AttachMate: Highlight Extraction from Email Attachments

Joshua Hailpern
HP Labs
1501 Pagemill Road
Palo Alto, CA 94304
joshua.hailpern@hp.com

Sitaram Asur
HP Labs
1501 Pagemill Road
Palo Alto, CA 94304
sitaram.asur@hp.com

Kyle Rector
Comp. Sci. and Eng. Dept
University of Washington
Seattle, WA 98195
rectorky@cs.washington.edu

ABSTRACT
While email is a major conduit for information sharing in enterprise, there has been little work on exploring the files sent along with these messages – attachments. These accompanying documents can be large (multiple megabytes), lengthy (multiple pages), and not optimized for the smaller screen sizes, limited reading time, and expensive bandwidth of mobile users. Thus, attachments can increase data storage costs (for both end users and email servers), drain users' time when irrelevant, cause important information to be missed when ignored, and pose a serious access issue for mobile users. To address these problems we created AttachMate, a novel email attachment summarization system. AttachMate can summarize the content of email attachments and automatically insert the summary into the text of the email. AttachMate also stores all files in the cloud, reducing file storage costs and bandwidth consumption. In this paper, the primary contribution is the AttachMate client/server architecture. To ground, support and validate the AttachMate system we present two upfront studies (813 participants) to understand the state and limitations of attachments, a novel algorithm to extract representative concept sentences (tested through two validation studies), and a user study of AttachMate within an enterprise.

Author Keywords
email; attachment; summaries; attachment summaries

ACM Classification Keywords
H.5.3. Group and Organization Interfaces: Asynchronous interaction

INTRODUCTION
Email functions are the life-blood of enterprise communication, facilitating work-related activities from task management to personal archiving and asynchronous communication [30]. Enterprise workers are consistently overloaded with the volume of email, which has led to a large segment of HCI literature that focuses on improving the UI of and interacting with email messages [29]. However, one major aspect of email - attachments - has gone largely unexamined.

UIST'14, October 05 - 08 2014, Honolulu, HI, USA

http://dx.doi.org/10.1145/2642918.2647420

Attachments are files (documents, slides, etc) that are sent along with an email to supplement the email's content, or as the main/only informational content. These files can be large (multiple megabytes), lengthy (multiple pages), and not optimized for smaller screen sizes, limited reading time, or expensive bandwidth of mobile users. Thus, attachments can increase data storage costs (for both end users and email servers), drain users' time when irrelevant, cause important information to be missed if ignored, and pose a serious access issue for mobile users. In addition, users do not always describe the content of their attachments in detail (if at all).

In this paper, we address these attachment problems with a system called **AttachMate**. Our approach informs users of attachment content by extracting representative highlights from text documents (PDFs, plain text, and MS Word DOC and DOCX files) attached to an email, and injecting those summaries as plain text into the email body itself. This allows both desktop *and* mobile users to easily read the summary from each attachment, judge the value of the content, and make an informed decision about whether or not to read the attachment itself (or if the summary provides enough information). Further, while the summaries are sent with the emails, the attachment itself is securely stored on the AttachMate server, for user access upon request. By storing all attachments on a server (regardless of how many recipients a message is sent or forwarded to), the attachment is not replicated in every inbox and sent mail box (which are stored locally and on the server). Our approach is more accessible, more informative, and reduces data storage and network bandwidth consumption which is especially helpful for mobile phone users.

The **primary contribution** of this paper is the AttachMate client/server architecture. To support, ground and validate AttachMate itself, we have three secondary contributions; **First,** a novel algorithm to extract representative concept sentences from an attachment (which was tested and refined through two validation studies); **Second,** two upfront investigations (with 813 participants) to understand the current state and limitations of attachments (in an enterprise); and **Third,** a user study of AttachMate use within a corporate environment.

RELATED WORK
This paper spans literature from two distinct communities: document summarization and HCI research on email. Subsequently, this section will discuss each area in turn.

Email Interaction Research

HCI research on email systems has examined a wide variety of topics from email as information management[30] to predicting salient emails in universities [6] to studying semantics of communication [11]. Among this large volume of work, few researchers have examined the role of attachments within this email ecosystem, especially as pertaining to helping users identify the content of attachments.

The majority of attachment research focuses on email construction [28], or creating intelligent email systems that alert users if they fail to include an attachment [1, 7]. Hanqal's work on visualizing email databases included an attachment visualization akin to a museum wall[15]. In addition, some work outside of the HCI and IR community has examined the utility of attachment patterns to detect malicious email content [2]. However, none of this literature addresses the specific problems of attachment overload, or informs users about the content of email attachments.

Document Summarization

Automatic summarization is the process by which a description of a document or collections of documents is generated by a computer algorithm. Given the large and expansive literature, this paper's contribution can best be positioned by viewing automatic summarization along two axes: extraction vs. abstraction, and corpus vs single document.

Extraction summarization pulls sentence fragments or full sentences from the original text to create a summary[23], while abstraction summarization uses advanced natural langue processing (NLP) techniques to generate new sentences that describe the document(s)[10]. While abstraction summarization can be more descriptive, it is far more complex to implement and difficult to get 'right' due to the heavy reliance on complex and still imperfect NLP techniques.

Corpus summarization uses a large collection of documents to build a model of the topics being discussed (e.g. Topic Modeling [3, 5], SumBasic[13], KLSum[24]) or opinions rendered (e.g. Opinion Mining [14]). Single document summarization [20, 17] utilizes only one document to create a summary. Corpus summarization approaches rely upon a large body of documents from which patterns about the "whole" can be derived, and are generally easier and more powerful because they have more data from which to draw summaries. Through this power of large numbers, summarization of Twitter topics can be done [5]. Within single document summarization, most algorithms are designed to summarize long (e.g. book)[32], well structured (e.g. chapters or sections) text[8, 27, 31], thus maximizing the amount of text and structural cues from which to derive summaries.

This paper's context specifically requires an algorithm that can handle single documents since attachments are not guaranteed to have a related document corpus. In addition, the algorithm should assume the documents are unstructured (not all attachments exhibit structural elements that can be extracted), and of unknown length (attachments can be very short or very long). There are few unstructured, variable length, single document summarization algorithms (all of which are extraction based). The most notable in this domain are TextRank[23] and LexRank[9]. Both of these algorithms use a simple graph-based approach, treating each sentence as a node. The summary sentence of the document is calculated by finding the centroid of the graph based on a distance vector[1]. This general approach can be thought of as a baseline for novel algorithm development.

