and meeting to cluster these entities, together with social network analysis to refine clusters. To our knowledge, this is the first attempt to employ recent desktop search utilities as a basis for extracting descriptions of user activities.

Approach

Our approach to inferring users' activities from workstation contents is implemented in the ActivityExtractor system, which consists of three components which (a) cluster emails based on analysis of their text content, (b) perform social network analysis on each cluster to potentially subdivide it into subclusters, and (c) construct a structured representation of each cluster (activity), associating calendar meetings and person names with the activity. The system also outputs an email classifier which can be used to incrementally classify future emails into the corresponding activity.

An activity description produced automatically by ActivityExtractor is shown in Figure 1. This description is the verbatim output of ActivityExtractor, except that we have removed person names from the list of keywords to improve readability, and we have removed phone numbers from the meeting descriptions for privacy reasons. The activity represented here corresponds to one of the research projects of this workstation's user. Note the activity description includes a list of keywords, people, meetings, email senders and recipients, and emails, along with a summary of the fraction of the user's email associated with this activity, and the fraction of this activity's email authored by the user. This activity description ties together related information from disparate parts of the workstation, and can be used, for example, to suggest relevant emails, people, and email addresses when a user is viewing a particular meeting on their calendar. The activity name here was selected by ActivityExtractor from a list of 23 activity names supplied in advance by the user. The person names were selected from a list of 540 contact names on the user's workstation. Among the 245 emails that ActivityExtractor placed in this cluster, 240 were judged by the workstation user to indeed be relevant to their "CALO ActivityExtractor" activity.

Capturing and Representing Workstation-Wide Information

Evidence for user's activities exists across diverse types of workstation data, including email, online calendars, file contents, directory hierarchies, and visited web pages. Fortunately, desktop search engines such as Google Desktop Search (GDS) make it feasible to efficiently search and retrieve this kind of information from across the entire workstation.

- *Activity Name*: CALO ActivityExtractor
- *Keywords (omitting person names)*: ActivityExtractor, TFC, IRIS, clustering, heads, emails, collected, clusters, SRI, ...
- *Person Names:* Adam Cheyer (0.49), Ray Perrault (0.36), Hung Bui (0.32), Melissa Beers (0.30), James Arnold (0.28), Jack Park (0.26), Sophie Wang (0.25), Tomas Uribe (0.25), jeanne ledbetter (0.24), Leslie Pack Kaelbling (0.24),...
- *Meetings*: CALO TFC telecon (0.59), CALO phone call (0.55), SRI Meeting – Chicago (0.48), SRI TFC Telecon and quarterly rpt (0.47), SRI visit. Bill and Ray. Call Melissa Beers when arriving (0.47), CALO annual mtg at SRI (0.45), ...
- *Primary Senders*: tom.mitchell@cmu.edu (75), sophie.wang@cs.cmu.edu (20), adam.cheyer@sri.com (16), perrault@ai.sri.com (14), tgd@eecs.oregonstate.edu (13), ...
- *Primary Recipients:* tom.mitchell@cmu.edu (94), adam.cheyer@sri.com (40), william.cohen@cs.cmu.edu (35), perrault@ai.sri.com (19), ...
- *Emails*: <email125>, <email72>, ... (245 emails in total)
- *User Activity Fraction*: 245/2822=0.086 of total emails
- *User Involvement*: user authored 30% of email (default 31%)

Figure 1: An activity description created automatically by ActivityExtractor. Real-valued numbers in parentheses indicate confidences that the corresponding person or meeting is associated with this activity. Integers in parentheses indicate number of occurrences of the entity within the 245 emails associated with this activity.

In order to create a single common representation for calendar meetings, people, email addresses, and emails, ActivityExtractor creates a *word distribution vector*, or bag-of-words, to represent each such entity. The exact algorithm for generating word distribution vectors varies by the type of entity, as follows.

- Emails: the word distribution for an email consists of the counts of the words found in the email subject line and body, with functional tokens such as "Re:" and "Fwd:" removed from the subject line.
- Email addresses: the word distribution assigned to an email address is the sum of the word distributions for each email sent by, and received by that email address.

