A Publicly Available Annotated Corpus for Supervised Email Summarization

Jan Ulrich, Gabriel Murray, and Giuseppe Carenini
Department of Computer Science
University of British Columbia, Canada
{ulrichj, gabrielm, carenini}@cs.ubc.ca

Abstract

Annotated email corpora are necessary for evaluation and training of machine learning summarization techniques. The scarcity of corpora has been a limiting factor for research in this field. We describe our process of creating a new annotated email thread corpus that will be made publicly available. We present the trade-offs of the different annotation methods that could be used.

Introduction

Email has become a part of most people's everyday lives. With its widespread use, the ability to manage email data efficiently has become paramount. Summarization provides one tool for reducing the information overload. A summary of an email thread limits itself to the salient parts of the email conversation. Email threads are of particular interest in summarization research because there is a great deal of structural redundancy due to their conversational nature.

Summarization can be divided into two different types: extractive and abstractive. In extractive summarization the important sentences are taken from the original document to form the summary. This means that the summary sentences simply constitute a subset of the source document sentences with little or no sentence transformation involved. Because the sentences are removed from their original contexts, summary coherence can suffer, but automatic re-ordering of the sentences can attempt to maximize coherence.

In an abstractive summary, the document is rewritten in a more concise form. Although an abstractive summary could achieve higher content compression, extractive summaries have been more feasible and are therefore the standard in multi-document summarization. Extractive techniques tend to be less domain dependent and do not require a deep understanding of the source material.

Extractive summarization lends itself well to machine learning as each sentence can be assigned a score and the highest scoring sentences chosen for the summary. Although these methods work well and have been used in email summarization (Rambow et al. 2004), a corpus is required for training the algorithms. Machine learning is an effective summarization technique, yet the bottleneck is that there is a very limited number of corpora available for training the algorithms. A corpus used for training usually also needs to match the material needed to be summarized.

Email Thread Summarization

Email thread summarization focuses on creating a summary based on the entire email conversation. While a single email might be short enough to be read through entirely, it is much more time consuming to read through an entire conversation. A summary can be useful for reviewing a current conversation before responding to it. In a corporate sense it can be used as corporate memory to review previous decisions. It can also be used as an index of the original emails for fast overview of email mailboxes.

Among the two types of summarization, abstractive and extractive, there has been much more work on extractive summarization for email threads. These approaches rely on features of sentences to determine which are the most important sentences to keep in a summary. One approach is to find a good feature and use it to choose the most important sentences. This is the approach taken by (Carenini, Ng, and Zhou 2007) with their Clue Word Summarizer. The Clue Word score is based on the occurrence of similar words in adjacent nodes in an email fragment quotation graph. An email fragment quotation graph represents the conversation structure more precisely than at the email level. This kind of feature can be computed for any email thread.

Another extractive summarization technique uses machine learning to decide among a combination of features for scoring the importance of email sentences. In order to train the machine learning method, annotated email data needs to be available. Although an annotated corpus is necessary for evaluation in both methods, more data is required for machine learning as the evaluation data needs to be different from the training data. The machine learning approach to email thread summarization presented in (Rambow et al. 2004) relied on the Ripper algorithm. They show that using email specific features, such as the number of recipients and the number of responses, improves summarization results compared to just multi-document text features, such as centroid similarity and length.

Other research on email summarization by (Corston-Oliver et al. 2004) has focused on task oriented summaries which combine extractive and abstractive techniques. Important sentences were extracted using an SVM and then reformulated into a task oriented imperative. In this way tasks could automatically be populated into the Outlook ”to-do” list.

The Need For an Annotated Corpus

While researchers ideally want to work with realistic, naturally-occurring data, privacy concerns mean that large collections of real email data cannot typically be shared. This has left the field with each project training and testing new approaches on its own dataset. With this lack of large research datasets comes the risk of overfitting algorithms to the limited data. We believe that a publicly available email corpus, complete with summarization annotations for training and evaluation purposes, would greatly benefit the research community. The more public benchmarks are available, the easier it is to compare diverging approaches to the summarization task. Here we present our current email annotation project, describing the data used, the annotation scheme employed, and the details of the user study.

