Activity-centric Search in Email

Einat Minkov
Language Technologies Inst.
Carnegie Mellon University
einat@cs.cmu.edu

Ramnath Balasubramanyan
Language Technologies Inst.
Carnegie Mellon University
rbalasub@cs.cmu.edu

William W. Cohen
Machine Learning Dep.
Carnegie Mellon University
wcohen@cs.cmu.edu

Abstract

We consider activity-centered tasks in email, including the novel tasks of predicting future involvement of persons from the enterprise in an ongoing activity represented by a folder, and a novel task where we identify email messages that are related to a to-do item. We also evaluate the task of email foldering, and the inverse task, of tracking messages relevant to a folder associated with an ongoing project. Empirical evaluation using authentic email data, applying a graph based link analysis method and a vector-space model, shows potential utility to facilitate activity management in email.

1 Introduction

Email plays a major role as a communication tool nowadays, both at the the enterprise and personal levels. Email also serves as an archive of historical information, which can be searched ad-hoc or mined in order to support the user's (or enterprise) ongoing needs. A variety of email-related tasks have been studied in the recent years that are aimed at facilitating email management and automatic analysis of email corpora. Example tasks include automatic finding of experts at the enterprise (e.g., [1, 16]), automaic recommendation of recipients for a given message (e.g., [4]), alias finding (e.g.,[11]), intelligent thread recovery (e.g., [13]) and more.

Naturally, email, as well as other entities at the work station including meetings, files and directories, correspond to different facets of underlying user *activities*, which evolve over time. Research has been conducted for automatically classifying emails into activities [6], for activity-centric collaboration [7] and for identifying threads of task activity based on the contents of emails and documents that people are working on [10, 5]. Mitchell et-al [15] suggest a framework for automatic extraction of user activities, based on analyzing the user's email, calendar, and practically the entire workstation contents accessible via Google Desktop Search. In the work of Belloti et-al [3], a user interface is suggested that is adapted to activity managment. Their goal is to support common activities such as organizing a meeting, planning a trip, conducting a performance review etc., as well as other user-defined activities. The suggested interface, named Activity-Centered Task Assistant (ACTA), is designed to create an efficient personal information management environment and provide context meta-data for machine learning and automation techniques. For example, it is desired that relevant emails, people, and email addresses be suggested to a user when viewing a meeting related to a particular activity on their calendar.

In this paper we describe a framework for activity-centered contextual search, that can naturally complement frameworks such as ACTA. In general, we represent email's content, social network and time information as well as activities, as a structured dataset (graph). Given this representation, we derive a measure of inter-entity similarity (relatedness). We apply finite graph walks to infer such a similarity metric, where related entities that are not directly similar to a query are reached via multi-step graph walk. In a previous work [13], this framework has been shown to be very effective for a couple of email processing tasks: namely, email threading and person name disambiguation. In this paper we extend this framework to represent also *activity* information.[1]

We phrase three different activity-centered tasks in email as search queries, and evaluate their performance using authentic data drawn from the Enron corpus. The first task considered is predicting which persons from the enterprise are to get involved in an ongoing

[1] In this paper, we sometimes use the word task with the same meaning intended.

activity in the future. While this task is somewhat related to the task of expert finding, predicting person-activity linkage is more challenging, as it involves the dynamics and variance of an activity. Second, we consider the task of foldering email messages to folders which denote an activity or a project, and vice versa. While foldering has been studied in the past, linking a folder to untagged messages is novel. Finally, we present preliminary results for another novel task, where given an email message that is related to a *to-do* task, we retrieve additional messages that are also related to the same implicit to-do item.

As an alternative to the graph walk paradigm, we compare against a vector-space model, where an entity is described by a summary of the messages that it is related to, as was applied in similar settings [15]. Given a similarity measure, we apply a mechanism of search, where related items are ranked by their similarity to a query.

The paper proceeds as follows. We first describe the schema used for representing email data and activities. We then provide a short overview of the similarity measures applied, including the graph walk based measure. The next sections present the tasks evaluated, the experiments conducted and their results. The paper concludes with a discussion of the results and future directions.