The word vectors for entities of type person name, calendar meeting, and activity title are all computed by employing Google Desktop Search (GDS) as a subroutine. The general strategy in these cases is to first construct a search query based on the entity, then construct the word distribution vector based on the returned search hits. For each document that matches the query (documents can be emails, files, web pages visited, or chat messages sent and received), GDS reports the document's name, plus a snippet of text characterizing the match. ActivityExtractor converts this set of document names and text snippets into a word distribution as follows:

- Person names: the name of the person is used as a query to GDS, and the returned document names and text snippets are used to create the word distribution.
- Meetings: the text description of the meeting (e.g., "meet with Sharon and William") is used as the GDS query. If no hits are obtained with this query, then progressive truncations of the query are considered, until a search success is obtained. (e.g., first attempt the query "meet with Sharon and William", then "meet with Sharon and", then "meet with Sharon", etc.).
- Activity names: the activity name is used as a query to GDS, to form the word distribution.

One can view the word distribution vector associated with each entity as a representation that summarizes the distribution of words that co-occur with the entity across the entire workstation. Because a common vocabulary is used to define the word distribution vectors regardless of entity type, the similarity of two arbitrary entities can be estimated by the dot product or cosine similarity between their word distribution vectors. For example, we can use this similarity metric to compute which email address is most related to a particular meeting. We can also use machine learning methods to cluster or classify these vectors, regardless of the type of entity. Below we describe the ActivityExtractor algorithm, and how it employs these descriptions to discover and describe user activities.

ActivityExtractor Algorithm

ActivityExtractor implements a family of algorithms for estimating coherent clusters of emails, meetings, person names, and email addresses that may correspond to ongoing activities. In addition to finding coherent clusters of these entities, it also analyzes these to build structured descriptions of each proposed activity, such as the description shown earlier in Figure 1.

ActivityExtractor takes the following inputs: (1) an email collection along with a set of additional entities (e.g., calendar meetings and person names) represented by their word distribution vectors, (2) a user-specified parameter which specifies the initial number of clusters to be computed in step 1 of the algorithm, and (3) (optionally) a set of activity names input by the user. Given these inputs, ActivityExtractor produces as output a set of activity descriptions such as the one in Figure 1, along with a mapping from each activity to the input emails and entities associated with it. It does this using a three-step algorithm:

1. Cluster the input entities based on their associated word distributions.
2. Use social network analysis to further subdivide each cluster.
3. Create structured descriptions of the activity associated with each cluster, associating additional types of entities (e.g., meetings, person names) with each cluster.

The following three subsections describe each of these algorithm steps in turn.

Step 1: Cluster the Word Vectors

Given a set of entities represented by their word distribution vectors, the first step of ActivityExtractor is to cluster these entities. As a first pre-processing step, each email thread is preprocessed by summing vectors for all emails within the thread, replacing each of these vectors by this sum. The collection of word distribution vectors is then clustered using an EM-based clustering algorithm, following (Nigam et al., 2000). Here, clustering is viewed as a problem of estimating the components of a mixture of multinomials assumed to generate the collection of observed word vectors. Each mixture component is interpreted as the generator for one of the clusters, and is described by a multinomial model in which the vector feature values are assumed to be conditionally independent given the mixture component. An EM algorithm is used, beginning with an initial assignment of vectors to clusters, then iteratively solving for a locally maximum a posteriori (MAP) assignment of vectors to clusters. This EM procedure iterates the following two steps until the change across iterations becomes negligible:

- E step: for each example vector v in the set of input vectors E, use the current model Θ to determine the probability distribution over the set of possible clusters C. That is, calculate the distribution over cluster labels, $P(C \mid v; \Theta)$, for each vector v. Let E' denote the set of these label distributions.
- M step: retrain the model Θ using the label distributions E' computed in the E step. More formally, compute the new MAP estimate $\Theta' = \arg\max_{\theta} P(\theta \mid E', \alpha)$, where α denotes the parameter vector for Dirichlet priors on Θ.

