AUTOMATIC DISCOVERY OF PERSONAL TOPICS TO ORGANIZE EMAIL

Arun C. Surendran
acsuren@microsoft.com

John C. Platt
jplatt@microsoft.com

Erin Renshaw
erinren@microsoft.com

Microsoft Research,
One Microsoft Way,
Redmond WA 98052

Abstract

We present in this paper a procedure to automatically discover a user's personal topics by clustering their emails. Unlike previous work, we automatically label topics using appropriate keywords. We show that, in order to get appropriate keywords, we must apply strong filters that use domain knowledge about e-mail and the workplace of the user. We demonstrate these keywords by creating an email/ document browser which makes use of these keywords as "standing queries" to create virtual folders that help organize, index and retrieve email efficiently. We present subjective user studies to show the usefulness of the strong filtering.

1 Introduction

Currently the onus of personalizing desktop services such as Windows Explorer or mail client is on the user. Users must create and maintain directory and folder structures and arrange their documents in these structures. Bellotti et. al., [Bellotti, 2003] argue that email is a "central place from which work is received, managed and delegated in organization" yet managing emails themselves has become a daunting task.

A quick look at the long list of documents dumped on to a user's home directory, or a user's mail inbox confirms that the average user is in need of some pro-active assistance in organization [Whittaker, 1996]. In fact, people place non-email related stuff in their Inbox by sending themselves mail [Cadiz, 2001]. Cadiz's study found that one of the most important tasks while handling email is "triage" – the process of handling and sorting emails. People tend to skip around and deal with "the most important messages first" [Cadiz, 2001]. Whitaker and Sidner identified three types of email organizers and suggested that one of the two immediate needs for an email client is an agent that automatically groups emails within categories [Whittaker 1996]. Finally, as demonstrated by Segal and Kephart [Segal, 1999] it may be helpful if the interface were to suggest a few labels that are likely to apply to the thread being considered, e.g. by similarity of contents or participants. Clearly the archiving strategy affects the retrieving strategy [Bälter 2001]. If the folder hierarchy is well-formed and well-used, retrieving messages should be easy.

Researchers agree that mail categorization is important. Earlier work in creating email categories revolved around creating user-defined flags [Cadiz, 2001], the use of simple attributes such as "Word file," "published paper," "shared with Jim," [Dourish, 1999]. Mock [Mock 2001] built a categorization framework that learned to classify emails into user created folders. These features require significant effort on the part of the user to provide tags, or to spend time creating folders that an agent can learn.

One issue with automatic categorization is that messages belong to more than one folder and assigning them to a particular folder may leave other folders incomplete [Cadiz, 2001]. There is also a problem of "out of sight, out of mind": emails that are moved out of the Inbox tend to be ignored. To combat these problems, [Dourish, 1999] and [Lewis, 1992] suggested using "standing queries": continually-updated virtual folders that correspond to user-provided queries.

Many of the arguments made for automatic organization of a user's inbox can also be made for a user's document store as well.

In this paper, we present a categorization system that is purely unsupervised and automatic: a pre-existing folder structure is not needed. The unsupervised system produces both relevant categories and relevant keywords.

The goal of this work is to automatically discover a person's topics of interest by looking at their email data. The end result of this process is to enable a wide variety of personalized services. One such service is an email/document categorization system that auto-arranges the entire user document store into topics that are very meaningful to the user. In this paper, we present such a system where the topics are created with no user input, and maintained with minimal interaction. We present a user interface that creates "standing queries" using these topics.

We use the email repository to discover a person's topics. A person's email store is a rich source of information that can be mined to enable various services. An explosion in recent years on mining this information has focused on discovering automatic actions, social network information, etc. It is clear that although information on a user's desktop is stored and accessed in separate "silos" (e.g. an email client such as Outlook tends to be independent of the desktop browser such as Windows Explorer) there is a strong correlation between the information stored in each of them. Recent works have begun to take advantage of such "cross-silo" effect [Huang, 2004] for information mining.