Available Email Corpora

Below we describe several relevant datasets that are publicly available, and assess their suitability for further annotation and research.

The Enron Corpus

The most popular email corpus for research has been the Enron dataset (Klimt and Y.Yang 2004), which was released during the legal investigation into the Enron corporation. This is an invaluable dataset since it contains uncensored messages from a corporate environment. Because this corpus consists of real, unaltered data, there were initially integrity problems that needed to be fixed. The dataset contains over 30,000 threads, but they end up being rather short with an average thread size of just over four and a median of two. The dataset consists of employee’s email folders, so it is also an accurate depiction of how users use folders.

TREC Enterprise Data

In the TREC Enterprise Track[1], the task is to search through an organization’s multimodal data in order to satisfy a given information need. Organizations often possess a great variety of data, from emails to web pages to spreadsheets and word documents. Being able to search, summarize and link these documents can greatly aid corporate memory. In recent years, TREC Enterprise has used two multimodal datasets for Question-Answer (QA) tasks.

The W3C Corpus The W3C corpus is data derived from a crawl of the World Wide Web Consortium’s sites at w3c.org. The data include mailing lists, public webpages, and text derived from .pdf, .doc and .ppt files, among other types. The mailing list subset is comprised of nearly 200,000 documents, and TREC participants have provided thread structure based on reply-to relations and subject overlap. There are more than 50,000 threads in total. W3C data has been annotated for QA topic relevance for use in TREC Enterprise 2005 and 2006.

The CSIRO Corpus Subsequent to TREC 2006, TREC Enterprise began using similar data from the Australian organization CSIRO. Because the QA topics and relevance annotations were drawn up by CSIRO employees themselves rather than by outside annotators, this dataset and annotation together represent more realistic information needs.

The main drawback of both the W3C and CSIRO corpora for summarization annotation purposes is that the content of the data can be quite technical, posing a problem for annotators unfamiliar with the organizations and the subjects discussed in the data. However, there is a sufficient number of emails in each corpus that the data could be filtered to create corpora subsets comprised of less technical emails.

Summarizing mailing lists can also be considered to be a slightly different task than summarizing emails from a user’s inbox. In the latter case there will be missing or hidden data that can sometimes be reconstructed, while this is much less of an issue with a mailing list.

PW Calo Corpus

As part of the CALO project[2], an email corpus was gathered during a four-day exercise at SRI, in which project participants took part in role-playing exercises and group activities. The resultant PW Calo corpus (Cohen, Carvalho, and Mitchell 2004) totals 222 email messages. The dataset also includes correlated meeting recordings, but currently the corpus is not freely available.

Previous Email Thread Annotations

The Enron corpus has previously been annotated for summarization (Carenini, Ng, and Zhou 2007). A subset of threads were selected, and important sentences were ranked by annotators in a user study. Each thread was summarized (using extraction) by 5 annotators which were asked to select 30% of the original number of sentences and label them as essential or optional. Each sentence was then given an overall importance score based on the combined annotations. An essential sentence carried three times the weight of an optional sentence. Using several annotators is assumed to produce a more meaningful result by diminishing the effect of individual differences. 39 threads were annotated in this way to produce a corpus for evaluation.

This annotation is targeted towards extractive summarization and it is hard to use with an abstractive summarization technique. It also has several arbitrary restrictions including the requirement of 30% sentence selection as well as only two choices for sentence importance. The weight of an essential sentence compared to an optional sentence is also arbitrary. Annotators might have a different conception of what is an optional sentence and an essential sentence. However the annotator’s directions are straight-forward and the annotation of selecting sentences is easier for the annotator than writing a summary from scratch.

[1] http://www.ins.cwi.nl/projects/trec-ent/

[2] http://caloproject.sri.com

Annotation Uses

For email summarization the annotation of an email thread is a thread summary. This summary is the model for a summarization technique and is used to evaluate different algorithms. One evaluation technique, called ROUGE (Lin and Hovy 2003), compares the n-gram overlap between the produced summary and the model summary. This is a bag of words approach which does not tell much about the coherence of the summary, but has been useful in multi-document summarization and is fast and easy to compute.