2 Email and Activity Representation

We model email as a heterogenous graph, where nodes denote typed entities, incuding *message*, *person*, *email-address*, *date* and *terms*. The suggested schema is described in Figure 1.[2] In the graph, nodes are interconnected with directed edges, which are typed. For example, a *message* can be linked to a *person* node with a relation of *sent-to*, *sent-from* etc. [13]. For every edge, there exists an inverse edge, in the opposite direction (the inverse edges are omitted in Figure 1). That is, the graph is highly connected and cyclic. In this representation, it is straight-forward to include related entities from the desktop, such as *meetings* [11]. In this paper, we add *activity* entities to the graph schema, as described in Figure 1. Namely, we draw a direct link between an activity to the email *messages* that it is related to.

In this paper, we refer to two notions of an activity. First, we consider user-created email *folders*. While not all folders pertain to a coherent activity (for example, a "sent-items" folder holds an eclectic collection of emails), folders are commonly used to tag a collection of messgaes related by an underlying activity. Folders are linked to messages directly in practice, which motivate the link structure in Figure 1.

[2]Entities of type email-address are modelled similarly to entities of type person. They were omitted from the figure for clarity.

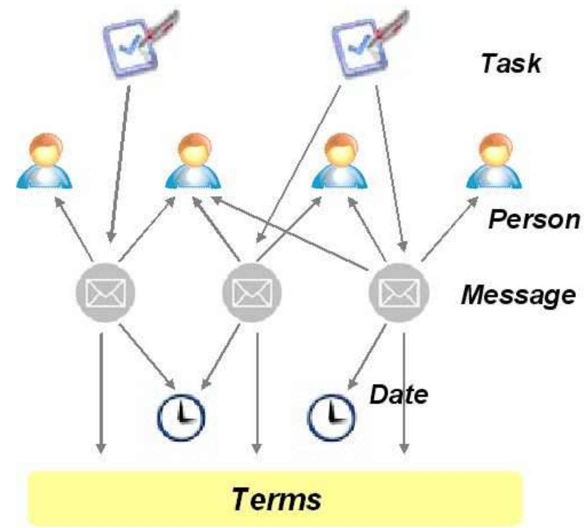

Figure 1: The suggested graph schema for representing email entities and activities.

The second form of an activity considered in this paper is *to-do* items. Examples include short-term activities such as setting up a meeting, writing a report etc. In this case, tasks are linked directly to a related *message* node. Links between messages that were not explicitly tagged and the relevant task can be inferred via similarity to other messages, for which the relevant task is known.

Given a graph that represents an email corpus as well as activities, and a measure of inter-entity similarity in the graph, various tasks can be phrased in terms of queries, within a retrieval framework.

3 Activity-centric Search

In this section, we first give a short description of the similarity measure based on graph walks [13]. We then describe an alternative similarity measure in the vector-space, similarly to the approach suggested by Mitchell et-al [15].

Following are relevant definitions and notations. A graph G consists of a set of nodes, and a set of labeled directed edges. We denote graph nodes by letters such as x, y, or z, and an edge from x to y with label ℓ as $x \xrightarrow{\ell} y$. Every node x has a type, denoted $\tau(x)$, where the graph scheme includes a fixed set of possible node types. For example, the graph scheme described in Figure 1 refers to six different node types, denoting *message*, *term*, *person*, *email-address*, *date* and *activity* entities.

Similarity between two nodes in the graph is defined by a weighted graph walk process, where an edge of type ℓ is assigned an edge weight determined by its type, θ_ℓ.[3] The probability of reaching node y from node x over a single time step, $Pr(x \longrightarrow y)$, is defined as the weight of their connecting edge, θ_l, normalized by the total outgoing weight from x. Given these transition probabilities, and starting from an initial distribution V_q of interest (a *query*), we perform a graph walk for a finite number of steps k. Further, at each step of the walk, a proportion γ of the probability mass at every node is emitted. Thus, this model applies exponential decay on path length. The final probability distribution of this walk over the graph nodes, R, is computed as follows: $R = \sum_{i=1}^{k} \gamma^i V_q \mathbf{M}^i$, where M is the transition matrix.[4] The answer to a query, V_q, is a list of nodes, ranked by the scores in the final distribution R. Given the resultant R, it is straight-forward to filter the rank list, to include only nodes of a specified node type τ_q.

Various tasks can be phrased in terms of the described similarity measure. For example, the task of *foldering* is phrased a query, where the initial distribution, V_q, consists of the entity denoting the *message* of interest, and the results are filtered to include the related nodes of type τ_q =*folder*.

