Generating Summary Keywords for Emails Using Topics

Mark Dredze[1], **Hanna M. Wallach**[2], **Danny Puller**[1], **Fernando Pereira**[1]

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

[2] Cavendish Laboratory
University of Cambridge
Cambridge CB3 0HE, UK
hmw26@cam.ac.uk

ABSTRACT

Email summary keywords, used to concisely represent the gist of an email, can help users manage and prioritize large numbers of messages. We develop an unsupervised learning framework for selecting summary keywords from emails using latent representations of the underlying topics in a user's mailbox. This approach selects words that describe each message in the context of existing topics rather than simply selecting keywords based on a single message in isolation. We present and compare four methods for selecting summary keywords based on two well-known models for inferring latent topics: latent semantic analysis and latent Dirichlet allocation. The quality of the summary keywords is assessed by generating summaries for emails from twelve users in the Enron corpus. The summary keywords are then used in place of entire messages in two proxy tasks: automated foldering and recipient prediction. We also evaluate the extent to which summary keywords enhance the information already available in a typical email user interface by repeating the same tasks using email subject lines.

ACM Classification Keywords

H5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.

General Terms

Design, Human Factors.

Author Keywords

email, foldering, keyword generation, recipient prediction, topic modeling

INTRODUCTION

Email inboxes typically display a limited amount of information about each email, usually the subject, sender and date. Users are then expected to perform email triage—the process of making decisions about how to handle these emails—based on this information. As the number of received email messages increases, tools to assist users with email triage become increasingly important. Additional concise and relevant information about each message can speed up the decision-making process.

Previous work on assisting users with email triage has focused on providing users with various types of additional information, including social information [21], short snippets of messages [27] and reply indicators [9]. While these methods make more information immediately available to the user, they do not provide a summary of message content. Several studies have proposed adding a short summary for each email [4, 7, 23, 29]. In practice, however, reading one or two sentences about each message is time consuming and displaying even a single sentence about each message requires considerable screen space, reducing the number of emails that can be displayed at once. Instead, this paper investigates an alternative technique: conveying the gist of each email in just a few words. The user can quickly glance at these email summary keywords when checking the subject and sender information for each message. This additional information should assist the user in making email triage decisions. Muresan *et al.* [20] first introduced this approach to summarization with a two-stage supervised learning system that selects nouns from individual emails using pre-defined linguistic rules. Unfortunately, the use of supervised learning techniques relies on user-specific keyword annotation of large numbers of emails for training purposes. Clearly, these data are not available for the average email user and it is unrealistic to expect each user to annotate several hundred emails in order to obtain such data.

In this paper, we develop and evaluate an unsupervised learning approach, which requires no annotated training data, for selecting email summary keywords. The key insight behind our approach is that a good summary keyword for an email message is not simply a word unique to that message, but a word that relates the message to other topically similar messages. We therefore use latent representations of the underlying topics in each user's inbox to find words that describe each message in the context of existing topics rather than selecting keywords based on a single message in isolation. We present and compare four methods for selecting email summary keywords, based on two well-known models for inferring latent topics from collections of documents: latent semantic analysis [8] and latent Dirichlet allocation [2].

We next discuss what makes a good summary keyword. We

IUI'08, January 13-16, 2008, Maspalomas, Gran Canaria, Spain.

then present two methods for selecting keywords: query-document similarity and word association. Each of these methods may be combined with one of two models: latent semantic analysis, and latent Dirichlet allocation. We evaluate the quality of the keywords generated by each method with two proxy tasks, in which the summaries are used in place of whole messages. Finally, we suggest future work.

CHOOSING GOOD SUMMARY KEYWORDS

We first consider what makes a summary keyword useful and how such keywords may be used. When a new email message is received, users typically look at the subject and sender. This serves two purposes: first, to prepare the user for the contents of the message—a kind of topic priming, and second, to influence decisions about how to handle the email message—for instance, whether to read the message now or later. For example, a message with the subject, "Dinner Next Week?" is less likely to be read immediately than a message with the subject "Urgent Client Meeting." As another example, a user might decide to read all messages about the quarterly budget, picking through the inbox listing for relevant messages. In both of these situations, the user is relying on a small amount of information—the subject and sender—to make email triage decisions. Providing the user with good summary keywords can facilitate these kinds of tasks.

The keywords that best assist users with email triage are quite different from the keywords used in other related tasks, such as *ad hoc* information retrieval or search. In retrieval tasks, good keywords are those that best distinguish each email from the other messages in a user's inbox. However, these keywords are too specific to be useful for email triage. Consider the following example:

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

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

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

This paper addresses the task of selecting keywords that satisfy all three requirements using latent concept models.

