“Sorry, I Forgot the Attachment:” Email Attachment Prediction

Mark Dredze
Computer and Information
Sciences Department
University of Pennsylvania
Philadelphia, PA 19104
mdredze@cis.upenn.edu

John Blitzer
Computer and Information
Sciences Department
University of Pennsylvania
Philadelphia, PA 19104
blitzer@cis.upenn.edu

Fernando Pereira
Computer and Information
Sciences Department
University of Pennsylvania
Philadelphia, PA 19104
pereira@cis.upenn.edu

ABSTRACT

The missing attachment problem: a missing attachment generates a wave of emails from the recipients notifying the sender of the error. We present an attachment prediction system to reduce the volume of missing attachment mail. Our classifier could prompt an alert when an outgoing email is missing an attachment. Additionally, the system could activate an attachment recommendation system, whereby suggested documents are offered once the system determines the user is likely to include an attachment, effectively reminding the user to include the attachment. We present promising initial results and discuss implications of our work.

1. INTRODUCTION

File transfer is one of the most common applications of email. Users transfer files both as a matter of convenience and as a necessary part of collaborative projects. Most attachments arrive in the context of activities, such as editing papers or preparing proposals. Often times the document is part of a long email, where the sender spends a large amount of time preparing the message itself and then sometimes forgets the document. As a result, emails often arrive without their intended attachments, only to generate a flurry of emails attempting to correct the problem. This annoyance of email can create a serious disruption in workflow when a critical document is not received on time, especially when the sender is now out of contact.

This work offers a two part solution to the problem. Before a message is sent, the system can check whether or not an expected document has been attached to the message. Instead of blindly sending out the email, the system could prompt user if it determined that an attachment was missing. To this end, we present a high precision classifier that could trigger such a warning when an outgoing message is missing a needed attachment. An accurate system should be able to cut down the problem of missing attachments by making users more aware of potential mistakes.

Another possible solution is to suggest to the user during message composition to include an attachment. As we show, a high-precision classifier is difficult to achieve and has the potential to annoy the user, dealing with the problem at the last minute. Instead, an attachment recommendation system could offer the user possible attachments that may be relevant to the current email, thereby reminding them during message composition to include their attachment. The Rememberance Agent [7] is an example of such a system as it constantly suggests relevant documents based on the contents of an Emacs window, including message composition. We want to be able to offer relevant documents, but offering relevant documents for every email is annoying. We present a high recall classifier that avoids unnecessarily suggesting documents for every email. The classifier could trigger a sidebar to display relevant documents for the current message. The goal would be to not only passively remind the user to attach a document before they send the email, but to make finding needed documents easier.

Horvitz [4] examined user goal prediction for providing aid to the user. Our work is similar in that we predict a specific goal, document attachment. Additionally, frustrated users have developed a Mac OS X Mail plugin that uses keywords to alert users to possible missing attachments [1]. While this simple approach can only handle a limited number of cases, it demonstrates the severity of the problem.

We first present our system implementation, followed by a description of our evaluation data, methodology and results. Our discussion focuses on the potential for transfer learning in future work, a necessity given our initial results.

2. SYSTEM

Our system is based on the same system used for Dredze et al. [3], which has been extended into a general framework for email classification. The system uses logistic regression for classification and includes numerous features for email classification, including message and subject content. Features include attributes such as unigrams, bigrams, and length of message. All attributes of the email were represented as binary features.

The classifier is implemented as a MaxEnt classifier using the MALLET Java package [6]. We obtained high recall and high precision classifiers by varying the default feature weight on a trained model. In addition to the above features, we also used several types of features tailored specifically to attachment prediction, such as proximity to important words like “attach”. We also removed several features from the system such as recipient, sender, and other user specific information. While these may be good features for classification for a single user, they do not apply to other users and fail to transfer. Future work will explore how we can incorporate these types of features and still permit transfer.

CEAS 2006 - Third Conference on Email and Anti-Spam, July 27-28, 2006, Mountain View, California USA

	Precision	*Recall*	*F*
Balanced			
Entire Corpus	0.85	0.56	**0.67**
Transfer	0.78	0.42	**0.54**
High Recall			
Entire Corpus	0.55	**0.84**	0.66
Transfer	0.40	**0.72**	0.51
High Precision			
Entire Corpus	**0.90**	0.49	0.63
Transfer	**0.87**	0.27	0.41

Table 1: Results for high recall, high precision, and balanced classifiers evaluated on the entire corpus and simulated user transfer. Results are averaged over 10 runs.

3. CORPUS

We evaluated both our high precision and our high recall classifiers on the Enron email corpus. While the original Enron emails contained attachment information, this information had been excluded from the prepared corpus [5]. We annotated the corpus with information from the original data file (a database dump) to produce an annotated subset of the corpus. Each email received an "X-Header" indicating whether or not it contained an attachment. Emails that had one or more attachments received an additional header indicating the name and type of the attachment. Emails received this header only if they actually were sent with an attachment by the original user. A positive label corresponded to a positive value in the "X-Header", meaning that the email had an attachment; negative labels were applied to emails that did not contain an attachment. Our negatively-labeled instances may include emails that *should* include an attachment, which was forgotten. It would be difficult to correct these labels manually, and we believe that they are a relatively small fraction of the overall corpus. For training and testing purposes, we selected email from 24 users, which had mailboxes larger than 30 messages, yielding a total of 7656 messages of which 1017 had attachments, about 13%. Each mailbox varied in size and was a combination of multiple email folders, including folders such as "inbox" and "discussion".