One well-known issue with annotation is that annotators often do not agree on the same annotation, reflecting the fact that there is no single best summary for a given source document. The solution is to use multiple annotators and combine their annotation. This has been done in the pyramids method (Nenkova, Passonneau, and McKeown 2007) where a proposed summary is evaluated based on the occurrence of summarization content units. Summary content units are parts of a sentence that express one idea.

Similarly, one can calculate weighted f-score (Murray and Renals 2007) using multiple extractive gold-standard summaries. The motivating idea for this metric is that a good machine summary might contain certain sentences from one gold-standard, other sentences from a second gold-standard, and so on.

Annotation is not only needed for evaluation, but is also useful for creating the actual summarization algorithm. Labeled email threads can be used as training for machine learning methods. Typically, the more data is available, the better result the algorithms achieve. The work of (Corston-Oliver et al. 2004) and (Rambow et al. 2004) are two examples of using annotated email thread corpora to train summarization algorithms.

Our Proposed Annotation

Abstractive summarization is the goal of many researchers since it is what people do naturally. However extractive summarization has been more successful and effective since it is easier to compute. A key goal is to successfully combine extraction with abstraction. In order to do this a corpus is needed that supports such a goal.

In our annotation effort we ask the annotator not only to select the most important sentences from the original text but also to write an abstractive summary of the thread. Links are then formed between each human written sentence and corresponding extracted sentences containing the same information. Extracted sentences can then be weighed by the number of times that they are linked. It also shows the transformation of each idea from the original text to the corresponding summary sentence. By having to form these links after the summary was written, it makes the annotator justify both their extractive choices as well as their written summary. Such an annotation approach has been used in the ICSI (Janin et al. 2003) and AMI (Carletta et al. 2005) corpora in meeting summarization (Murray et al. 2006).

We also propose to annotate several features which will hopefully be useful in machine learning summarization techniques.

Meta Sentences

In work on automatic summarization of meeting speech, Murray and Renals (Murray and Renals to appear 2008) made use of meta comments within meetings. Meta comments are cases where a speaker is referring explicitly to the meeting itself or to the discussion itself, e.g. “So we’ve decided on the user interface.” They found that including these comments in meeting summaries improved summary quality according to several metrics. In order to see if a similar benefit can be found in email summarization, we have enlisted our annotators to label sentences as meta or non-meta, where a meta sentence refers explicitly to the email discussion itself. We hypothesize that the detection of such sentences may aid the coherence of our extractive summaries.

Speech Acts

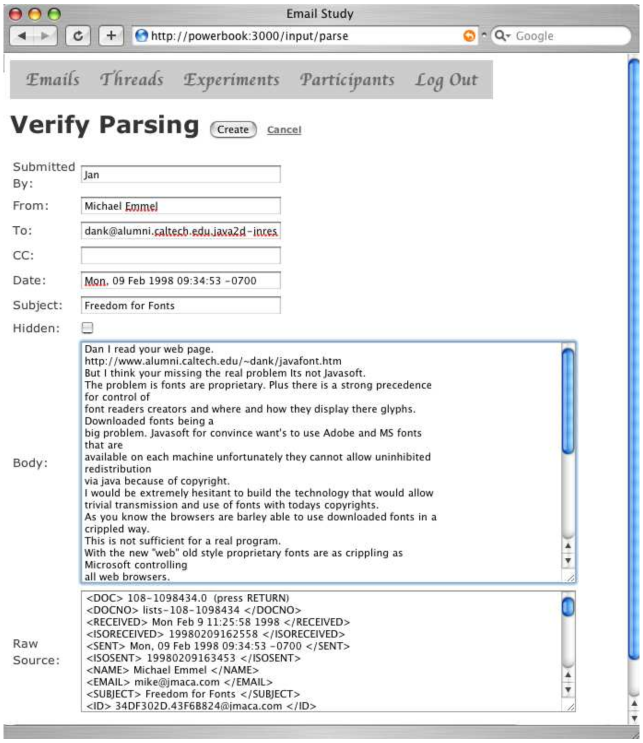
Emails are a modern form of communication. Compared with online messengers or blogs, email is a more formal communication method. When people use email, they usually have a specific purpose in mind. This specific purpose is often shrouded in small talk to make a request less direct. In summarization the specific purpose of an email is the important part. We therefore annotate sentences based on their speech act. Previous work has classified an entire email into a specific speech act (Carvalho and Cohen 2005). However a more detailed annotation at the sentence level is useful for summarization. In the original work, emails were classified as *Propose*, *Request*, *Commit*, *Deliver*, *Meeting*, and *deliveredData*. These categories are not mutually exclusive. We propose to use a subset of these speech acts which can not be computed automatically and are information rich. The annotation would consist of the following categories: *Propose*, *Request*, *Commit*, *Agreement/Disagreement* and *Meeting*. A *Propose* message proposes a joint activity; a *Request* asks the recipient to perform an activity; a *Commit* message commits the sender to some future course of action; an *Agreement/Disagreement* is an agreement or disagreement with a joint activity; and a *Meeting* message is regarding a joint activity in time or space. *Deliver* is excluded because most emails deliver some sort of information and *deliveredData* is excluded because attachments or hyperlinks can be automatically parsed from an email message.