LATENT CONCEPT MODELS

Latent concept models [2, 13, 14, 16, 28] treat documents as having an underlying latent semantic structure, which may be inferred from word-document co-occurrences. The latent structure provides a low-dimensional representation that relates words to concepts and concepts to documents. In this paper, we use two widely-used latent concept models to generate email summary keywords: latent semantic analysis (LSA) and latent Dirichlet allocation (LDA). This section provides a brief overview of both methods.

LSA and LDA both represent text corpora such that the distribution of words in each document is expressed as a weighted combination of concepts or topics. In LSA, each concept is a real-valued vector and the the weights are real-valued too; in LDA, the concepts are distributions over words and the weights are mixing probabilities representing distributions over topics. The LSA concepts and weights can be easily computed using singular value decomposition, which often works well in practice. Estimating the LDA concepts and weights is more involved, but the model has the benefit of having a clear probabilistic interpretation that is a better fit to text and that supports many model extensions within the framework of hierarchical Bayesian models. As a result, LDA has been shown to improve over LSA in a wide range of applications [2,31]. Furthermore, the effective application of LDA to the task of selecting email summary keywords opens the possibility of refining the model to the specific attributes of email [16] and to this task in particular.

Latent Semantic Analysis

Latent semantic analysis, introduced by Deerwester *et al.* [8], models a text corpus as a word-document co-occurrence matrix X, in which each row corresponds to a word in the vocabulary and each column corresponds to a document. The element X_{wd} indicates the number of times word w occurred in document d. LSA decomposes this matrix into a set of K orthogonal factors, using singular value decomposition. This results in the matrices, U, S and V, whose product approximates the original word-document matrix. For a corpus with D documents and W words in the vocabulary, U will be a $W \times K$ matrix, where each row corresponds to a word in the corpus and each column corresponds to one of the K factors. Similarly, V will be a $D \times K$ matrix, where each row corresponds to a document in the corpus. S will be a $K \times K$ matrix consisting of the K orthogonal factors. While the original word-document matrix typically contains a positive function of word-document co-occurrences, U and V are real-valued and indicate the positive or negative association between each word and document and a particular latent factor. The K factors are thought of as latent concepts in the corpus. Words with similar meanings and usage patterns, such as "canine" and "dog," will be strongly associated with the same latent factors, while dissimilar words will not. The most appropriate number of latent factors K depends on the corpus. Throughout our experiments, we set K to 50. We leave automatic determination of K for future work.

Latent Dirichlet Allocation

Latent Dirichlet allocation provides another way of modeling latent concepts in corpora [2, 13, 26]. In contrast to

LSA, which represents words and documents as points in Euclidean space, LDA is a generative probabilistic model that treats each document d as a finite mixture over an underlying set of topics, where each topic t is characterized as a distribution over words. For example, a corpus of newspaper articles might contain latent topics that correspond to concepts such as "politics," "finance," "sports" and "entertainment." Each article has a different distribution over these topics: an article about government spending might give equal probability to the first two topics, while an article about the World Cup might give equal probabilities to the last two. LDA is a generative model: each word w in a document d is assumed to have been generated by first sampling a topic from a document-specific distribution over topics $\theta^{(d)}$, and then sampling a word from the distribution over words that characterizes that topic $\phi^{(t)}$. Furthermore, $\phi^{(t)}$ and $\theta^{(d)}$ are drawn from conjugate Dirichlet priors,

$$\theta^{(d)} \sim \text{Dir}(\alpha) \tag{1}$$

$$\phi^{(t)} \sim \text{Dir}(\beta), \tag{2}$$

and so $\theta^{(d)}$ and $\phi^{(t)}$ may be integrated out. The probability of w is therefore given by

$$P(w|d, \alpha, \beta) = \sum_{t=1}^{T} P(w|t, \beta)P(t|d, \alpha), \tag{3}$$

where T is the number of latent topics. Given a corpus of documents, statistical inference techniques may be used to invert the generative process and infer the latent topics and document-specific topic mixtures for that corpus. We used a Gibbs-EM algorithm to optimize the Dirichlet parameters α and β and infer the latent topics and document-specific topic mixtures. Gibbs-EM alternates between optimizing α and β and sampling a topic assignment for each word in the corpus from the distribution over topics for that word, conditioned on all other variables. The number of topics T, like the number of latent factors in LSA, is corpus-dependent. In all our experiments, we set the number of topics T to 100 and ran the Gibbs-EM algorithm for 500 iterations.

GENERATING SUMMARY KEYWORDS

In this section we present two ways to select email summary keywords, one based on query-document similarity, and the other based on word association. Each approach may be used in conjunction with either LSA or LDA. For each email, the pool of candidate keywords is restricted to only those words that actually occur in the email.