In commercial systems, Microsoft has created an Auto Summarize function within the Microsoft Word software[2]. While they do not officially document their algorithm, according to [12], Microsoft appears to use averaged term frequency within a sentence. We use the output of Microsoft's Auto Summarize functionality as another baseline for comparison.

SCOPE AND MOTIVATION

Given the limited research on attachment interaction and single unstructured document summarization, this work mainly contributes to both areas. Thus, the primary contribution of this work is **AttachMate**, a system that automatically summarizes email attachments, includes the summarization in the email, and stores the original attachments on a password protected server. This allows the user receiving the email to easily view the summary (even on mobile devices), as it is stored in plain text, and reduces the network bandwidth and data storage (local and server side) consumption by storing all attachments in the cloud and only sending links to end users. This system is validated through a real-world study.

In order to ground this work in additional motivation beyond existing research literature, we conduct two studies of email attachment usage in an enterprise setting: a survey of 777 enterprise employees' perception of email attachments, and an Outlook add-in that logged attachment usage for the daily workflow of 19 users from HP.

In order to facilitate attachment summarization this paper presents a secondary contribution, a novel approach to single unstructured document summarization through extraction. Multiple potential solutions are tested and compared to existing baselines via two Mechanical Turk studies.

While any of the above paper components could be expanded into a full paper, our goal was a single comprehensive AttachMate *system* publication. These components and studies were done to justify, create and test the systems impact.

ATTACHMENT PERCEPTION IN ENTERPRISE

Before building or designing any software system, we wanted to better understand the perception and challenges of attachments in an enterprise setting. A survey was distributed to the employees of HP located in Palo Alto, CA to which 777 employees responded (30.47% Male, 66.92% Female, 2.61% not reported) with a mean age of 39.37 years (σ=11.45 ▂▅▃▂▁). A broad and representative set of users responded,

[1]As a distance vector, TextRank uses words in common while LexRank uses TF-IDF and cosine similarity. Presumably LexRank would use term frequency for single documents.

[2]Part of Microsoft Office for Windows.

% of Total Attachments...	Mean (SD)	Median [IQ Range]	Histogram
...That You Read	0.73 (0.27)	0.80 [0.50,0.90]	
...That Have Value	0.53 (0.27)	0.50 [0.30,0.75]	

Table 1: Survey Quantitative Responses
percentage read refers to desktop access only

from multiple roles within the company[3] and educational attainment[4]. The survey asked basic demographic questions and some quantitative (Table 1) and short answer questions.

Overall, respondents reported reading a fairly high percentage of attachments (73%), though they found that only around half actually have any value. This suggests that many documents are read unnecessarily. Of those attached documents that are *not* opened, participants reported (short answer) that the primary reasons for not opening was: a perceived lack of relevance (25.51%), attachment length (18.69%), would take too much time to read (16.96%), lack of interest in the content (15.94%), assume most of the content is in the email body (12.32%), and 10.58% other reason. Though there are many reasons for people to not open an attachment, these study results explicitly showed that ∼60% of the reasons were related to the length (making it too cumbersome or long to read) or unknown content.

Mobile attachment access was more pessimistically characterized, with 50.06% of respondents saying they never open attachments on mobile devices at all. In short answer responses, the primary issues with attachments on mobile devices were small screen/document length (22.17%), time to read/access (19.81%), difficulty in reading (18.73%), support for files on the mobile OS (12.06%), correct rendering/formatting of the attachment (11.95%), download time over cellular network (8.93%), and 20.78% other responses.

These responses indicate that almost half of all attachments have no value and yet, on desktop email systems, 27% of all attachments being opened are not important. In the respondents' own words, users are wasting time reading irrelevant, long documents. This problem is even worse on mobile devices, where half of all users don't bother to open attachments at all. For these mobile users, the top issues are not with content, rather the quality of the reading/access experience on the small devices. This suggests that solutions are needed to help both desktop *and* mobile users discriminate which attachments to read. For those on mobile devices, it is necessary to provide a minimal amount of access to attachment content without the burdens of attachment access itself.

ATTACHMENT USAGE IN ENTERPRISE

To further ground this work and understand how attachments are *actually* used in enterprise, we create an Outlook add-in (OAI) that logs email attachment usage on a secure server. We recruited 55 participants from the users who had completed the original survey, who filled out a demographic survey, and were sent instructions on how to get the OAI. Because we did not track which users actually participated (due to privacy), we can only report data from all 55 participants that signed up, though our final data set consisted of only 19 unique users (41% Female, with a mean age of 48.04, σ=11.11). As in the last study, participants spanned a large portion of an enterprise[5] with a wide variety of educational attainment[6].

Participants were asked to participate for 1-2 weeks (5-10 business days). However, the actual duration was up to each individual. Users participated on average for 8.68 days (σ=3.83) with a median participation length of 9 days [6,12].

During the experimental period, 7090 emails were logged[7]. Users received on average 42.27 emails (σ=24.88) per day, of those messages 25.32% (σ=11.46) contained attachments, and 32.84% (σ=15.24) of attachments are documents[8]. Roughly 8.79% of all emails (σ=6.73) contained documents as attachments. Of those emails with any type of attachment, on average 2.19 files were included (σ=1.20). When documents are sent, on average 1.34 (σ=0.35) documents are included. Documents averaged 0.58 MB (σ=0.48) in size, with the average largest document at 6.69 MB. Of those emails with documents as attachments, a large 65.5% of document attachments are opened.

From this real-world data, we can see that documents are fairly large, taking up a considerable amount of physical storage on local computers (and on exchange servers). In regard to attachment access, user data suggests that users are opening a large percentage of documents needlessly. A solution, like AttachMate, could therefore be used to improve access to, and reduce data footprint from, attachments.