One important issue in EM-based clustering algorithms is initializing the model Θ before the first iteration. While EM is guaranteed to converge to a locally maximum a posteriori estimate of Θ, the initial conditions determine which local maximum is reached. We consider two alternative approaches to initializing the model:

1. VDI: This approach assigns five initial vectors to each cluster, attempting to maximize the average inter-cluster distance and minimize the average within-cluster distance. Distance is measured by the cosine similarity between feature vectors. We use a heuristic method to search for these initial clusters.
2. GDI: This approach uses the optional user input activity names (e.g., "machine learning course 701") as queries to Google Desktop Search and forms the corresponding word vectors as described in previous section. The word vectors for the various activity names are used to train the initial model Θ.

Step 2: Perform Social Network Analysis

The clusters output from step 1 tend to group together entities with similar word distributions, but they are far from a perfect reflection of the user's activities. One key difficulty is that some emails do not belong to any ongoing activity – they are simply unsolicited emails, or very brief exchanges not related to ongoing activities, yet they must be assigned to some cluster during Step 1. Step 2 performs a social network analysis on the senders and receivers of emails associated with each cluster output from step 1, using this analysis to subdivide each cluster, thereby identifying communities of communicating email addresses, and splitting off isolated emails not part of a larger community.

In more detail, a social network graph is created from the emails in each cluster. For each cluster *c*, an undirected graph $G_c(N,E)$ is created, where each node in N represents an email address. An edge is added to E between two nodes if at least one email in the cluster contains both nodes in its header. The weight on this edge is the number of emails containing both nodes in the header. The workstation user's email address is excluded from the node set N, because their email address is obviously linked to every node in the graph.

The graph G_c is then examined to find its disconnected subgraphs, each representing a coherent subcommunity of email addresses (the nodes within this sub-graph). The output of Step 2 is a set of subclusters for each initial cluster c, where each subcluster $S_{c,i}$ contains all emails associated with the ith disconnected subgraph within G_c. The net effect of this subclustering is to separate out different subcommunities of email addresses.

ActivityExtractor by default outputs only the largest subcluster it finds for each initial cluster. Quite often this largest subcluster reflects a more accurate description of the initial cluster, whereas the smaller subclusters often reflect emails that are not truly associated with any of the ongoing activities.

Step 3: Create Final Activity Descriptions

Once it has produced the final clusters from step 2, ActivityExtractor constructs a description of each cluster (activity). In particular, it extracts (1) keywords based on a chi-squared test of the relevance of each word to the cluster, (2) the list of senders and recipients of emails within the cluster, rank ordered by the volume of mail they sent and received (3) the fraction of user email that falls into this cluster, as a measure of its importance, (4) the fraction of email within the cluster that is authored by the user, as a measure of the user's engagement in this activity, (5) meetings from the online calendar whose word vectors best match the cluster, and (6) person names from the input contacts list whose word vectors best match this cluster.

Experimental Evaluation

This section summarizes results from ActivityExtractor applied to three of the authors' workstations. Our experimental evaluation of ActivityExtractor was driven by three primary questions: First, what is the general quality of the clusters and the activity descriptions produced by ActivityExtractor? Second, what is the impact of using information from the entire workstation, as captured by Google Desktop Search? Third, what is the impact of incorporating the social network analysis step?

We evaluated ActivityExtractor using three workstations that are in routine use. Note we cannot evaluate ActivityExtractor on isolated email collections, because it employs information from the entire workstation via Google Desktop Search.

- *Workstation A*. This workstation is used routinely by a university professor. The email collection contains essentially all emails sent and received over a period of 45 days, minus spam emails. This collection contained 2822 emails, which were not organized into folders. This workstation has an online calendar containing 1231 unique meeting text strings. We extracted 2159 email addresses and 470 person names from the email headers. Finally, the user contributed 23 activity names reflecting activities he was involved in, such as "faculty hiring and interview."
- *Workstation B*. This workstation is used routinely by a university staff researcher. The email collection contains 420 emails, organized into 8 folders. There are 310 email recipients, 75 email senders, 76 person names, and 25 unique calendar meeting strings. The user contributed a short activity name for each folder.
- *Workstation C*. This workstation is used routinely by a university graduate student. The email collection contains 617 emails, which were organized into 11 folders. There are 66 email senders, 227 email recipients, and 56 person names. This workstation has no online calendar. The user contributed a short activity name for each folder.