Much like [Huang, 2004], in this work we cluster a user's email. This paper has three key ideas that differentiate it from previous work.

(1) We label each cluster using a few relevant keywords. The "topic" hence is simply the most descriptive keywords chosen from a set of documents. As we will see later, this level of characterization enables a variety of services which are not possible with simple clusters of documents. In this work, clustering is only the means to achieve this end. Earlier work [Huang, 2004] was more focused on inferring "activities" like meeting dates, times, most frequent email sender, etc.

(2) Earlier work forced clusters to be evaluated as if they came from a supervised learning algorithm [Huang, 2004; Dourish, 1999; Mock 2001]. They asked a set of users to create folders and assign emails to them. The goal was to reconstruct these folders from the document via clustering. This evaluation is a tedious process requiring enormous amount of pre-labeled data from users covering all their topics of interest. Working with email stores has come to teach us how diverse and unpredictable people's email reading/storage habits can be [Mackay, 1988]. Supervised evaluation does not reward a clustering algorithm that constructs folders that were not seen before, nor does it measure whether clustering makes the distinction between topics, subtopics and related topics that a user would like to see. This is the inherent problem of personalization – it is *subjective*: it may not be measurable as an error rate. Hence our performance scores are based on people's subjective evaluation of the quality of the clustering output.

(3) We present a user interface – a personalized browser – that can auto-arrange all the email/document according to the discovered topics.

2 Automatic Discovery of Personal Topics

2.1 Definition of "personal topics"

A personal topic is defined as any cohesive concept that is relevant to the user – it could be an activity they participate in, an event they organized or attended, a person or a group of people they associate with, etc. A group of people can sometimes be defined by a concept that appears in emails, e.g., a project, a person, an activity, mailing group. Alternatively, a group of people can be defined by information that does not appear in emails, e.g., circle of friends that do not mention that they are friends. Most commonly, a personal topic would be signaled by the occurrence of words relating to a common activity. Sometimes these activities are too numerous and diverse for any one of them to represent the group, in which case the names of the people in the circle is the only thing representative of the circle itself.

2.2 Clustering

We use clustering as a means to derive the topics, not as an end in itself. Clustering emails to get representative "concepts" has been tried in [Boone, 1998]. There the clusters were pre-formed and the concept words are the ones that were most common among the documents in the group. In [Huang, 2004] clustering was used to get information about each group of documents – who sent the most emails, obtain names and dates, and task classification.

We tried different ways to cluster the data. To minimize the variations due of random initialization, we found that a multi-level clustering scheme works best. We represent documents using *tfidf* vectors of selected words. We used a cosine distance measure to measure document similarity. We initialize the clusters using Buckshot [Cutting, 1992] which is agglomerative clustering on a small sample of documents. We run K-means using these initializations on all the documents. Finally we run Probabilistic Latent Semantic indexing (PLSI) [Hoffman, 2001] using the K-means clusters as initial clusters. We found that this scheme works better than running PLSI with random initialization. A tempered version of PLSI was used to improve the quality of the estimation [Hoffman, 2001]. PLSI is well suited for topic extraction since it represents each document as mixture of topics, with each topic characterized by the distribution of words in them.

In each stage we weed out clusters that do not meet certain criteria. During Buckshot and K-means we weed out clusters that do not have more 10% of all documents. We do not recluster after weeding these out. After PLSI we only consider topics that have a prior probability that exceeds a particular threshold (0.1).

We also tried other clustering methods like mixture of multinomials [McCallum, 1998], hierarchical agglomerative clustering [Cutting 1992] but the sequence of methods we mentioned earlier gave the best results without spending an exorbitant amount of computation.

2.2.1 Preprocessing emails

Using domain dependent "bland/spicy" filter

It is important to do more than just filtering stop words to ensure the quality of clusters. We realized that many of the keywords that appear in the topics tend to be dominated by the domain of the user: e.g., the word "Microsoft" or "research" tend to occur prominently for people in our group. These words should be stop words in our situation. To create a domain-dependent stop word list, we take the index of all the web sites inside Microsoft and pick the 1000 most common words and filter these out.