ATTACHMATE: SYSTEM

Given the number of challenges highlighted in the above two studies we created **AttachMate**, a system that automatically summarizes email attachments. Summaries are injected as plain text in the body of the email, and the original attachments are stored on a server with a unique password for protection. AttachMate allows the user receiving an email to easily view the summary (even on mobile devices), reduces the network bandwidth consumed by the email (as the attachment only is transferred once, to the server), and greatly relieves the need for data storage (as attachments are only stored in one location, rather than every server and local mailbox). AttachMate was designed to be unobtrusive to users' daily routines. To this end, we built a server-client architecture with Microsoft Outlook[9] add-in (OAI) as our client. The OAI sends the emails and attachments to the **A**ttach**M**ate **S**erver (**AMS**) which performs the summarization, attachment storage, and email sending (with summaries included).

When the OAI is installed, if a user clicks the send button on an email with an attachment (PDF, MS Word or TXT file),

[3]273 Management, 41 Research, 272 Engineering, 25 Finance, 8 Legal, 27 Administrative Assistant, 31 Business, 127 Other
[4]71 PhD/JD/MFA, 237 MS/MA, 379 BS/BA, 85 AD, 32 HS
[5]21 Management, 5 Business, 2 Marketing, 4 Legal, 6 IT, 12 Engineering, 2 Finance, 3 administrative assistant
[6]3 PhD/MFA/JD, 19 MS/MA, 24 BS/BA, 5 AD, 5 HS
[7] Statistics reported are normalized on a per-user basis, and partial days (only containing data from the AM or PM) were thrown out so as to accurately report rates (per day) of usage.
[8]Examples of non-document attachments are images
[9]Outlook is the required mail client at HP

```
================ Highlights from "cloud media.docx" ================

This is particularly true in the case of Social Media where the data
sources are varied and no single tool can provide all the data required
for meaningful analytics.

Additional platforms and metric providers can also be integrated with
ExtractPro as and when they are identified.

These enhancements will make it a complete end-to-end social media
data extraction application.

To Download "cloud media.docx" please click (password: VCD70) -
http://11.222.333.44:80/attachmate/attachments/getAttachment/z2UghmvhaZlrreihKoIr

====================================================================
```

Figure 1: Sample Summary Using AttachMate

AttachMate intercepts the email and asks the user (via pop-up) if they would like to use the AttachMate service[10]. If the user chooses to use AttachMate, the entire content, metadata, signature (as specified in the Outlook preferences), and the attachment(s) are sent to the AMS.

The AMS stores each attachment in the cloud. Every file stored is checked against any other files (via hash) to determine if the file is redundant (this reduces storage costs on the server end)[11]. The AMS then creates a unique URL for each file, and a randomly generated password to protect access. Distinct URLs and passwords are generated so each sent attachment appears to be unique. The content of each file is then extracted, and run through the summarization algorithm (described later).

The content of the summary is then inserted into the body of the email, before the sender's signature (if present)[12]. In addition to the summary sentences, AttachMate includes the URL to download the file, and the password for access. Visual delineation of the AttachMate summaries is included so that the reader can easily find the break points between the summary and body copy (Figure 1). The revised email content, and additional meta data are then sent via SMTP server to the recipient (spoofing the sender's address). AttachMate does not associate attachments with user names (senders or receivers), nor does it store the messages themselves.

Subsequently, the email recipient's mailbox (server or desktop) never receives the attachments themselves as the attachment(s) are only transferred once (sender to AMS). Downloads are therefore only executed by explicit user request[13]. Overall, this reduces storage costs, network costs, and access speeds as files are only ever stored once, and not replicated across multiple exchange server mailboxes or local Outlook caches. In addition, when emails are traditionally replied to or forwarded, attachments are also be included, thus increasing the proliferation of the file. With AttachMate, simply forwarding the links and passwords allows attachments to be shared (with summaries), but the files remain on the server (further reducing bandwidth and storage) across multiple mail service provides or on systems that do not have deduplications built in. Lastly, attachment storage on the server is further optimized by keeping only one copy of each unique file (though distinct URLs and passwords are generated so each sent attachment appears to be unique). Thus, redundant attachments are only stored once.

[10]This ensures full transparency, requiring an explicit opt-in for any email using AttachMate.

[11]It should be noted that we do not claim that deduplication is a novel idea of this work, simply a technique to further optimize attachment server storage.

[12]If no signature is detected, the summary is injected either before (if the email is long) or after (for short messages) the body of the email.

[13]Given that only 65.5% of attached documents are opened, this would greatly reduce download bandwidth consumption

	Politics	Entertainment	Science	Sports	Technology
Grade Level[15]	11.81(2.76)	9.92(2.99)	14.27(4.06)	10.24(1.67)	12.49(3.10)
Reading Level[16]	49.23(14.47)	56.25(13.29)	36.58(17.74)	59.55(9.27)	46.25(11.91)
Fog Index[17]	8.79(1.37)	7.30(2.00)	9.92(2.61)	8.56(0.67)	9.22(2.35)
# Sentences	70.50(30.32)	57.25(6.75)	49.25(31.08)	52.75(14.77)	51.50(33.45)
Word Count	1481.00(523.88)	1020.25(208.36)	1068.50(415.81)	1117.50(306.86)	1053.75(497.70)

Table 2: Documents Source Statistics

Mean and Std Values Reported

SUMMARIZATION ALGORITHM: DESIGN AND TESTING

A major component in the creation of AttachMate is to define an *extraction* summarization algorithm to help summarize the content of a *single unstructured document*. As discussed above in the related work, summarizing single documents without any guaranteed length or structure is currently an open problem. We therefore set out to create a new extraction summarization algorithm to be used within the AttachMate system that could out-perform the existing techniques.

Rather than simply finding one sentence to summarize an entire attachment (e.g. centroid of a network), our goal was to design an algorithm that could find three "highlights" from a given document. By showing three[14], rather than one sentence, our aim is to provide users a broader view of an attachment's content and aid users in determining if the document should be read in full. This is especially necessary for mobile users where the time and effort required to read an attachment is much higher. In addition, not every document has one "perfect" sentence that covers all of its content. To this end, we designed, built and tested a series of algorithms to summarize an individual document. This section details the two rounds of development and testing. Testing was conducted using Amazon Mechanical Turk.

Documents

In order to test the comparative performance of any summarization algorithm, we collected a series of documents from 5 subject areas (Political Articles, Entertainment Reviews, Scientific Documents, Sports News, and Technology News). Within each subject area, 4 documents were selected from various sources and authors. Thus, each approach could be tested on a broad set of documents of various complexity, length and reading levels (Table 2).