A different approach previosuly suggested in the personal information management domain [15] is to model the various entities using word distributions. In this approach, the word distribution assigned to an email-address, for example, is the sum of the word distributions for each message sent by, and received by that email-address. Note that according to the schema described in Figure 1, all of the entity types have a direct link to related messages. We will therefore represent *activity* entities in the same fashion, that is, as a summation of the word distributions corresponding to the messages that they are known to be related to. The similarity of two arbitrary entities can be estimated by the dot product or cosine similarity between their word distribution vectors. In this paper, a TF-IDF weighting scheme is applied to all word distributions.

4 Corpora

Two corpora are used in the experiments.

The **Cspace** corpus contains email messages collected from a management course conducted at Carnegie Mellon University in 1997 [14]. In this course, MBA students, organized in teams of four to six members, ran simulated companies in different market scenarios.

[3]In this paper, we consider uniform edge weights.

[4]We tune k empirically, and set $\gamma = 0.5$, as in [12].

User	Snapshop date	# Messages
Kaminski.V	Feb 1st, 2001	1,193
Beck.S	Oct 1st, 2000	1,334
Kitchen.L	Sep 1st, 2001	1,065
Farmer.D	July 1st, 2000	741

Table 1: Corpora details

The **Enron** corpus is a collection of mail from the Enron corpus that has been made available to the research community [9]. This corpus can be easily segmented by user.[5] To eliminate spam and news postings we removed email files sent from email addresses with suffix ".com" that are not Enron's; widely distributed email files sent from addresses such as "enron.announcement" or "enron.chairman" at "enron.com"; and emails sent to "all.employees@enron.com" etc. Text from forwarded messages, or replied-to messages were also removed from the corpus. In deriving terms for the graph, terms were Porter-stemmed and stop words were removed.

In these paper, we consider the saved email of four different users. Each of the individual user corpora are truncated up to a particular point in time (individual to each user, to adapt to their individual periods of activity). The corpora include foldering information, as created by the users. For each folder, up to 80 most recent messages are maintained. That is, for each user we consider a snapshot of his or her corpus, where history is limited. The relevant corpora statistics for the four users are presented in Table 1.[6]

5 Person-Activity Prediction

There has been much interest recently in the task of expert finding using an enterprise email corpora [1, 16]. A somewhat similar task studied is finding relevant recipients for a given message or a meeting invitation [11, 4]. In this work, we consider a novel and more ambitious task, where we are to predict persons that are likely to get involved in an ongoing project activity, represented by a folder in an email corpus. While finding experts and recommending recipients relies on evidence observed in the past, the prediction of future ivolvement of persons from the enterprise in an ongoing project may depend on the dynamics of the project and other factors that are unknown within the email corpus alone, and possibly hard to predict in general. Nevertheless, it is reasonable that some of the people that will get involved in a project can be predicted

[5]Specifically, we used the "all_documents" folder, including both incoming and outgoing files.

[6]The Enron processed corpora as used in this paper are available from the first author upon request.

based on observed email correspondence. The task of person prediction, or recommendation, for an activity may be valuable to an organization, as it may promote early involvement of relevant individuals in a project. In the experiments conducted we are interested in evaluating the utility of person prediction for an ongoing activity, using authentic data drawn from the Enron corpus.

More formally, in the person-activity prediction task we are given a folder that is associated with a project activity as a query. The entities retrieved are of type *email-address*. We assume that the email messages in the folder provide textual and social network evidence, which allows the prediction of persons likely to get involved in the project in the future. As described earlier, we consider a snapshot of an email corpus at a particular point in time. Predictions are evaluated based on the email traffic that took place later in time.

5.1 Experimental Setup

We evaluate this task for a total of seven folders, drawn from the Enron corpus, that are associated with project activity. Table 2 details the folder names and the relevent user name, the number of messages maintained under the subject folder in the corpus,[7] the number of persons that are already associated with the messages in the folder, and the overall number of persons that known in the whole corpus of the relevant user.

In the experiments, a query is defined as the node representing the subject *folder*, and all of the entities of type *email-address* are ranked using the graph walk and the vector space models. Only addresses which are novel to the subject folder are considered valid answers. We apply a graph walk of $k = 4$ steps, using uniform edge weights. We apply the vector space model using both dot-product and cosine similarity, where we compare the TF-IDF weighted word distribution representing the folder with the distributions that describe each of the relevant email-address entities.