Using only noun phrases that appear in subject of e-mails

Another key factor in quality of keywords was to whittle down the list of potential candidates to begin with. While [Huang, 2004] retained dates, days of the week, etc (all body words in general) we found that these words are more related to day-to-day activities rather than to do with the "topics" of the person. In fact, retaining these words hurt of the labeling of the topics. Our experiments suggest that the noun phrases that occur in the subject line form a good pool of representative or keywords/key phrases. We access email information from the MSN Desktop Search Index and first create a list of all the "Subject" lines in the emails. This list of subjects is then processed by a noun phrase tagger that is described in [Xun, 2000] to produce a list of noun phrases. We supplement these noun phrases with the set of all words that appear in any of these phrases. Then we query the index to find a count of all occurrences of these supplemented noun phrases in the entire body of the e-mail.

Using email metadata: author/recipient/cc information

In an additional experiment, we extract author/recipient/cc information and add them to the word list. When we added the author/recipient/cc information, we simply added new tokens to each e-mail document, corresponding to whom the e-mail was from and to. Thus, e-mail to or from Erin would gain an additional token of person_Erin_Renshaw. We included the person information for two reasons: (1) to help in further refining a "topic" based on people associated with it: e.g., the words "block email" are refined by the user group stop-spam@microsoft.com[1] that is associated with them and (2) to help in clearly separating topics that may share similar words but involve different people, e.g. "noise suppression" and "noise sup analysis" can be separated because the discussion involved different groups of people.

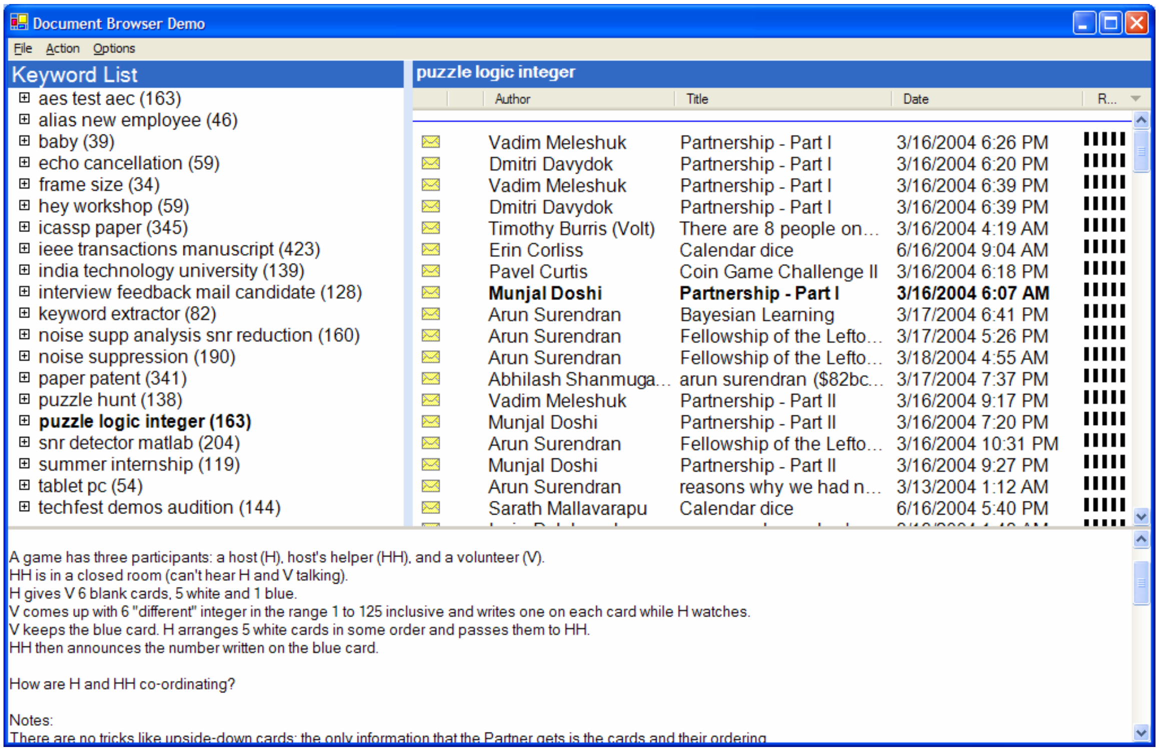
We will present results later to compare the quality of keywords with and without the above filters.