[14]We chose a "reasonable" number of sentences, as is standard practice in the Data Mining and IR community.

[15]Flesch-Kincaid grade level indicates that a student at that current U.S. school grade should be able to understand said document (e.g. 8.0 is eighth grade). 7.0 to 8.0 is "optimal."

[16]The Flesch reading ease rates text on a 100 point scale, with higher scores being easier to understand. 60-70 is "optimal."

[17]Years of education to understand a document in a single reading (e.g. 12.0 is a high school senior). 8.0 is considered "optimal."

	Algorithm	Mean (SD)	Median [IQR]	Histogram
Overall	Word	3.74 (1.69)	4 [2,5]	
	WDBC	**4.87 (1.27)**	**5 [4,6]**	
	Clus Cntr	4.11 (1.55)	4 [3,5]	
Politics	Word	3.87 (1.86)	4 [2,6]	
	WDBC	**5.00 (1.11)**	**5 [4,6]**	
	Clus Cntr	3.85 (1.46)	4 [3,5]	
Entmnt	Word	2.73 (1.37)	2 [2,4]	
	WDBC	**4.86 (1.18)**	**5 [4,6]**	
	Clus Cntr	4.14 (1.26)	4 [3,5]	
Science	Word	4.39 (1.80)	5 [3,6]	
	WDBC	4.58 (1.36)	5 [3,5]	
	Clus Cntr	**4.99 (1.45)**	**5 [5,6]**	
Sports	Word	3.54 (1.40)	3 [3,5]	
	WDBC	**4.91 (1.40)**	**5 [4,6]**	
	Clus Cntr	3.58 (1.57)	3 [2,5]	
Tech	Word	4.17 (1.49)	5 [3,5]	
	WDBC	**5.01 (1.29)**	**5 [4,6]**	
	Clus Cntr	3.99 (1.62)	4 [3,5]	

(a) Turk Study Summary Statistics - Round 1

	Algorithm	Mean (SD)	Median [IQR]	Histogram
Overall	**SBDC**	**4.76 (1.50)**	**5 [4,6]**	
	WDBC2	4.75 (1.50)	5 [4,6]	
	KSBT	4.63 (1.54)	5 [4,6]	
Politics	SBDC	4.61 (1.40)	5 [4,5]	
	WDBC2	**4.64 (1.54)**	**5 [3,6]**	
	KSBT	4.15 (1.53)	4 [3,5]	
Entmnt	**SBDC**	**4.84 (1.31)**	**5 [4,6]**	
	WDBC2	4.75 (1.51)	5 [4,6]	
	KSBT	4.33 (1.56)	5 [3,5]	
Science	SBDC	5.13 (1.38)	5 [4,6]	
	WDBC2	4.97 (1.27)	5 [4,6]	
	KSBT	**5.25 (1.31)**	**5 [5,6]**	
Sports	SBDC	4.51 (1.59)	4 [3,6]	
	WDBC2	4.39 (1.55)	5 [3,5]	
	KSBT	**5.04 (1.36)**	**5 [4,6]**	
Tech	SBDC	4.74 (1.74)	5 [4,6]	
	WDBC2	**5.02 (1.52)**	**5 [4,6]**	
	KSBT	4.42 (1.63)	5 [3,5]	

(b) Turk Study Summary Statistics - Round2

Table 3: Comparative User Study Results

Algorithm with largest mean score highlighted in bold — Overall refers to the combined dataset encompassing all 5 subject areas.

Round 1 – Algorithm Design

Our first approach, **Word Distance Based Clustering**, adapts the principles of summarization techniques for long, well-structured documents, to our problem space of single documents of unknown length and undefined, or nonexistent structure. As baselines for comparison, we use a commercially available summarization tool **MS Word** and our implementation of [23, 9] approaches for this context, **Cluster Center**.

Baseline - MS Word

To generate a summary using MS Word, we placed each document into an Office 2010 MS Word document. We then used the internal summarize feature to produce three sentences, which were used as that document's MS Word summary.

Baseline - Cluster Center

Following a parallel approach to that of TextRank[23] and LexRank[9], we created an approach that found the center of a cluster of documents (where each sentence was a node). However, TextRank and LexRank only produce one sentence as the center. In order to generate a fair comparison, we utilized k-means clustering to discover three cluster centers resulting from clustering sentences into three "topic" clusters.

We define a metric to measure sentence distance, analogous to LexRank's term frequency and TextRank's word co-occurence in TextRank. We use an information-theoretic definition of sentence distance [21] from the link clustering literature [16], and calculate the average of pairwise distance between words [22] for any two given sentences[18]. The pairwise distance between two words is calculated using [21], which uses WordNet (a graph of words linked by weighted edges based on semantic similarity) to find the semantic distance between the two concepts [19]. Thus, we can derive three cluster centers in a very similar approach to that of [23] and [9], which we refer to as Cluster Center.

[18]Only information heavy words (nouns and verbs) were used, and all words were lemmatized.

Word Distance Based Clustering (WDBC)

In order to create a new extraction-based, single document summarization algorithm we examined the types of features that other long-structured approaches have used. There are 4 main feature types that are used for the "selection of representative sentences" within long-structured documents [32]:

- **Thematic (semantic):** the meaning or content of the words
- **Location:** the relative or absolute location (physical placement) between words, sentences or paragraphs [20]
- **Headings/Structure:** using only elements that are explicitly providing sections or titles [8]
- **Cue Phrases:** probability of a sentence being relevant is determined by the presence of pragmatic words from a dictionary[19] *(e.g. above all, notably, unfortunately)* [8]

Using Location and Headings/Structure is not robust because attachments can be of variable length, variable structure, with/without headings and varied in content. We therefore focused on integrating the Thematic and Cue Phrases based approaches. Our algorithm, WDBC, specifically addresses the thematic features by building directly on TextRank[23] and LexRank[9] as follows:

1. *Extract document text from file and filter to include on information heavy words (nouns and verbs)*
2. *Lemmatize words to eliminate plurals, tense and conjugations*
3. *Remove extremely low frequency words*[20] *and low content sentences*[21].
4. *Compute similarity matrix between sentences and cluster using same technique as in Cluster Center*

[19]We utilized the Cue Phrase list from [18].