Query-Document Similarity

In information retrieval, it is often necessary to retrieve the set of documents that are most relevant to a query. Selecting summary keywords for an email message can be viewed as analogous to this task. Each candidate keyword is treated as a one word query and the similarity between that keyword and the email message is computed.

***LDA-doc*:** When using LDA for information retrieval, the document that is most relevant to a given query is the one that maximizes the conditional probability of the query given the document [3,31]. Similarly, the candidate keyword c that is most relevant to an email message d is the keyword that maximizes the conditional probability of c given d:

$$P(c|d, \alpha, \beta) = \sum_{t=1}^{T} P(c|t, \beta)P(t|d, \alpha), \tag{4}$$

where $P(c|t, \beta)$ and $P(t|d, \alpha)$ are posterior distributions obtained from all the emails in the user's inbox (including this one) up to this point in time and a set of corresponding topic assignments from a single Gibbs sample. The candidate keywords with the highest probability are those that are highly probable in the most probable topics for this document.

***LSA-doc*:** The similarity between a candidate keyword c and an email message d can be computed using LSA by taking the dot product of U_c and V_d, where U_c is the c^{th} row of the matrix U and V_d is the d^{th} row of the matrix V [8, 10]. As was the case with the LDA variant described above, candidate keywords that have similar concept membership to the email message will a receive a higher score.

Word Association

Word association scores pairs of words based on their association with each other [26]. A second approach to selecting summary keywords for an email message involves choosing as keywords those words that are most closely associated with the words that occur in the message.

***LDA-word*:** The degree to which a given word c is associated with another word w can be determined by treating w as a cue word and computing the conditional probability $P(c|w)$ that c is generated as a response to cue word w. This probability, under LDA, is given by:

$$P(c|w, \alpha, \beta) = \sum_{t=1}^{T} P(c|t, \beta)P(t|w, \alpha, \beta). \tag{5}$$

Candidate word c will have a high probability if it has a high probability in the topics that are most likely according to the posterior distribution over topics given w.

This similarity metric can be extended to measure the extent to which a word c is similar to an entire document, by computing the product of equation 5 over all the words in the document. Note that the words in the document are treated as a set: each word occurs only once in the product, regardless of the number of times it occurred in the document:

$$\begin{aligned} P(c|d, \alpha, \beta) &= \prod_{w \in d} P(c|w, \alpha, \beta) \\ &= \prod_{w \in d} \sum_{t=1}^{T} P(c|t, \beta)P(t|w, d, \alpha, \beta) \\ &\propto \prod_{w \in d} \sum_{t=1}^{T} P(c|t, \beta)P(w|t, \beta)P(t|d, \alpha). \end{aligned} \tag{6}$$

The proportionality in the last line is obtained using Bayes' rule: $P(A|B) = \frac{P(B|A)P(A)}{P(B)} \propto P(B|A)P(A)$.

***LSA-word*:** A similar technique may be used in conjunction with LSA. The product of the three probabilities in equation 6 is a measure of the similarity between two words c and w in document d under topic t, weighted by the probability of topic t in the document in question. In LSA, the similarity between words c and w may be computed by taking the dot product of the vectors that represent these words in the latent space, weighted by the strength of each factor for that document. The association between candidate keyword c and document d is computed by computing the sum of this quantity over all words w in the document:

$$\text{assoc}(c, d) = \sum_{w \in d} \sum_{k=1}^{K} U_{ck} U_{wk} V_{dk}. \quad (7)$$

The sum over words w in document d, which combines positive and negative association scores, is the LSA counterpart of the product of probabilities for LDA in equation 6.

EVALUATION

The keyword generation methods described in the previous section—*LDA-doc*, *LSA-doc*, *LDA-word*, *LSA-word*—were run on selected users from the Enron data set [15], a publicly available data set containing around 150 users and approximately 250,000 emails. Prior to generating keywords, we removed common stop words and email-specific words, such as "cc," "to" and "http."

The length of each summary was set to nine keywords. However, the optimal number of keywords is an open research question in interface design. If a message contained fewer than nine keywords with non-zero scores, a shorter summary was used. An example email and the corresponding summaries can be seen in figure 1.

Term frequency-inverse document frequency (TF-IDF) was used as a baseline against which *LDA-doc*, *LSA-doc*, *LDA-word* and *LSA-word* were compared. TF-IDF is a statistical technique for evaluating the importance of a word to a document. Words that occur rarely in a corpus, but often in a document will be ranked as being very important to that document. To generate summary keywords using TF-IDF, the nine highest scoring words for each email, according to TF-IDF, were selected. For completeness, the entire message body was used as an upper baseline.