5.2 Results

It is reasonable that a user be presented by the recommending system with a relatively short ranked set of email addresses (or names). We therefore evaluate performance in terms of recall at the top ranks. Recall is measured by the number of email-addresses that indeed appeared in the folder later in time. The number of correct answers is also specified (referred to as "Answers" in the tables). Table 3 and Table 4 give the recall at the top 10 ranks and the top 20 ranks, respectively. Results are given for the TF-IDF weighted vector-space model using cosine similarity (Cosine) and dot-product similarity (DP), and for the graph walk method (GW).

[7] In general, in every folder we consider the latest 80 messages (where applicable). This allows both efficiency in maintaining email history, can possibly avoid bias towards larger folders, and also keeps the corpus up-to-date.

Folder	**User**	**# Known**	**# Targets**
London	Kamniski.V	111	611
Europe	Beck.S	144	635
Portland	Kitchen.L	106	552
East-power		156	
Mexico		49	
CES	Farmer.D	55	336
Wellhead		38	

Table 2: Person-activity prediction: Datasets' details

It is shown that the graph walk method gives the best performance in three out of the seven examples and cosine similarity is perferable in two of the cases, considering the top 10 results. Within the top 20 ranked results, the graph walk is preferable for four of the folders, where cosine similarity gives better results for two of the folders. Dot-product is inferior to cosine similarity in all cases. Interestingly, the graph walk predicts at least one person per folder within the top 20 predictions, whereas the cosine similarity is successful at prediting correct persons in only three out of the seven examples. For the *CES* folder, no correct prediction is found within the top 20 ranked results.

Overall, the results indicate that it is possible to predict future involvement of persons from the enterprise in an ongoing activity to some extent. Consider that our form of evaluation is strict, and it is possible that email-addresses (persons) predicted, who have not in fact appeared in the folder later in time, are also relevant and may be useful to a user. In terms of the methods applied, we consider the cosine similarity and the graph walk to give roughly comparable results. A major advantage of the graph walk, however, is that it can readily provide an explanation about the project-person relationship in the corpus. Namely, it is possible to present the user the primary paths leading from a folder to a person, including the traversed relation types. An explanation mechanism should be useful in motivating recommendations to a user who is reasonably familiar with the corpus.

6 Email Foldering

The foldering task [8] has been considered in the past. Previous works have shown good performance using algorithms such as TF-IDF [17] as well as Naive Bayes, MaxEnt and SVM [2]. In this work we evaluate the

Folder	# Answers	Cosine	DP	GW
London	19	0	0	**1**
Europe	33	0	0	**1**
Portland	25	**6**	5	2
East-power	14	1	0	**3**
Mexico	9	0	0	0
CES	13	0	0	0
Wellhead	9	**2**	0	1

Table 3: Person-activity prediction results: Recall at the top 10 ranks

Folder	# Answers	Cosine	DP	GW
London	19	0	0	**1**
Europe	33	0	0	**1**
Portland	25	**7**	**7**	3
East-power	14	1	0	**4**
Mexico	9	0	0	**2**
CES	13	0	0	0
Wellhead	9	**2**	0	1

Table 4: Person-activity prediction results: Recall at the top 20 ranks

performance of the graph-walk framework for this task, and compare it to TF-IDF. Unlike previous works, which classified email messages to the relevant folder, we approach this task as a *ranking* problem. In general, We assume a scenario where a user may be interested in associating a message to multiple folders relevant.[8]. Suggesting a ranked list of folders to the user supports multiple choice, where it is desired to have the most relevant folders placed at the top of the list.

In addition to the foldering task, we also consider in this paper the *inverse* problem, where the goal is to retrieve messages that are most relevant to a given folder. Consider a scenario where a user tags most messages with the relevant folder, but happens to skip some messages. Once this user is interested in retrieving a specific mistakenly untagged message, he will not be able to find it in the relevant folder. The task of message-folder tracking can be useful in such settings, as well as in the general case, where messages related to a particular activity are sought, while they may have been associated to other folders. To our knowledge, this task has not been studied previously.

6.1 Experimental Setup

In the experiments, we use the Enron corpora described in Table 1. In order to evaluate the foldering task, we consider the consequent "future" messages in each folder (past the corpus snapshot date). A query is defined as the message of interest, and entities of type *folder* are retrieved. In the graph representation,

[8]Multi-tagging is supported, for example, of the popular *gmail* application

Folder	User	# Test msgs
Conferences	Kaminski.V.	10
Resumes		10
London		10
Stanford		8
Congratulations	Beck.S	5
Recruiting		10
Europe		10
HR	Kitchen.L	10
East-power		10
Wellhead	Farmer.D	10

Table 5: Foldering: Datasets's details

the test messages are linked to all of the entities in the graph (persons, email-addresses, terms etc.) that it relates to. The test messages, however, are not linked to their corresponding folder. The folder nodes are reached by the multi-step graph walk. In the vector space, we compute the cosine as well as dot-product similarity of the TF-IDF vectors which represent the test message and each of the folders. Since in the Enron email corpora a message is attached to a single folder, the actual folder assigned to a message is considered as a single correct answer for each query.