2.3 Multi-document keyphrase extraction

Multi-document keyphrase extraction distinguishes our method from previous email categorization/clustering methods. Some earlier methods learn pre-assigned categories which have to be picked by the user which is what we wanted to avoid in the first place. Further, pre-assigned categories do not have the ability to change with time: it is critical for the user to give new category lists/folder assignments to the system.

Our approach to multi-document keyphrase extraction is to pick a few characteristic keywords/keyphrases for each topic and use those as a characterization of the topic itself. One of the advantages of using PLSI for clustering is that it automatically characterizes each topic by the distribution of words in them [Hoffman, 2001]. We exploit this feature of PLSI to pick the most likely words for each topic as representative for that topic. We only pick words that are within half of the probability of the most likely word. Depending on the topic, this can range from two to five words. Most of the topics have one or two words to represent the topic. We also extract an additional set of keywords that lie between one-half and one-fifth of the most likely word. All words that are sub-phrases of other keyphrases are removed e.g. the word "puzzles" is removed if the phrase "puzzles and logic" appears in the list, but not if "puzzlesafari" appears. These words were used in evaluating the topics and later in extracting documents from the email/document browser. If words tagged as people appear in the top keywords list, these are not used to characterize a topic unless they are the only

[1] Not a real e-mail address



- ⊞ aes test aec (163)
- ⊞ alias new employee (46)
- ⊞ baby (39)
- ⊞ echo cancellation (59)
- ⊞ frame size (34)
- ⊞ hey workshop (59)
- ⊞ icassp paper (345)
- ⊞ ieee transactions manuscript (423)
- ⊞ india technology university (139)
- ⊞ interview feedback mail candidate (128)
- ⊞ keyword extractor (82)
- ⊞ noise supp analysis snr reduction (160)
- ⊞ noise suppression (190)
- ⊞ paper patent (341)
- ⊞ puzzle hunt (138)
- ⊞ **puzzle logic integer (163)**
- ⊞ snr detector matlab (204)
- ⊞ summer internship (119)
- ⊞ tablet pc (54)
- ⊞ techfest demos audition (144)

	Author	Title	Date	R...
✉	Vadim Meleshuk	Partnership - Part I	3/16/2004 6:26 PM	▮▮▮▮▮
✉	Dmitri Davydok	Partnership - Part I	3/16/2004 6:20 PM	▮▮▮▮▮
✉	Vadim Meleshuk	Partnership - Part I	3/16/2004 6:39 PM	▮▮▮▮▮
✉	Dmitri Davydok	Partnership - Part I	3/16/2004 6:39 PM	▮▮▮▮▮
✉	Timothy Burris (Volt)	There are 8 people on...	3/16/2004 4:19 AM	▮▮▮▮▮
✉	Erin Corliss	Calendar dice	6/16/2004 9:04 AM	▮▮▮▮▮
✉	Pavel Curtis	Coin Game Challenge II	3/16/2004 6:18 PM	▮▮▮▮▮
✉	**Munjal Doshi**	**Partnership - Part I**	**3/16/2004 6:07 AM**	▮▮▮▮▮
✉	Arun Surendran	Bayesian Learning	3/17/2004 6:41 PM	▮▮▮▮▮
✉	Arun Surendran	Fellowship of the Lefto...	3/17/2004 5:26 PM	▮▮▮▮▮
✉	Arun Surendran	Fellowship of the Lefto...	3/18/2004 4:55 AM	▮▮▮▮▮
✉	Abhilash Shanmuga...	arun surendran ($82bc...	3/17/2004 7:37 PM	▮▮▮▮▮
✉	Vadim Meleshuk	Partnership - Part II	3/16/2004 9:17 PM	▮▮▮▮▮
✉	Munjal Doshi	Partnership - Part II	3/16/2004 7:20 PM	▮▮▮▮▮
✉	Arun Surendran	Fellowship of the Lefto...	3/16/2004 10:31 PM	▮▮▮▮▮
✉	Munjal Doshi	Partnership - Part II	3/16/2004 9:27 PM	▮▮▮▮▮
✉	Arun Surendran	reasons why we had n...	3/13/2004 1:12 AM	▮▮▮▮▮
✉	Sarath Mallavarapu	Calendar dice	6/16/2004 5:40 PM	▮▮▮▮▮