[20]A word is considered low frequency if it occurs less than 3 times or its frequency divided by total word count is less than 20%.

[21]At least 3 information heavy words.

	Algorithm Comparison	T-test
Round 1	Word vs WDBC	$<0.001^{\S}$
	Word vs Clus Cntr	$0.001^{\dagger}$
	WDBC vs Clus Cntr	$<0.001^{\S}$
Round 2	SBDC vs WDBC2	0.92
	SBDC vs KSBT	0.23
	WDBC2 vs KSBT	0.26
	WDBC vs WDBC2	0.23

Table 4: Turk Study Algorithm Comparisons
* *Less than 0.05,* † *Less than 0.01,* § *Less than 0.001*

5. *Within each cluster:*
   (a) *Remove sentences with less than 2 cue words (if no valid sentences, set threshold to 1, if still none, include all sentences)*[22]
   (b) *Take sentence with most unique words as representative sentence*
   (c) *If more than one sentence has the same number of unique words, take the one with the largest Inverse Term Frequency*

Mechanical Turk Study Design

Amazon Mechanical Turk (MT) HITs were constructed from each of the 20 documents. HITs were not grouped together so as to reduce order effects. A HIT consisted of the original source text, and the constructed summaries presented in random order. For each summary, participants were asked to respond to the statement *"The above three sentences give me a good overview of the article"* with a 7-point Likert scale (Strongly Disagree (1) to Strongly Agree (7))[23]. Each HIT was completed by 20 Turkers, yielding 400 measures of quality per summary (4 documents across 5 subject areas).

To ensure "legitimate" HIT completion, one "fake summary" was included with sentences extracted from other documents about different topics (e.g. a Science article having a summary from Sesame Street). These "fake summaries" were intended to be so outrageous that they would be ranked Strongly Disagree. If a Turker did not rate the "fake" summary as Strongly Disagree, then that response was thrown out and another HIT on the same document was posted to MT.

An ANOVA and Student's T-test are used to compare the algorithms' performance. While performing multiple comparisons may suggest statistical adjustment to a more conservative value (i.e., Bonferroni correction), we choose to highlight multiple thresholds of significance following [26]. **For transparency, we report t-test results and summary statistics broken down by subject area.** However, it is outside the scope of this paper to optimize for subject area.

Evaluation Limitations

Evaluating summarization presents a significant challenge, especially for large corpuses. This is mostly due to reviewers comparing the computer generated responses to their own mental images of an ideal human-generated summary. Therefore, receiving a perfect Strongly Agree is considered unlikely given the present standard of summarization tools.

Further, this study design is a controlled lab study, and like most lab studies has limited ecological validity in that these results do not account for the broader context of a document with a users workflow or email. In addition, we opted to use a human-centric evaluation technique rather than a gold standard based approach (e.g. ROUGE) due to the added complexity inherent in generating a high quality gold standard summary for each document. Our direct comparison technique finds the (statistically) best technique available to researchers, for integration within AttachMate.

Evaluation & Discussion - Round 1

Master level Turkers[24] were recruited to participate in the evaluation. Each completed HIT was paid 75 cents. 27 HITs were rejected for invalid responses to the "fake" summary. Results from this round are reported in Table 3a, which include mean, median and histograms of the distribution of MT responses. ANOVA comparing MS Word, WDBC and Cluster Center resulted in $p < 0.001$(F=56.15). Comparative t-test outputs between each algorithm are reported in first half of Table 4.

Overall WDBC performed quite well with a median score of 5, and a mean of 4.87. It is notable that WDBC statistically outperformed both MS Word and Cluster Center (the two baselines for comparison). In addition, when examining the histograms, inter quartile range and standard deviation, WDBC was much tighter as compared to the other existing techniques. While not a perfect score on the 7-point scale, which is challenging (as detailed earlier), WDBC is a stark and consistent improvement over alternative approaches.

Round 2 – Algorithm Design

Although high performing, WDBC has one major limitation – computing a similarity matrix between sentences (with average of pairwise distance between words) runs in $O(n^2 \log n)$ [22], and does not scale. While it runs in a matter of seconds on very short documents, it takes around 5 minutes on a 10 page CHI paper. We therefore wanted to determine if an alternative approach could be created that would run faster and perform as well as or better than WDBC. To this end, we created two new algorithms **Key Sentence By Thirds** (KSBT) and **SVD Based Distance and Clustering** (SBDC)[25]:

Key Sentence By Thirds (KSBT)

For KSBT, rather than clustering a document based on semantic distance of information heavy words (WDBC), we simplified our approach by dividing the document into thirds, based on the physical location of each sentence (first third, middle third, last third). This is an extremely fast method that leverages some sense of location [32]. Further, we streamline our representative selection process within each third using a

[22]While examining the raw text, few sentences had at least 2 or 3 cue words, and as the literature suggests, those sentences had high value.

[23]Ordered rank does not give relative distance/quality while individual assessment on an absolute scale does.

[24]Master level Turkers have 95% approval rate and a minimum of 1000 approve HITs

[25]Both approaches filter out non-information heavy words, and lemmatize remaining words.

proxy for semantic information, Singular Value Decomposition (SVD)[26], Cue Phrases, and location [32].

SVD is able to filter out the noise in relatively small or sparse data (often used as dimensionality reduction). SVD views each sentence as row in a sentence-word occurrence matrix (which can be calculated in O(n)), and its output can be used to calculate a weighted list of words, whose weight can be thought of as how "central" a word is to a document (a proxy for, though not exactly, semantic information [32]). We can then calculate the centrality of a sentence by summing the SVD output weights of the words in a given sentence. SVD values are calculated across the whole document.

The most representative sentence in each third is then selected by sorting all sentences (with an SVD score > 0 and Cue Phrases > 0) by the number of Cue Phrases present. Ties are broken by the sentence with the smallest distance (in number of sentences awry) to the start or end of the document (whichever is smaller). If there are no Cue Phrases > 0 or all are the same value, we select the most representative sentence by sorting all sentences by SVD score and taking the largest. Likewise, if all sentences have the same SVD score (or are all 0), we select the sentence with the highest Cue Phrase score.