We had to make several modifications to the corpus to prepare it for our task. First, many attachment emails actually contained a forwarded message as an attachment, the default behavior of some clients. We also excluded some user mailboxes that appeared to be composed mostly of machine generated email. Furthermore, there were several hundred attachment emails that were formatted reports relating to financial data. While this email is easier to classify, the large similarity and volume of these messages unduly positively influenced performance. We removed these emails from the corpus. Finally, if a email was sent to two users, it appeared twice in the corpus. We removed these duplicate emails.

We discovered a significant problem with the corpus data for our task. Since the emails in the corpus had actually contained attachments, residual artifacts were introduced by various email clients. For example, some attachment messages included artifacts such as "⟨⟨File: E&Y Memo.doc⟩⟩" or "–E&Y Memo.doc". Since a real email draft would never include these attributes, we needed to remove these artifacts. We hand checked 200 messages that had attachments, randomly sampled from each user's inbox to obtain a wide variety of email types. As each email was checked, we developed a list of artifacts to remove. After automatically processing the corpus to remove these artifacts, we rechecked the 200 emails to verify that they were clean of any of these features.

While each of these modifications predictably lowered our system's performance, we feel that the resulting corpus more accurately reflects real world email and that our performance numbers more closely model a real world system.

4. EVALUATION

Using our cleaned dataset of 7656 messages we conducted two evaluations. First, we randomly split the corpus along an 80/20 train-test split, training and testing classifiers on the split corpus. These results are presented as *Entire Corpus.* Next, we split the corpus by user, sorting the users randomly using an 80/20 train-test split. If a user was placed in the train group, all email from that user was used for training; the same was done for the test group. This ensured that if an email belonged to the train or test set, all other emails in that user's mailbox were placed into the same split. Therefore, a mailbox used in the testing of the classifier did not affect training. This represented a pseudo-transfer task between users and is presented as *Transfer.* While we would ideally like to test our classifier on a user's sent mail, this information was not available in the corpus. We plan to evaluate further transfer scenarios in our future work, such as transferring a classifier trained on Enron data to non-users. We evaluated our high precision and high recall classifiers, as well as a balanced classifier, on both of these datasets 10 times. Our results show the average of the 10 runs.

5. DISCUSSION AND FUTURE WORK

Table 1 presents results for the three classifiers. For testing on the entire corpus, the balanced classifier produces promising results which favor precision. If an email discusses a document then it will have an attachment (precision), while many emails that contain attachments lack discussion about the document (recall). We noticed several interesting examples where an email did not directly refer to an attachment and we were unable to conclude if there was an attachment from the language of the email except by checking if one was included. This indicates that the problem may be complex even for a human reader. An interesting result is that when attempting to transfer a classifier between users, performance suffers substantially, 13 points in the balanced classifier case. Different users are likely to deal with different types of attachments so each user's lexicon varies substantially. A small precision drop compared to a substantial drop in recall lends evidence to this hypothesis.

This observation leads us to explore user transfer in future work. Any real world system would need to operate on a new user, so transfer between users is vital. The Enron corpus displays terms unknown in other environments, such as "rentrolls". While the presence of "rentroll" may be a good feature in the Enron domain, it is a poor indicator in most email. Other users may discuss documents such as "budgets" or "proposals" giving these features importance in attachment prediction. Effective transfer would allow for a mapping between these words in the Enron domain to another user's lexicon. Structural Correspondence Learning uses "pivot features" common in the source and target domains to match important non-overlapping features [2]. SCL has been used in domain adaptation and could be a useful tool in transfer learning between users. This may be especially relevant to attachment prediction where there are few important cross-user features.

6. ACKNOWLEDGMENTS

Dredze is supported by a NDSEG fellowship. This material is based upon work partially supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. NBCHD030010. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the DARPA or the Department of Interior-National Business Center (DOI-NBC).

7. REFERENCES

[1] Avoid sending mail with unattached attachments. http://www.macosxhints.com, 2006.

[2] J. Blitzer, R. McDonald, and F. Pereira. Domain adaptation with structural correspondence learning. In *The Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Sydney, Australia, 2006.

[3] M. Dredze, J. Blitzer, and F. Pereira. Reply expectation prediction for email management. In *The Second Conference on Email and Anti-Spam (CEAS)*, Stanford, CA, 2005.

[4] E. Horvitz. Principles of mixed-initiative user interfaces. In *CHI*, pages 159–166, 1999.

[5] B. Klimt and Y. Yang. The enron corpus: A new dataset for email classification research.

[6] A. McCallum. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

[7] B. Rhodes and T. Starner. The remembrance agent: A continuously running information retrieval system. In *The Proceedings of the First International Conference on Practical Applications of Intelligent Agents and Multi-Agent Technology (PAAM'96)*, pages 486–495, London, 1996.