The inverse foldering problem is addressed in a similar fashion. In this case the folder forms the query, and all of the test messages pertaining to the folder are the correct answers.

Table 5 details the folder names and the relevant user name included in the experiments, as well as the number of test messages per each folder.

6.2 Results

The results are presented in terms of mean average precision (MAP). In the case of foldering (Table 6), performance is very good. The TF-IDF cosine similarity measure gives the best results in six of the ten examples. The graph walk is superior in the remaining four instances. Also in this case, the cosine similarity measure performs better than dot-product. This suggests that document length normalization contributes to performance.

The results of the inverse folder-message trakcing task are presented in Table 7. The table also details the number of targets; i.e., the number of messages that are valid candidates (excluding the messages that are already attahced to the folder). For this task, the cosine similarity measure gives good performance. The performance of the dot-product similarity is far behind its cosine counterpart. The performance of the graph walk in this case is inconsistent, yielding the best performance for two of the examples, and failing for others. Random sampling of the results show some bias towards long messages. This fact, together with the

Folder	# Targets	Cosine	DP	GW
Conferences	33	**0.79**	0.41	0.56
Resumes	33	**0.85**	0.72	0.60
London	33	**0.95**	0.90	0.90
Stanford	33	**1.00**	0.94	0.81
Congratulations	89	0.47	0.27	**0.53**
Recruiting	89	**0.88**	0.73	0.83
Europe	89	**1.00**	0.55	0.53
HR	32	0.58	0.17	**0.90**
East-power	32	**0.95**	0.50	**0.95**
Wellhead	16	0.76	0.69	**0.79**

Table 6: Foldering: Results (MAP)

Folder	# Targets	Cosine	DP	GW
Conferences	779	**0.34**	0.15	0.04
Resumes	779	**0.34**	0.08	0.01
London	779	**0.35**	0.08	0.29
Stanford	832	**0.99**	0.12	0.13
Congratulations	1070	**0.37**	0.02	0.13
Recruiting	1070	0.50	0.09	**0.80**
Europe	1005	**0.27**	0.05	0.03
HR	1051	0.26	0.06	**0.52**
East-power	555	**0.97**	0.43	0.63
Wellhead	1005	**0.60**	0.09	0.10

Table 7: Folder-message tracking: Results (MAP)

gap between the cosine and dot-product suggest that length normalization is a key factor in these settings.

7 Linking "to-do" Items to Messages

In this task, we are interested in the retrieval of messages relevant to a given *to-do* item. We define a to-do item as a short-term task, similarly to the concept of activity in ACTA [3]; for instance, organizing a meeting. We assume that a user creates a to-do entity, once he or she receives (or sends) a related email message. In particular, the user interface and data structure of ACTA supports such an operation. Recall that the to-do entities are represented as an *activity*, according to the representation schema described in Figure 1. Given a particular to-do activity, we are interested in the retrieval of additional email messages pertaining to it. We will next present preliminary results for this task.

7.1 Experimental Setup

In the experiments, we use the CMU Cspace corpus. We annotated the email messages of one of the users, including a total of 156 messages, sent and received over a period of three months, with the implied to-do tasks. Overall, 127 action items have been identified, where an email message can be assigned multiple to-do items (or none). Example to-do items are given in Table 8.

meeting Sunday 2pm
financial information preparation
board meeting at 6pm sept 3
review product price information
enter data to file

Table 8: Sample to-do tasks

Given an *activity* node representing a task, our goal is to retrieve the email messages that are related to the same task.[9] In the graph, a task is linked directly to a single related email message. Links have been also added between consequent date nodes in the graph, in order to model temporal proximity. A graph walk of $k = 5$ was applied in the experiments. We compare against a cosine similarity model, where similarity is computed between a TF-IDF weighted word distribution representing the email message which is linked to the task node, and each of the other messages.