Quality of Learned Activity Descriptions

We tested the quality of the final output of ActivityExtractor in two different ways: measuring the accuracy of the individual fields in the activity description, and measuring the accuracy of the email clusters associated with the activity. In this test, we used GDI initialization based on Google desktop search, input names of activities contributed by the workstation users as input, and used only the largest subclusters produced by the social network analysis in Step 2 of the algorithm.

First consider the accuracies of the individual fields in the activity descriptions. To evaluate these, we measured the accuracy of the five highest-confidence values assigned by ActivityExtractor for each of the following fields: PersonNames, Meetings, EmailSenders, EmailRecipients, and Keywords. In case the total number of values was truly less than five (e.g., if in fact there were fewer than 5 meetings truly associated with the activity), then we considered only this smaller number of values instead of 5.

	Wst A	Wst B	Wst C
Meetings	0.75	1.00	na
Person Names	0.89	0.89	0.67
Email Senders	0.98	0.92	0.90
Email Recipients	0.92	0.96	0.80
Keywords	0.67	0.80	0.76

Table 1. Accuracy of different fields in the 23 inferred activity descriptions for Workstation A, 8 descriptions for Workstation B, and 11 for Workstation C. Accuracies are for the five top-ranked values of each field, or among all values if there were fewer than five. Note workstation C has no online calendar, hence no assigned meetings.

The results of this analysis are summarized in Table 1. Depending on the field, accuracies range from .67 to .98, whereas default accuracies for random guessing are below 0.01 for most of these fields, and below 0.10 for all of them. Interestingly, email addresses (both senders and recipients) associated with activities are quite accurate, despite the fact that the assignment of emails to activity clusters is imperfect. The reason for this is that the important email addresses associated with an activity appear repeatedly in the email cluster, so that the choice of the dominant email addresses is robust to errors in assignments of individual emails. Note also that the assignment of person names and meetings are quite accurate, reflecting directly the ability of the word distribution vectors gathered via desktop search to capture a useful summary of each person name and meeting.

	Wst A	Wst B	Wst C
Accuracy	0.80	0.83	0.73
Before SNA	na	0.66	0.58

Table 2. Accuracy of email assignments to activities in each of the three workstation data sets. The top row gives accuracy using the full ActivityExtractor algorithm. The bottom row shows the accuracy obtained when the Social Network Analysis (SNA) step is omitted.

A second method for evaluating output activities is to examine the accuracy of email assignments to activities. Here we considered all emails (not just the most confident), comparing the email assignments by the program to ground truth labels. To obtain these labels the workstation user examined each cluster, determined which of their activities was best represented by the subcluster, then assigned this as true label for emails in the cluster. For workstations B and C these true labels were taken from the predefined folder assignments which were not observable to the program. Workstation A had no folder assignments, so the user hand labeled these emails as described above. The results are summarized in the top row of Table 2.

Impact of Social Network Analysis

To determine the impact of social network analysis, we also examined the accuracy of the email assignments to activities when the social network analysis step (step 2 of the algorithm) was omitted. The resulting accuracies are shown in the bottom row of Table 2. As is apparent there, social network analysis adds a great deal to the purity of the email clusters associated with output activities.

To gain some insight into the impact of social network analysis, consider one of the typical email clusters output from step 1 for workstation B. This particular cluster contained 28 emails, and the best match to the user's predefined activity names was to the "Help desk" activity, which the user intended to represent their ongoing interactions with the local computer facilities group. In fact, among these 28 emails social network analysis revealed that there were three disjoint communities of email senders and recipients, and therefore split this cluster into three groups. The largest of these indeed corresponds to the user's communication with the local computer facilities group. The other two subclusters involved email exchanges about setting up other remote computer accounts. This example illustrates the typical interplay between the topical clustering based on word distributions in Step 1 of ActivityExtractor, and the ability of social network analysis to introduce community-based distinctions between emails with similar topics and words.

Impact of Desktop Search

The use of Google desktop search as a subroutine is a key aspect of the system, and is used in two different ways. As discussed above, it is used to construct word distribution vectors for person names and meetings, so that they can be associated with activities. A second use of Google desktop search is to assign a word distribution

vector to the activity names input by the user, so these can be used to initialize the EM clustering process in Step 1 of the ActivityExtractor algorithm. To test the impact of this kind of initialization, we ran the step 1 clustering algorithm using both desktop initialization (GDI as described above) and the alternative VDI.