Ideally, the quality of summary keywords would be assessed by the owner of the mailbox for which the summaries were generated. This is impossible for the Enron data set. As indicated by the summaries in figure 1, it is also difficult to determine how best to assess the quality of a summary. Furthermore, the task of evaluating all four LDA- and LSA-based generation methods, in addition to TF-IDF, would be prohibitively time consuming given the volume of mail involved. Keyword quality was therefore assessed using two proxy email prediction tasks. These tasks simulate the sorts of decisions a user would make using the keywords.

Two email prediction tasks were chosen as proxies for evaluating the keyword generation methods: automated foldering and recipient prediction. These tasks were chosen because they are well-defined, have previously been applied to the Enron data set and usually rely on the entire message body for making predictions. They are also typically tackled using different learning methods, allowing for a diverse evaluation. For each keyword generation method, both tasks were carried out on the Enron data set, with the message bodies replaced by the generated summaries. Predictions were therefore made using the summary keywords only.

User	*Messages*	*Total Messages*
beck-s	751	10168
farmer-d	3020	11395
kaminski-v	3172	25769
kitchen-l	2345	4691
lokay-m	1966	4299
sanders-r	863	5956
williams-w3	2542	3164

Table 1. The number of messages in the ten largest folders for each of the seven Enron users selected for the automated foldering task, as well as the total number of messages for each user.

Automated Foldering

Many email users archive and organize their email messages into folders. Automated foldering is the task of automatically predicting the appropriate folder for a given email message. This task was first introduced by Segal and Kephart [24] and has subsequently been explored in several settings [1, 15]. Since the messages in each folder are typically related by one or more common topics, automated foldering is a good task for evaluating summary keyword generation methods based on latent concept models.

We used an automated foldering task similar to that of Fink *et al.* [11] to evaluate the LDA- and LSA-based keyword generation methods. Each email was represented by a binary vector, where each position in the vector corresponds to a summary keyword. These vectors were used as input to a multi-class classifier with one class for each folder. The keyword generation methods were evaluated on the seven users used by Fink *et al.*, except that prediction was only run on the ten largest folders (excluding non-archival email folders, such as "inbox," "deleted items" and "discussion threads") for each user. This was done to compensate for the use of simpler features than Fink *et al.*. Even though only a subset of the messages were used for evaluation, the LDA and LSA models for each user were trained on all messages in the user's mailbox. Table 1 indicates the number of messages for each user.

The generation systems were evaluated using both online and batch learning algorithms. Online learning algorithms resemble a real-world setting: the algorithm receives a message, predicts a folder, and is then told which folder was in fact correct. All emails were processed in this fashion and the total classification accuracy was computed. For each user and generation method, ten trials were conducted on randomly shuffled data. MIRA [6, 19], a variant of the perceptron algorithm for large margin online classification, was

SUBJECT: ASE Hypertiles from Final Report Out
Sally -
Attached are the hypertiles from the final report out at yesterday's ASE Studio Workshop. The CD is finished and on its way to Houston. The files are organized by team:
Hammer - Sales and Marketing, Vision Stmt, Mission Stmt, Target Market, How to Approach, Pricing, SLA
Pliers - Producst and Services - Consulting Based
Saw - Infrastructure Transition Plan
Wrench - Producst and Services - Basic Outsourcing
I hope these help with your meeting tomorrow. Let me know if there is anything else I can do to help.
Lisa P

TF-IDF	*LSA-doc*	*LSA-word*	*LDA-doc*	*LDA-word*
producst	meeting	meeting	team	team
pliers	team	market	meeting	meeting
stmt	services	houston	services	services
hammer	houston	report	lisa	lisa
wrench	report	team	ase	ase
hypertiles	market	final	attached	attached
sla	tomorrow	transition	report	report
studio	final	yesterday	studio	studio
mission	plan	tomorrow	outsourcing	outsourcing

Figure 1. An email from the Enron corpus (*beck-s/ase/4*) and the summaries produced by the LDA- and LSA-based methods and a TF-IDF baseline. TF-IDF selects words that are overly specific to this message, including a misspelled word. The methods based on latent concept models select more general words that better capture the gist of the email. For example, *LDA-doc* and *LDA-word* both generate "team," "meeting," "ase" and "report."

used to perform the online classification. Batch learning algorithms process all messages prior to making any predictions. A maximum entropy classifier [18], along with ten-fold cross validation, was used for batch classification.