SVD Based Distance and Clustering (SBDC)

At a conceptual level, KSBT's division of a document into three groups is extremely arbitrary. SBDC attempts to replace the document division by thirds with a clustering that is potentially more representative of distinct thematic parts. We begin by using SVD again to get a weighted list of words. Using the top 500 words from SVD, we create a similarity matrix of sentences, where the value in each cell is the cosine similarity between the vector representations of two given sentences. The vector representation of a sentence is the same as a row in the sentence-word occurrence matrix used in KSBT, except we use the weight of each word from SVD rather than just the number one, so that more "important" words get more impact[27].

Using this similarity matrix, we cluster sentences using k-means, into three thematic clusters. This approach has multiple benefits over the semantic approach in WDBC, the most notable being; the complexity is limited to the top 500 words, and; unlike the semantic distance calculation, SBDC does not need to calculate the distance of all pair-wise combinations of words in each sentence pair combination. Using the three clusters, we follow the same ranking approach used in KSBT.

Round 2 – Evaluation & Discussion

We conducted a second MT Study following the same protocol outlined earlier. This study compared KSBT and SBDC. In addition, we included the best performing algorithm from Round 1, WDBC, as a baseline for comparison (we refer to this in our tables as WDBC2). Turkers were recruited[28] with 95% approval rate and a minimum of 1000 approve HITs. Each completed HIT was paid 50 cents. 67 HITs were rejected for invalid responses to the "fake" summary. Results from this round are reported in Table 3b. ANOVA comparing WDBC2, KSBT and SBDC resulted in $p < 0.43$ (F=0.93). Comparative t-test output between each algorithm is reported in the second half of Table 4 to further highlight the lack of statistical difference found during the ANOVA.

In addition, we compared the performance of Word Distance Based Clustering in both experiments to see if the distribution of Turkers' responses are the same. The comparative T-test (Table 4) does not show statistical difference. However, because lack of statistical difference does not mean statistical similarity, we used Rita and Ekholm's measure of similarity [25][29]. This similarity metric utilizes a θ, or tolerance in the means between two data sets. We set a conservative θ to be one third a Likert interval (0.333). This represents $\frac{1}{18}$ (5.56%) of the possible answer range, and just 19.18% of the variance of Word Distance Based Clustering ($\sigma^2 = 1.74$) and 14.82% of the variance of WDBC2 ($\sigma^2 = 2.25$). The similarity test shows WDBC and WDBC2 are statistically similar ($p < 0.05$) as are WDBC2 vs. KSBT and WDBC2 vs. SBDC.

Summarization Algorithms Discussion – Overall

The Round 2 algorithms appear to have statistically equivalent performance to each other, and WDBC. However, both Round 2 approaches run faster, and scale better. Given that these Round 2 solutions run equally well (performance and quality), and perform better than existing standards (MS Word and Cluster Center), we need a metric by which to select the algorithm to use in AttachMate. Because KSBT's division is relatively arbitrary, we choose to use SBDC (which is more grounded) in the final version of AttachMate.

EXPERIMENT: ATTACHMATE DEPLOYMENT

In order to test the value and usage of AttachMate, we conducted a real-world, ecologically valid study in an enterprise setting. For experimental purposes, we instrumented the AMS to log attachment download access attempts as well as the number of senders and receivers of AttachMate email messages. Users' email addresses were not linked with the emails or attachments, and all activity was recorded using unique hashes of the sender (and receiver's) email addresses. This allows us to track individual users, while maintaining the required privacy and anonymity within HP.

AttachMate was deployed, and a broad invitation was sent out to all HP employees located in Palo Alto, CA to which 51 responded by filling out a demographic survey. Respondents were sent OAI download and install instructions. The OAI

[26]SVD is a more robust and semantically sensitive method for deriving word centrality as compared to raw word count as used in techniques like SumBasic[13].

[27]For example, a word with weight .6 that occurs 3 times in a sentence, would have a value of 1.8 in its bucket rather than 3.

[28]Master status was not required, however all were required to be living in the US, suggesting high familiarity with english.

[29]Rita and Ekholm's measure uses a similarity limit, θ such that the difference in the averages of the two data sets is smaller than θ in absolute value. This can be determined by examining the 95% confidence interval for the difference between the two data sets. If the θ is greater than the right 95% confidence interval and -θ is less than the left 95% confidence interval, 'there is a difference' with $p<0.05$.

		Mean (SD)	Median [IQ Range]	LN Histogram
Attachment	# Recipients	1.476 (0.722)	1.00 [1.00,2.00]	
	# Downloads (Unique IP)	0.314 (0.56)	0.00 [0.00,1.00]	
	Size (MB)	0.339 (1.01)	0.058 [0.018,0.223]	
	Traditional Footprint (MB)	1.60 (4.442)	0.262 [0.082,1.003]	
	AttachMate Footprint (MB)	1.216 (3.993)	0.196 [0.061,0.735]	
	Reduction %	70.897 (18.727)	75.00 [50.00,75.00]	
Document	# Recipients	1.356 (0.788)	1.00 [1.00,1.00]	
	# Downloads (Unique IP)	0.456 (0.633)	0.00 [0.00,1.00]	
	Size (MB)	0.415 (1.115)	0.084 [0.045,0.413]	
	Traditional Footprint (MB)	1.87 (4.544)	0.377 [0.216,1.672]	
	AttachMate Footprint (MB)	1.48 (4.41)	0.312 [0.164,1.333]	
	Reduction %	77.089 (18.057)	75.00 [75.00,90.00]	

Table 5: AttachMate Storage "Per File"
Data Normalized "Per User"— Traditional Footprint is with exchange infrastructure
Reduction Percentage is AttachMate/Traditional for each file

was downloaded 41 uniques times, with 28 unique AttachMate email senders during the study period. Due to privacy, we do not know which of the 51 respondents downloaded and used the OAI and all demographic information presented is from the full 51 respondents. As earlier, participation duration was left to the discretion of the individual, though we encouraged 5-10 business days of usage.

At the end of the study, a questionnaire was distributed to participants. This included Likert Scale, short answer, and SUS usability metric [4] questions. Due to the privacy limitations outline above, we were unable to directly followup with individual participants to ensure a high response rate. Subsequently, the survey was sent to all 51 initial respondents, resulting in only 6 survey completions (roughly 21% of unique senders). While this data may not be fully representative of all user experiences, we present results from the survey to help inform the observed behavior using AttachMate. In addition, due to the privacy concerns, we were unable to directly contact *recipients* of AttachMate emails to determine their reaction.