A game has three participants: a host (H), host's helper (HH), and a volunteer (V).
HH is in a closed room (can't hear H and V talking).
H gives V 6 blank cards, 5 white and 1 blue.
V comes up with 6 "different" integer in the range 1 to 125 inclusive and writes one on each card while H watches.
V keeps the blue card. H arranges 5 white cards in some order and passes them to HH.
HH then announces the number written on the blue card.

How are H and HH co-ordinating?

Notes:
There are no tricks like upside-down cards; the only information that the Partner gets is the cards and their ordering

Figure 1. The Personalized Email/Document Browser based on automatically discovered personal topics.

words in the top list. The people words are used to assist in evaluating the topic.

3 Personalized Email/Document Browser

Using the personal topics extracted in the previous section, we built a personalized browser that auto-arranges all documents on a person's disk into these categories. Figure 1 shows a screen shot of the browser. On the left of the screen, the topics are listed and are appropriately labeled using keywords. The topics shown in Figure 1 are real topic discovered for a particular user in our evaluation. Each topic is represented by the top (visible) keywords and some hidden keywords. When a user clicks on a topic (in Figure 1 the topic "puzzle logic integer" is selected in bold), the browser uses all the keywords (visible and hidden) to retrieve all documents that contain these words. Then the retrieved documents are displayed on the right, and can be sorted based on their title, author, date or relevance to the topic. The relevance ranking is currently simply based on sum of the *tfidf* counts of all the keywords in the given document, but it can be easily replaced with other ranking mechanisms. In Figure 1, the documents returned are emails from a puzzles and logic discussion list, with a particular integer puzzle dominating the list. When the user clicks on each item in the email/document list, the item is displayed in the bottom window if it is an email, contact, or task item, and can be opened in a separate window if it is a pdf or other kinds of files that require a separate program to open them. In Figure 1, the mail introducing the puzzle "Partnership – I" is selected, and its content shown in the screen on the bottom.

A quick look at the list of topics discovered for this person shows a combination of projects (noise suppression, echo cancellation) and activities they participate in (interviews, puzzle hunt), discussion lists they are active in (puzzles and logic group) and important events in their lives (birth of a baby). There are some bad topics ("hey workshop") which seems to be a mixture of emails pertaining to workshops and friends.

In the next section, we evaluate the quality of the topics generated by our system.

4 Evaluation of Personal Topics

Since these topics are meant to be personal, we decided that a personal evaluation is best. Eight subjects were presented the keywords representing the topics that were extracted from their e-mail.

We tested three different ways of preprocessing emails: (1) Using all the words in the email body (2) Using word filters (including the spicy/bland filters and the subject-line noun phrase filters), and (3) using word filters plus adding author/recipient/cc information.

The algorithms were presented to the users in random order, without being labeled.