Figure 2 shows the classification accuracy on the automated foldering task using both batch and online learning algorithms, averaged over all users. The greater the accuracy, the better the foldering performance. The generation methods based on LDA and LSA all outperform TF-IDF in the batch setting, while all methods except for *LSA-word* outperform TF-IDF in the online setting. In general, batch performance exceeded online performance, which is unsurprising since online learning was run for a single iteration only.[1] Additionally, in the online setting the LDA- and LSA-based methods outperform using the entire message body, because the single-pass online learner can overfit when many words are involved. In contrast, using the entire message body does better in the batch setting. The differences between TF-IDF and the four LDA- and LSA-based generation methods when evaluated in the batch setting are statistically significant at $p = 0.05$ using McNemar's test measured on the aggregate predictions of each method across seven users. [2]

In both the online and batch settings, the methods based on query-document similarity outperform those based on word-association. Furthermore, using LDA consistently results in improvements over the results obtained using LSA. *LDA-doc* therefore achieves the best performance. Interestingly, in the batch setting, the accuracy obtained using *LDA-doc*comes close to the accuracy obtained using the entire message body. This indicates that the summary keywords generated by this method are a good approximation of the complete message content in the context of foldering.

Recipient Prediction

In large organizations, it is typical for multiple people to be involved in projects. This makes it easy to forget to include one or more recipients on a project-related email. Recipient prediction systems aim to prevent this by suggesting possible recipients to the user during message composition. Previous work has explored several learning methods for constructing recipient prediction systems [5, 22]. These systems are trained on previous messages sent by the user in order to determine associations between the words in each email and the message's recipients. When given a new message, a ranked list of potential recipients, based on the email's content, is presented to the user. Recipient prediction systems are evaluated by measuring the degree of agreement between the suggested and correct recipients.

Recipient prediction serves as a good proxy task for evaluating summary keywords generated using latent concept models. In our experiments, we use Carvalho and Cohen's system [5]. [3] This system employs both content similarity and statistical learning techniques. Training emails are labeled with their recipients and a K-nearest neighbor classifier is constructed. Given a new email, a list of potential recipients is constructed by voting among the closest messages.

[1] When online algorithms are run in a batch setting, it is common to run them for multiple iterations over the data.

[2] Measuring significance in the online setting is complicated because multiple predictions are issued for each message across the ten randomized runs.

[3] The authors thank Vitor Carvalho for providing access to and assistance with this system.

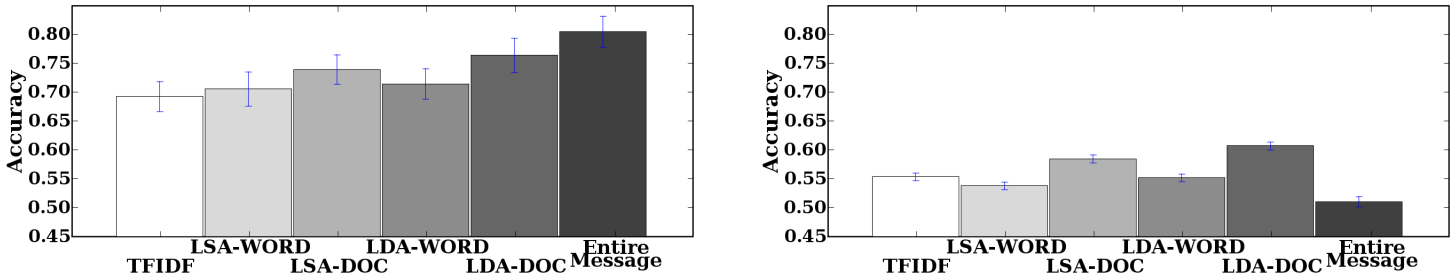

Figure 2. Automated foldering results using batch (left) and online (right) learning, averaged across all seven Enron users. Each graph shows the accuracy achieved on the foldering task using keywords generated by TF-IDF and the four and LSA-based methods, as well as the entire message. Error bars indicate standard deviation across each test.