Demographics

Of the 51 individuals that responded to the survey, 54.9% were male. The average age was 40.99 (σ=10.43). As in prior studies, the educational attainment[30], subject area[31] and employment within HP[32] was highly variable, representing a broad cross-section of the company.

Results & Discussion: AttachMate Usage

On average, participants used AttachMate for 7.30 days each (median use length of six days). There were 28 unique senders, and 67 unique receivers of emails. Because each email can be sent to multiple recipients, it is important to examine AttachMate and the attachment usage from two distinct perspectives; those of the sender, and of the recipient. From the senders' perspective, 66 emails were sent using AttachMate, with a total of 105 attachments being sent (of which 73 were documents). Only 27.62% of the attachments and 38.36% of documents were downloaded. From the receivers' perspective, 93 emails were received, with a total of 155 attachments being received (99 of which were documents). Only 18.71% of attachments and 38.28% of documents were downloaded. These relatively low attachment download rates are well under the rate of 73% reported in the qualitative responses form users (Table 1) and real-world rate of 65.5% of documents in our enterprise logging study (presented earlier in the paper). This analysis strongly suggests a marked change in behavior when the only experimental intervention was AttachMate. This favorable change in user behavior suggests that AttachMate summaries were highly beneficial in information presentation *and* document discrimination, allowing users to discriminate which attachments to open and read.

Results & Discussion: AttachMate User Response

All participants mentioned the summarization of attachments to be the "best" feature of the AttachMate system. When presented with the statement *"Having Summaries is the key feature to AttachMate being successful"* and a 5-point Likert scale response, the average response was 4.6 (three participants marked 5 (strongly agree), two marked 4, and one marked 3). This is higher compared to other features such as Summary Quality (4.33), Saving Bandwidth (4.25), Mobile Access To Attachments (4.4). The only higher performing feature was Security of Files, to which all respondents reported 5 (Strongly Agree).

Mean users responses to the SUS usability metric was 70.00, which is considered slightly above average[4]. This was a surprisingly low score given the amount AttachMate was used, and the other responses on the survey. However, when we examined the short answer responses to *"What modifications are needed to make AttachMate Perfect?"*, participants universally mentioned two key features:

- Support for Microsoft PowerPoint and Excel files
- Preview Summaries Before Sending Emails

In short, the usability limitations stemmed from not supporting *all* file types and not presenting users with previews of the emails. Given that these two issues are the source of the slightly above average SUS score, we strongly believe with a large file type support[33] and a preview mode, the major issues of AttachMate can be addressed. Further, these issues do not appear to be intrinsic to or a reflection upon the summarization algorithm or the AttachMate system/infrastructure itself. This is codified in a short-answer response from one participant *"I really love the summarizer... What a cool job you have – inventing really valuable things."*

Results & Discussion: AttachMate Storage Benefits

While AttachMate's summarization system provides benefits for end users, its storage infrastructure provides financial benefits for their corporate employers. Table 5 shows the storage consumption for each file, normalized by user[34]. On average, documents are just under half a Megabyte in size. However, when we consider the multiple locations where the file is stored (sender's local sent folder, sender's exchange

[30] 49.02% BS/BA, 37.25% MS/MA, 5.88% PhD/MFA/JD, and 7.84% HS or HS Equivalent.

[31] 31.37% Business & Law, 21.57% CS, 15.69% Finance, 15.68% Science & Engineering 9.80% Humanities, 5.88% Communications

[32] 45.10% Management, 27.45% Engineering, IT and Research, 13.73% Business, 7.84 Admin % HR, 5.88% Finance

[33] Yet what a summary of an Excel or PowerPoint looks like is a question for future work.

[34] Data are normalized per user so that individuals who send more or less files have equal weight in the summary statistics.

sent folder, each receiver's server inbox, each receiver's local inbox), the average document footprint balloons to 1.87 megabytes. However, with AttachMate's improved storage this is reduced by 22.91% on a per file basis. Across all attachments, the reduction is larger, 29.10%. It should be noted, that this is without any redundant file optimization (only storing one copy of a duplicate file) enabled. This feature was not used during the study because it can only show impact over a large, ongoing dataset and the current experiment was too short and limited in participants.

DISCUSSION & FUTURE WORK

Overall, AttachMate successfully tackled the attachment file challenges uncovered earlier in this research. User responses, suggested that AttachMate is tackling a real problem, though the benefits of the AttachMate approach have not reached their full potential as additional types of files could benefit front the auto summarization process. This not only highlights the value of the summarization, but the success of our summarization technique (supporting the feedback from the MT studies). In addition, AttachMate reduces the data footprint of transferred documents by 22.91%, and 29.10% for all attachments. This is largely due to the provided summaries, which allow users to better triage which attachments need to be downloaded.

User feedback, and lessons learned through the development process, also suggest new directions for this work to expand. First, AttachMate should have a preview mode. When attachments need to be summarized, those summaries should be previewed by end-users. This additional mode could allow users to further refine and improve summaries by allowing users to see the "top N" highlights (as determined by the summarization algorithm). Subsequently, users could approve or replace sentences allowing for quick, but customized, summaries to be injected.

Another area of suggested feedback from our experiment is additional file types. While performance on PowerPoint slides may likely follow that of documents, determining what a "summary" of a spreadsheet looks like is unknown. This could be an open area of future work.

Lastly, if the AttachMate system is fully integrated with an exchange or other email service, attachments could be summarized if sent from non-AttachMate users. While this will not improve data storage costs, it will improve readability for end users, especially those on mobile devices. Further integration within a client can expand the "preview" mode, by visually showing users from where in the documents the summaries come from. This can allow users to keep summaries in context.

CONCLUSION

While much time and attention has been placed on the ubiquity of email within and beyond enterprise, little work has been done to address the growing problem of email attachments. These accompanying documents can be large (multiple megabytes), lengthy (multiple pages), and not optimized for the smaller screen sizes, limited reading time, or expensive bandwidth of mobile users. Thus, the triage of these overwhelming number of documents is left up to the end user, draining time, increasing storage/bandwidth costs, and if missed can cause important information to go unseen.