Subjects were given one set of keywords extracted from condition (1), and two repetitions of conditions (2) and (3) above, to give more power to the statistical test distinguishing between the two conditions. We asked the subjects to grade each topic using three labels – “Good”, “Mixed” and “Bad”. A set of keywords were assigned a relevancy score by the fraction:

$$\text{score} = (N_{\text{good}}+N_{\text{mixed}}/2)/(N_{\text{good}}+N_{\text{mixed}}+N_{\text{bad}}).$$

where N_X is the number of topics rated by a subject to have keyphrases of quality X. The mean scores for each condition and person from the experiments are shown in Table 1, below:

Table 1. Average keyphrase score for the 8 subjects.

Subject number	**Unfiltered score**	**Word filter score**	**Word filter + metadata score**
1	0.308	0.790	0.895
2	0.053	0.369	0.330
3	0.143	0.474	0.450
4	0.200	0.836	0.842
5	0.455	0.760	0.819
6	N/A	0.510	0.538
7	0.441	0.817	0.863
8	0.058	0.780	0.818

We applied two-way ANOVA to find whether there is a statistically significant difference in score based on subject and on pre-processing algorithm .We found that there is significance dependence of score on both pre-processing algorithm ($p=4x10^{-10}$) and on subject ($p=10^{-12}$). We then performed three t-tests on the three pairs of pre-processing algorithms, to determine if there were significant differences between the algorithms. The results of these t-tests are shown in Table 2,

Table 2. Confidence intervals for mean algorithm performance

Algorithm	**Mean**	**95% Confidence Interval**
Word filter + metadata	0.694	0.652---0.736
Word filter	0.667	0.625---0.709
Unfiltered	0.214	0.150---0.279

Table 2 shows that, using the standard method of bag-of-words on e-mail messages is disastrous: the performance is far worse than any of our filtering methods. Simply using the two filters dramatically improves clustering and keyphrase extraction. Adding the e-mail metadata does not improve performance by a statistically significant amount.

Note that three of our eight subjects had substantially worse performance than average: the dependency of performance on person is statistically significant. More research must be done to make the clustering algorithm robust for everyone.

For scenario (3) above, where author/recipient/cc information was used, we also wanted to evaluate if the correct people were associated with each topic. Thus, in addition to the quality of the keyphrases, we wanted to measured the quality of the social clusters that were generated by the algorithm. We tested this simultaneously while evaluating topics – we took the list of “people” who ended up in the top keywords and listed them separately. We asked the subjects to rate the list of people as “Very Relevant”, “Partly Relevant” and “Not Relevant” to the topic. Analogously to the keyphrase score, the social relevancy score is

$$\text{score} = (N_{\text{relevant}} + N_{\text{partly relevant}}/2) \\ / (N_{\text{relevant}} + N_{\text{partly relevant}} + N_{\text{irrelevant}}).$$

where N_X is number of topics rated by a subject to have assigned people of quality X. The score for the eight subjects are shown in Table 3.

Table 3. Average social score for 8 subjects

Subject Number	**Social score**
1	0.930
2	0.325
3	0.467
4	0.969
5	0.837
6	0.557
7	0.811
8	0.959

There is a strong linear relationship between the keyphrase score and the social score, as can be seen in Figure 2. Linear regression yields a slope of 1.0921 and an intercept of -0.0264. This regression is significant ($p < 0.0001$). This strong linear relationship is not surprising: both the keyphrases and the social information come from the same algorithm. When the algorithm produces a low-quality cluster, it produces low-quality keyphrases and low-quality associated people.

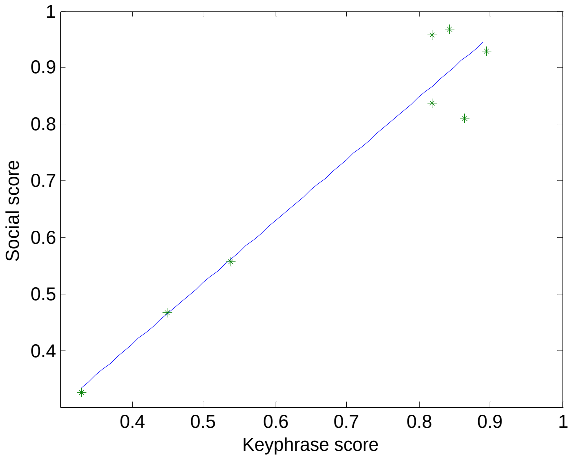


Figure 2. Linear regression on social score vs. keyphrase score. Subject scores shown as green dots, regression line in blue.