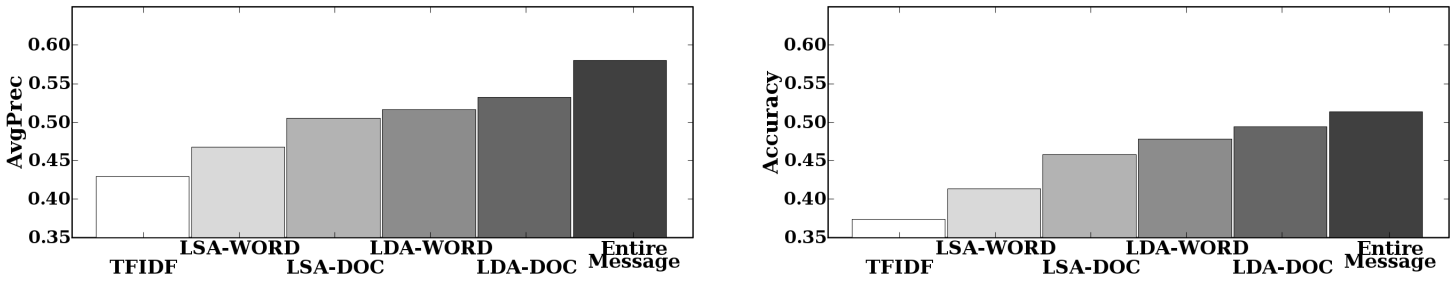

Figure 3. Results for the recipient prediction task showing the average precision (left) and accuracy (right) averaged across all seven Enron users, for TF-IDF and the four LDA- and LSA-based methods, as well as the entire message.

User	*Sent Messages*	*Total Messages*
geaccone-t	310	1352
germany-c	2692	9581
hyatt-k	400	1532
kaminski-v	1084	25769
kitchen-l	980	4691
white-s	432	2657
whitt-m	273	702

Table 2. The number of sent messages for each of the seven Enron users selected for the recipient prediction task, as well as the total number of messages for each user.

As with the automated foldering task, the text of every email message was replaced with summary keywords. The generation methods were evaluated on the seven Enron users used by Carvalho and Cohen. The list of users and the size of their sent mail folders is shown in table 2. For each user, sent messages were ordered chronologically. The last fifty messages were reserved for testing and the system was trained on the remaining messages. The system was run on the summaries generated by TF-IDF and each of the LDA- and LSA-based methods, as well as the full text of the messages.

Average precision and accuracy results, averaged across all seven users, are shown in figure 3. For both of these evaluation metrics, all four LDA- and LSA-based generation methods outperform the TF-IDF baseline. The improvements of *LSA-doc*, *LDA-doc* and *LDA-word* over the TF-IDF baseline and *LSA-word* are statistically significant at $p = 0.05$ using the Wilcoxon signed rank test. Furthermore, the generation methods based on LDA outperform those based on LSA and the methods that use query-document techniques beat those that are based on word association. Overall, the best generation method is *LDA-doc*, which performs statistically significantly better than *LSA-doc* and achieves accuracy comparable to using that obtained using the entire message.

DISCUSSION

The results obtained on the automated foldering and recipient prediction tasks demonstrate that summary keywords generated using latent concept models do indeed serve as a good approximation of message content. In almost all cases, the LDA- and LSA-based methods outperformed TF-IDF, and in many cases the LDA-based methods achieved performance close to that obtained using the entire message. We now explore additional properties of the generated keywords and discuss implications for users.

One of the requirements for summary keywords was that they be descriptive of the message and neither too general nor too specific. The extent to which keywords generated by the LDA- and LSA-based systems satisfy this requirement can be determined by analyzing frequency information for the keywords. The vocabulary size of the set of all summaries for a given method provides an indication of the specificity of the words in those summaries: a larger vocabulary indicates that each word is used fewer times, that is, the words are more specific. Table 3 lists the vocabulary size and average number of occurrences of each word for

Method	*Unique Words*	*Total Usage*
TF-IDF	10896	36.49
Entire Message	18213	29.16
LSA-doc	1793	205.33
LSA-word	3346	120.96
LDA-doc	2059	183.38
LDA-word	3301	123.88

Table 3. The average number of unique words in the summaries generated by each generation method, as well as the average number of times each keyword appears in the entire mailbox. Each summary contains a maximum of nine words. Results are averaged over the twelve Enron users used for the automated foldering and recipient prediction tasks.

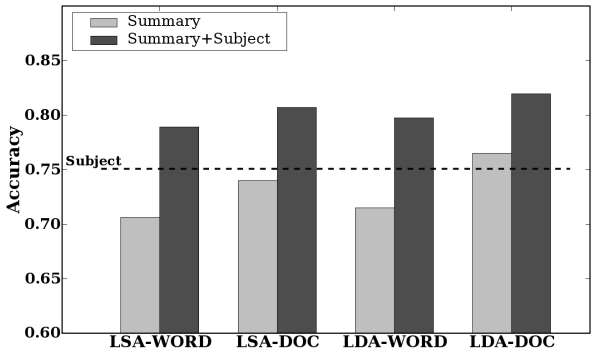

Figure 4. The accuracy, averaged across seven Enron users, for the automated foldering task using batch evaluation for the message subject, the summaries generated by the the LDA- and LSA-based methods, and the summaries combined with the subject.

each method. The summaries for all four of the LDA- and LSA-based methods have much smaller vocabulary sizes than those generated using TF-IDF. This, combined with the fact that the TF-IDF results were typically much worse than those of the other methods, indicates that TF-IDF is selecting keywords that are too specific, while the methods based on concept models are selecting more general keywords that better relate to common words in the users' topics.