This paper presents the first system that directly addresses the problem of attachment overload, AttachMate. In addition to the system infrastructure itself, this research presents a novel single-unstructured document summarization algorithm that outperforms the existing approaches and commercial options. To better inform and ground this research, we present two upfront studies (813 participants) to understand the state and limitations of attachments, especially within enterprise. Lastly, a ecologically valid real-world deployment study of the entire AttachMate architecture was conducted with very encouraging results; reducing the data footprint, better informing users via summaries and helping users better discriminate which files to open.

While this research presents many contributions (study of attachments in enterprise, summarization algorithm, AttachMate system), and there are many exciting future directions this research can go, this work's primary contribution is the validation that email attachment overlaid is an important problem that can be addressed through grounded, robust and well tested solutions like AttachMate.

REFERENCES

1. Albakour, M.-D., Kruschwitz, U., and Lucas, S. Sentence-Level attachment prediction. In *IRFC'10*, Springer-Verlag (May 2010).
2. Bhattacharyya, M., Hershkop, S., and Eskin, E. MET: an experimental system for Malicious Email Tracking. In *NSPW '02: Proceedings of the 2002 workshop on New security paradigms*, ACM (Sept. 2002).
3. Blei, D. M., Ng, A. Y., and Jordan, M. I. Latent dirichlet allocation. *The Journal of Machine Learning Research 3* (Mar. 2003), 993–1022.
4. Brooke, J. SUS-A quick and dirty usability scale. *Usability evaluation in industry* (1996).
5. Chua, F., and Asur, S. Automatic Summarization of Events From Social Media. In *ICWSM* (2013).
6. Dabbish, L. A., Kraut, R. E., Fussell, S., and Kiesler, S. Understanding email use: predicting action on a message. In *CHI '05: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM Request Permissions (Apr. 2005).
7. Dredze, M., Lau, T., and Kushmerick, N. Automatically classifying emails into activities. In *IUI '06: Proceedings of the 11th international conference on Intelligent user interfaces*, ACM Request Permissions (Jan. 2006).
8. Edmundson, H. P. New Methods in Automatic Extracting. *Journal of the ACM (JACM 16*, 2 (Apr. 1969).
9. Erkan, G., and Radev, D. R. LexRank: Graph-based lexical centrality as salience in text summarization. *J Artif Intell Res(JAIR)* (2004).

10. Fiszman, M., Rindflesch, T. C., and Kilicoglu, H. Abstraction summarization for managing the biomedical research literature. In *Proceedings of the HLT-NAACL* ... (2004).
11. Gilbert, E. Phrases that signal workplace hierarchy. In *CSCW '12: Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, ACM Request Permissions (Feb. 2012).
12. Gore, K. Cogito Auto Sum. *Slate.com Accessed 1/31/2013*, `http://www.slate.com/articles/briefing/articles/1997/02/cogito_auto_sum.html`.
13. Haghighi, A., and Vanderwende, L. Exploring content models for multi-document summarization. *NAACL 2009* (2009).
14. Hailpern, J., and Huberman, B. A. Echo: the editor's wisdom with the elegance of a magazine. In *EICS '13: Proceedings of the 5th ACM SIGCHI symposium on Engineering interactive computing systems*, ACM Request Permissions (June 2013).
15. Hangal, S., Lam, M. S., and Heer, J. MUSE: reviving memories using email archives. In *UIST '11: Proceedings of the 24th annual ACM symposium on User interface software and technology*, ACM Request Permissions (Oct. 2011).
16. Jain, A. K., Murty, M. N., and Flynn, P. J. Data clustering: a review. *Computing Surveys (CSUR 31*, 3 (Sept. 1999).
17. Katragadda, R., Pingali, P., and Varma, V. Sentence position revisited: a robust light-weight update summarization'baseline'algorithm. In *Proceedings of the Third* ... (2009).
18. Knott, A. *A Data-Driven Methodology for Motivating a Set of Coherence Relations*. PhD thesis, The University of Edinburgh: College of Science and Engineering: The School of Informatics, July 1996.
19. Leacock, C., Miller, G. A., and Chodorow, M. Using corpus statistics and WordNet relations for sense identification. *Computational Linguistics 24*, 1 (Mar. 1998), 147–165.
20. Lin, C.-Y., and Hovy, E. Identifying topics by position. In *ANLC '97: Proceedings of the fifth conference on Applied natural language processing*, Association for Computational Linguistics (Mar. 1997).
21. Lin, D. An information-theoretic definition of similarity. In *ICML* (1998).
22. Manning, C. D., Raghavan, P., and Schütze, H. *Introduction to Information Retrieval*. Cambridge University Press, July 2008.
23. Mihalcea, R., and Tarau, P. TextRank: Bringing order into texts. In *Proceedings of EMNLP* (2004).
24. Nenkova, A., and Vanderwende, L. The impact of frequency on summarization. Tech. Rep. MSR-TR-2005-101, Microsoft Research, 2005.
25. Rita, H., and Ekholm, P. Showing similarity of results given by two methods: A commentary. *Environmental pollution* (2007).
26. Savitz, D. A., and Olshan, A. F. Multiple comparisons and related issues in the interpretation of epidemiologic data. *American Journal of Epidemiology* (1995).
27. Seki, Y., Eguchi, K., and Kando, N. Compact summarization for mobile phones. *Mobile and ubiquitous information access* (2004).
28. Tang, J. C., Lin, J., Pierce, J., Whittaker, S., and Drews, C. Recent shortcuts: using recent interactions to support shared activities. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM Request Permissions (Apr. 2007).
29. Whittaker, S., Bellotti, V., and Moody, P. Introduction to this special issue on revisiting and reinventing e-mail. *Human-Computer Interaction 20*, 1 (June 2005).
30. Whittaker, S., and Sidner, C. Email overload: exploring personal information management of email. In *CHI '96: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM Request Permissions (Apr. 1996).
31. Yang, C. C., and Wang, F. L. Automatic summarization of financial news delivery on mobile devices. In *WWW'03* (2003).
32. Yang, C. C., and Wang, F. L. Hierarchical summarization of large documents. *Journal of the American Society for Information Science and Technology 59*, 6 (Apr. 2008).