7.2 Results

We applied both the graph walk method and the cosine similarity paradigm to 20 to-do items, selected radomly. The majority of tasks sampled considered reviewing of legal issues, requests for information and meetings. The relevancy of top 5 items in each of the ranked lists returned by two approaches were evaluted. We found that using the graph walk with uniform edge weights, precision at the top 5 ranks was 0.46. Tuning the graph edge weight manually, where higher weights were assigned to message-person and message-date links, rather than message-terms, yielded improved precision of 0.57 at the top 5 ranks. The TF-IDF cosine similarity resulted in precision of 0.49 at the top 5 ranks.

We find these sampled results to be encouraging. In the future, we intend to use learning to adapt the graph walk performance to this problem (see [12]). Based on the results reported, we conjecture that time and recipient information (social network linkage) are very informative for this task. Textual information may not allow good discrimination between multiple instances of the same task. In general, the graph walk approach has the advantage of allowing to assign higher importance to recipient information and of modeling a timeline.

[9] Note that a to-do task is considered here as a specific instance of a possibly general task. For example, organizing a group meeting on a particular date, as illustrated in Table 8. Per this example, only messages which refer to this specific meeting are considered as correct answers.

8 Conclusion

In this paper we have presented several novel email-related activity-centric tasks: predicting which persons from the enterprise are likely to get involved in an ongoing activity, linking email messages to folders and vice versa, and linking to-do items to related email messages. We evaluated all tasks using authentic data from the Enron and the CMU Cspace email corpora, and compared the performance of a graph-walk based link analysis approach with a simpler TF-IDF vector space model.

The methods performed comparably for some of the tasks, suggesting that their combination may lead to further gains in performance. A TF-IDF cosine similarity measure performed preferably in retrieval of messages related to a folder from a large corpus, due to document length normalization. A similar mechanism can be embedded in the graph walk framework. The graph walk framework is beneficial in tasks where social network information plays a primary role, as is the case for the last task studied. In general, the graph walk framework also has a capability of providing explanations to a user about its predictions that use a natural terminology of entities and relations.

In the future, we would like to implement the tasks described here in a task-centric interface such as ACTA. A user is encouraged to link tasks directly to relevant persons, dates and so on in the ACTA activity-centric interface, which should lead to improved contextual retrieval.

References

[1] K. Balog and M. Rijke. Finding experts and their details in e-mail corpora. In *WWW*, 2006.

[2] R. Bekkerman, A. McCallum, and G. Huang. Automatic categorization of email into folders: Benchmark experiments on enron and sri corpora. In *Technical Report, Computer Science department, IR-418*, 2004.

[3] V. Bellotti and J. D. Thornton. Managing activities with tv-acta: Taskvista and activity-centered task assistant. In *Personal Information Management Workshop, SIGIR*, 2006.

[4] V. Carvalho and W. Cohen. Ranking users for intelligent message addressing. In *ECIR*, 2008.

[5] V. R. Carvalho and W. W. Cohen. On the collective classification of email "speech acts". In *SIGIR*, 2005.

[6] M. Dredze, T. Lau, and N. Kushmerick. Automatically classifying emails into activities. In *IUI*, 2006.

[7] W. Geyer, J. Vogel, L. Cheng, and M. Muller. Supporting activity-centric collaboration through peer-to-peer shared objects. In *ACM GROUP*, 2003.

[8] S. Henderson. Genre, task, topic and time: facets of personal digital document management. In *CHI*, 2005.

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

[10] N. Kushmerick and T. Lau. Automated email activity management: an unsupervised learning approach. In *IUI*, 2005.

[11] E. Minkov and W. W. Cohen. An email and meeting assistant using graph walks. In *CEAS*, 2006.

[12] E. Minkov and W. W. Cohen. Learning to rank typed graph walks: Local and global approaches. In *WebKDD and SNA-KDD joint workshop at AAAI*, 2007.

[13] E. Minkov, W. W. Cohen, and A. Y. Ng. Contextual search and name disambiguation in email using graphs. In *SIGIR*, 2006.

[14] E. Minkov, R. Wang, and W. Cohen. Extracting personal names from emails: Applying named entity recognition to informal text. In *HLT-EMNLP*, 2005.

[15] T. Mitchell, S. Wang, Y. Huang, and A. Cheyer. Extracting knowledge about users activities from raw workstation contents. In *AAAI*, 2006.

[16] D. Petkova and W. B. Croft. Hierarchical language models for expert finding in enterprise corpora. In *ICTAI*, 2006.

[17] R. Segal and J. Kephart. Incremental learning in swiftfile. In *ICML*, 2000.