In addition to evaluating summary keywords as an approximation to message content, it is also important to determine the extent to which summary keywords provide the user with additional information over the message's subject line. To investigate this, two additional experiments involving the automated foldering and recipient prediction tasks were carried out: one with the entire message replaced by the subject line and the other with the message contents replaced by both the summary keywords and email subject. Figures 4 and 5 show the results obtained in these experiments. The result obtained using just the email subjects is shown as a dashed line. On the automated foldering task, only *LDA-doc* achieved better performance than this. However, when combined with the subject, all the LDA- and LSA-based methods had significantly better results than those obtained using the subject alone. These results are statistically significant at $p = 0.05$ for *LSA-doc*, *LSA-word* and *LDA-word* and at $p = 0.01$ for *LDA-doc* using McNemar's test. On the recipient prediction task all four methods based on latent concept models give performance improvements over using only the subject. The improvements obtained using the LSA-based methods are significant at $p = 0.05$ using the Wilcoxon signed rank test, while the those obtained using the LDA-based methods are significant at $p = 0.001$.

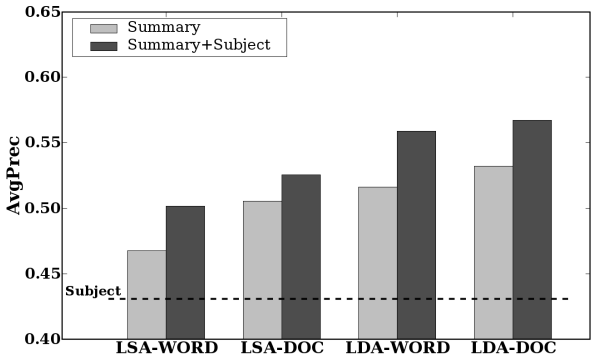

Figure 5. Average precision for the recipient prediction using only the message subject, the summaries generated by the LDA- and LSA-based methods, and the combination of the two.

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

FUTURE WORK

There are other latent concept models that could be used instead of LDA or LSA to further enhance performance. Wang and McCallum's topical n-gram model [30] integrates phrase discovery and topic modeling and would allow for the selection of summary phrases as well as summary keywords. In other work, McCallum *et al.* [17] condition the distributions over topics for email messages on senders and recipients. Incorporating this person-specific information could potentially improve keyword quality.

Any system in which email summary keywords are used must be able to to generate keywords quickly upon email arrival. Unfortunately learning latent concept models can be time-consuming. While this can be alleviated by running the techniques discussed in this paper during idle system time or on a webmail server, further work on the development and use of latent concept models that rapidly adapt to and process new documents would be beneficial.

Another area for future work is developing and evaluating methods for incorporating summary keywords into email client user interfaces. There is significant potential for creating innovative ways to display summary keywords, as well as carrying out practical evaluations of different presentation

methods and verifying the extent to which summary keywords assist real users with triage decisions.

The methods presented in this paper can also be used for tasks other than keyword summary generation. Recent work on blog tagging suggests that automatically suggesting appropriate tags for blog posts depends on topic identification [25]. Other recent work by Goodman and Carvalho [12] explores methods for generating implicit queries for search from emails. The keyword selection methods described in this paper could be applied to both of these tasks.

CONCLUSIONS

Email summary keyword selection using latent concept models can be carried out automatically without user intervention. The utility of the generated keywords, measured on two proxy tasks—automated foldering and recipient prediction, is significantly higher than that of keywords generated using TF-IDF. Specifically, the keywords generated using an approach based on LDA and query-document similarity concepts are consistently better than those generated using other methods addressed in this paper. Additionally, summary keywords generated using the LDA- and LSA-based methods presented in this paper were shown to provide additional information over email subject lines, and are therefore most effective when used in conjunction with email subjects. This provides significant impetus for the inclusion of such summary keywords in email client user interfaces.

ACKNOWLEDGMENTS

This work was supported in part by a NDSEG fellowship, in part by a University of Pennsylvania Provost's Undergraduate Research Mentoring Program Fellowship, and in part by the Defense Advanced Research Projects Agency (DARPA) under contract number NBCHD03001. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA or the Department of Interior-National Business Center (DOI-NBC).