5 Summary

There is consensus among researchers that automatic categorization is necessary for emails, but there is little consensus on how to obtain categories that are important for the user without much user input. We present a method to automatically extract a person's topics of interest by clustering email. We label the clusters using appropriate keywords to represent these topics. We use these topics to create "standing queries" for organizing both email and other documents. A key component of creating good topics is the use of e-mail specific word filtering before clustering and keyphrase extraction. An informal user study shows that subjects find topics extracted with word filtering to be meaningful 69.4% of the time.

6 References

[Bälter, 2001] O. Bälter, and C. Sidner, "Bifrost Inbox Organizer: Giving Users Control over the Inbox". *NADA Technical Report* TRITA-NA-P0101. Royal Institute of Technology, Stockholm, Sweden.

[Bellotti, 2003] V. Bellotti, N. Ducheneaut, M. Howard and I. Smith, "Taking email to task: the design and evaluation of a task management centered email tool", *Proceedings of the Conference on Human Factors in Computing Systems (CHI-2003),* pp 345-352, 2003.

[Boone, 1998] G. Boone, "Concept Features in Re:Agent, an Intelligent email Agent", *Second Annual Conference on Autonomous Agents*, pp. 141-148, 1998.

[Cadiz, 2001] J. J. Cadiz, L. Dabbish, A. Gupta and G. Venolia, "Supporting Email Workflow", Microsoft Tech Report MSR-TR-2001-88, 2001.

[Cutting, 1992] D. R. Cutting, J. O. Pederson, D. Karger and J. W. Tukey, Scatter/Gather: "Cluster-based Approach to Browsing Large Document Collections", SIGIR'92, pp- 318-329, 1992.

[Dourish, 1999] Dourish, p., Edwards, K., LaMarca, A., and Salisbury, M., "Presto: An Experimental Architecture for Fluid Interactive Documents Spaces", *ACM Transactions on Computer-Human Interaction*, 6(2), 1999.

[Huang, 2004] Y. Huang, et. al., "Inferring Ongoing Activities of Workstation Users by Clustering Email", *Conference on Email and Anti-Spam*, 2004.

[Hoffman, 2001] T. Hoffman, "Probabilistic Latent Semantic Indexing", *Machine Learning.* 42, 177-196, 2001.

[Lewis, 1992] D. D. Lewis, *Representation and Learning in Information Retrieval*, Ph.D. Dissertation, Dept. of Computer and Information Science, Univ. of Massachusetts, Amherst, 1992.

[Mackay, 1988] W. Mackay, "More than Just A Communication System: Diversity in the Use of Electronic Mail", *Proceedings of the CSCW 1998 Conference on Computer Supported Co-operative Work*, pp. 344-353, 1988.

[McCallum, 1998] A. McCallum and K. Nigam, "A Comparison of Event Models for Naïve-Bayes Text Classification", *AAAI-98 Workshop on Text Categorization*, 1998.

[Mock, 2001] K. Mock, "An Experimental Framework for Email Categorization and Management", *SIGIR 2001*, pp 392-393, NY.

[Segal, 1999] R. Segal and J. Kephart, "Mailcat: an intelligent assistant for organizing email", In *Proc. 3rd Int. Conf. on Autonomous Agents* (Agents 99).

[Whittaker, 1996] Whittaker, S. and Sidner, C., "Email Overload: Exploring Personal Information Management of Email", *Proceedings of the ACM Conference on Human Factors in Computing Systems* (CHI 1996).

[Xun, 2000] E. Xun, C.Huang, and M. Zhou, "A Unified Statistical Model for the Identification of English BaseNP", *ACL-2000, The 38th Annual Meeting of the Association for Computational Linguistics,* Hong Kong, October 2000.