The results are displayed in Table 3, which gives the average of the cluster accuracies for workstations B and C, for both initialization methods. The final row in the table gives the baseline cluster accuracies obtained by randomly assigning emails to folders. The results show a substantial improvement in clustering accuracy when using GDI, the initialization method based on desktop search.

	Workstation B	Workstation C
VDI	0.48	0.41
GDI	0.66	0.58
random	0.13	0.09

Table 3. Impact of using desktop search to initialize activity clusters. Average of cluster accuracies when using VDI versus GDI initialization of clusters. GDI uses Google Desktop Search and input activity names, whereas VDI has access to neither. Accuracies are from Step 1 clustering, before social network analysis. "Random" shows the expected accuracy when emails are randomly assigned to activities.

Conclusions

We have presented an approach to automatically extracting structured descriptions of a user's ongoing activities from the raw contents of their workstation, using a combination of word-based clustering and social network analysis. A key element of our approach involves using desktop search to construct word distribution vectors for entities of different types (emails, people, meetings, etc.), providing a uniform representation that captures information from across the workstation.

Experimental results over three workstations indicate that (1) quite accurate structured descriptions of activities can be created, even when activity clusters are imperfect, (2) accuracy of activity clusters improves when clusters are initialized using word distributions gathered from user-provided activity names and Google Desktop Search, (3) accuracy of clusters improves further when social network analysis is used to split clusters into subcommunities of email senders and recipients, and (4) word vectors calculated using desktop search can be used to accurately associate calendar meetings and person names with the extracted activities.

These results indicate there is significant potential for building intelligent agents that monitor and reason about user activities.

Acknowledgements

This research was supported in part by Darpa under the CALO/PAL program.

References

Cohen, W. W., Carvalho, V. R. and Mitchell, T. *Learning to Classify Email into "Speech Acts"* EMNLP 2004.

Culotta, A., Bekkerman, R. and McCallum, A. *Extracting social networks and contact information from email and the Web*, CEAS 2004.

Dredze, M., Lau, T. and Kushmerick, N. *Automatically Classifying Emails into Activities*. IUI 2006.
EmailNet. A System for Automatically Mining Social Networks from Organizational Email Communication. NAACSOS 2003.

Geyer, W., Vogel, J., Cheng, L. and Muller, M. Supporting Activity-centric Collaboration through Peer-to-Peer Shared Objects. ACM GROUP 2003.

Huang, Y., Govindaraju, D., Mitchell, T. M., Carvalho, V. R. and Cohen, W. W. *Inferring Ongoing Activities of Workstation Users by Clustering Email.* First Conference on Email and Spam, 2004.

Kushmerick, N., Lau, T. *Automated Email Activity Management: An Unsupervised Learning Approach*, IUI, 2004.

McCallum, A., Corrada-Emmanuel A. and Wang, X. *Topic and Role Discovery in Social Networks, IJCAI* 2005.

Mock, K. *An experimental framework for email categorization and management*. In Proc. Int. Conf. Research and Development in Information Retrieval. 2001.

Nigam, K., McCallum, A., Thrun, S. and Mitchell, T. *Text Classification from Labeled and Unlabeled Documents using EM. Machine Learning*. 2000.

Rohall, S., Gruen, D., Moody, P., Wattenberg, M., Stern, M., Kerr, B., Stachel, B., Kushal, D., Armes, R. and Wilcox, E. *Remail: A reinvented email prototype.* In Proc. Conf. Human Factors in Computing Systems 2004.

Sahami, M., Dumais, S., Heckerman, D. and Horvitz, E. A Bayesian approach to filtering junk e-mail. In Proc. AAAI-98 Workshop on Learning for Text Categorization, 1998.

Segal, R. and Kephart, J. Incremental learning in SwiftFile. In Proc. Int. Conf. Machine Learning, 2000.

Surendran, A. C., Platt, J. C., Renshaw, E. *Automatic Discovery of Personal Topics To Organize Email.* CEAS 2005.

Tyler, J. R., Wilkinson, D. M. and Huberman, B. A. *Email as spectroscopy: Automated discovery of community structure within organizations*. Technical report, Hewlett-Packard Labs, 2003.