REFERENCES

1. Ron Bekkerman, Andrew McCallum, and Gary Huang. Automatic categorization of email into folders: Benchmark experiments on Enron and SRI corpora. Technical Report IR-418, University of Massachusetts Amherst, 2004.
2. David Blei, Andrew Ng, and Michael Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.
3. W. Buntine, J. Löfström, J. Perkiö, S. Perttu, V. Poroshin, T. Silander, H. Tirri, A. Tuominen, and V. Tuulos. A scalable topic-based open source search engine. In *Proceedings of the IEEE/WIC/ACM Conference on Web Intelligence*, pages 228–234, 2004.
4. Giuseppe Carenini, Raymond Ng, and Xiaodong Zhou. Summarizing email conversations with clue words. In *Proceedings of the Sixteenth International World Wide Web Conference (WWW2007)*, 2007.
5. Vitor R. Carvalho and William Cohen. Recommending recipients in the Enron email corpus. Technical Report CMU-LTI-07-005, Carnegie Mellon University, 2007.
6. Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 2006.
7. Angelo Dalli, Yunqing Xia, and Yorick Wilks. Fasil email summarisation system. In *COLING*, 2004.
8. S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.
9. Mark Dredze, Tessa Lau, and Nicholas Kushmerick. Automatically classifying emails into activities. In *Proceedings of the International Conference on Intelligent User Interfaces*, 2006.
10. Susan T. Dumais. LSI meets TREC: A status report. In *Text REtrieval Conference*, pages 137–152, 1992.
11. Michael Fink, Shai Shalev-Shwartz, Yoram Singer, and Shimon Ullman. Online multiclass learning by interclass hypothesis sharing. In *International Conference on Machine Learning (ICML)*, 2006.
12. Joshua Goodman and Vitor R. Carvalho. Implicit queries for email. In *CEAS*, 2005.
13. T.L. Griffiths and M. Steyvers. A probabilistic approach to semantic representation. In *Proceedings of the 24th Annual Conference of the Cognitive Society*, 2002.
14. T. Hoffman. Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, 1999.
15. B. Klimt and Y. Yang. The Enron corpus: A new dataset for email classification research. In *ECML*, 2004.
16. Andrew McCallum, Andres Corrada-Emmanuel, and Xuerui Wang. Topic and role discovery in social networks. In *IJCAI*, 2005.
17. Andrew McCallum, Xuerui Wang, and Andres Corrada-Emmanuel. Topic and role discovery in social networks with experiments on Enron and academic email. In *Journal of Artificial Intelligence Research*, 2007.
18. Andrew Kachites McCallum. MALLET: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.
19. Ryan McDonald, Koby Crammer, Kuzman Ganchev, Surya Prakash Bachoti, and Mark Dredze. Penn StructLearn. http://www.seas.upenn.edu/ strctlrn/StructLearn/StructLearn.html, 2006.
20. Smaranda Muresan, Evelyne Tzoukermann, and Judith L. Klavans. Combining linguistic and machine learning techniques for email summarization. In *CONLL*, 2001.
21. Carman Neustaedter, A.J. Bernheim Brush, Marc A. Smith, and Danyel Fisher. The social network and relationship finder: Social sorting for email triage. In *Proceedings of the Conference on Email and Anti-Spam (CEAS)*, Mountain View, CA, 2005.
22. Chris Pal and Andrew McCallum. CC prediction with graphical models. In *Conference on Email and Anti-Spam (CEAS)*, 2006.
23. Owen Rambow, Lokesh Shrestha, John Chen, and Chirsty Lauridsen. Summarizing email threads. In *HLT/NAACL*, 2004.
24. R. Segal and J. Kephart. Mailcat: An intelligent assistant for organizing e-mail. In *Proceedings of the Third International Conference on Autonomous Agents*, 1999.
25. S Sood, S Owsley, K Hammond, and L Birnbaum. TagAssist: Automatic tag suggestion for blog posts. In *ICWSM*, 2007.
26. Mark Steyvers and Tom Griffiths. Probabilistic topic models. In D McNamara, S Dennis, and W Kintsch, editors, *Latent Semantic Analysis: A Road to Meaning*. Laurence Erlbaum, in press.
27. G. Venolia, L. Dabbish, J.J. Cadiz, and A. Gupta. Supporting email workflow. Technical Report MSR-TR-2001-88, Microsoft Research, 2001.
28. Hanna M. Wallach. Topic modeling: Beyond bag-of-words. In *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, Pennsylvania, 2006.
29. Stephen Wan and Kathy McKeown. Generating overview summaries of ongoing email thread discussions. In *COLING*, 2004.
30. Xuerui Wang and Andrew McCallum. A note on topical n-grams. Technical Report UM-CS-2005-071, University of Massachusetts Amherst, 2005.
31. Xing Wei and W. Bruce Croft. LDA-based document models for *Ad-hoc* retrieval. In *SIGIR*, 2006.