The order of these annotations also becomes interesting in summarization, as seen in (Carvalho and Cohen 2005). If a request is chosen as an important sentence then the corresponding commit should also be included in the summary. This sequential relationship can be obtained from the email ordering in the threads.

We are also considering annotating sentences according to subjectivity.

The W3C Corpus Data

We decided to use the W3C corpus for our summarization annotation project. A central reason for choosing this data is that, even though we are annotating only a small portion of the mailing list threads, having a very large amount of related but unannotated data could prove very beneficial for



Verify Parsing

Submitted By: Jan

From: Michael Emmel

To: dank@alumni.caltech.edu.java2d-inres

CC:

Date: Mon, 09 Feb 1998 09:34:53 -0700

Subject: Freedom for Fonts

Hidden: □

Body:

Dan I read your web page.
http://www.alumni.caltech.edu/~dank/javafont.htm
But I think your missing the real problem Its not Javasoft.
The problem is fonts are proprietary. Plus there is a strong precedence
for control of
font readers creators and where and how they display there glyphs.
Downloaded fonts being a
big problem. Javasoft for convince want's to use Adobe and MS fonts
that are
available on each machine unfortunately they cannot allow uninhibited
redistribution
via java because of copyright.
I would be extremely hesitant to build the technology that would allow
trivial transmission and use of fonts with todays copyrights.
As you know the browsers are barley able to use downloaded fonts in a
crippled way.
This is not sufficient for a real program.
With the new "web" old style proprietary fonts are as crippling as
Microsoft controlling
all web browsers.

Raw Source:

```
<DOC> 108-1098434.0  (press RETURN)
<DOCNO> lists-108-1098434 </DOCNO>
<RECEIVED> Mon Feb 9 11:25:58 1998 </RECEIVED>
<ISORECEIVED> 19980209162558 </ISORECEIVED>
<SENT> Mon, 09 Feb 1998 09:34:53 -0700 </SENT>
<ISOSENT> 19980209163453 </ISOSENT>
<NAME> Michael Emmel </NAME>
<EMAIL> mike@jmaca.com </EMAIL>
<SUBJECT> Freedom for Fonts </SUBJECT>
<ID> 34DF302D.43F6B824@jmaca.com </ID>
```

Figure 1: Imported emails automatically get parsed into fields and their sentences get segmented.

the development and refinement of summarization systems. Though much of the W3C email data is very technical, we ultimately chose 30 threads that were somewhat less technical than average, so that the annotators would face less difficulty in making their summarization annotations. An additional criterion for selecting these threads is that they are sufficiently long to justify being summarized. The average number of emails per thread in this subset is just under 11, and the number of participants per thread is just under 6.

Before the data is annotated, some pre-processing is needed. Because the automatic sentence segmenters we applied yielded sub-optimal results on email data, we carried out manual sentence segmentation on the relevant emails. We also formatted the emails so that all quoted text is in the same format, preceded by one or more quote brackets, e.g. >>.

The Annotation GUI Interface

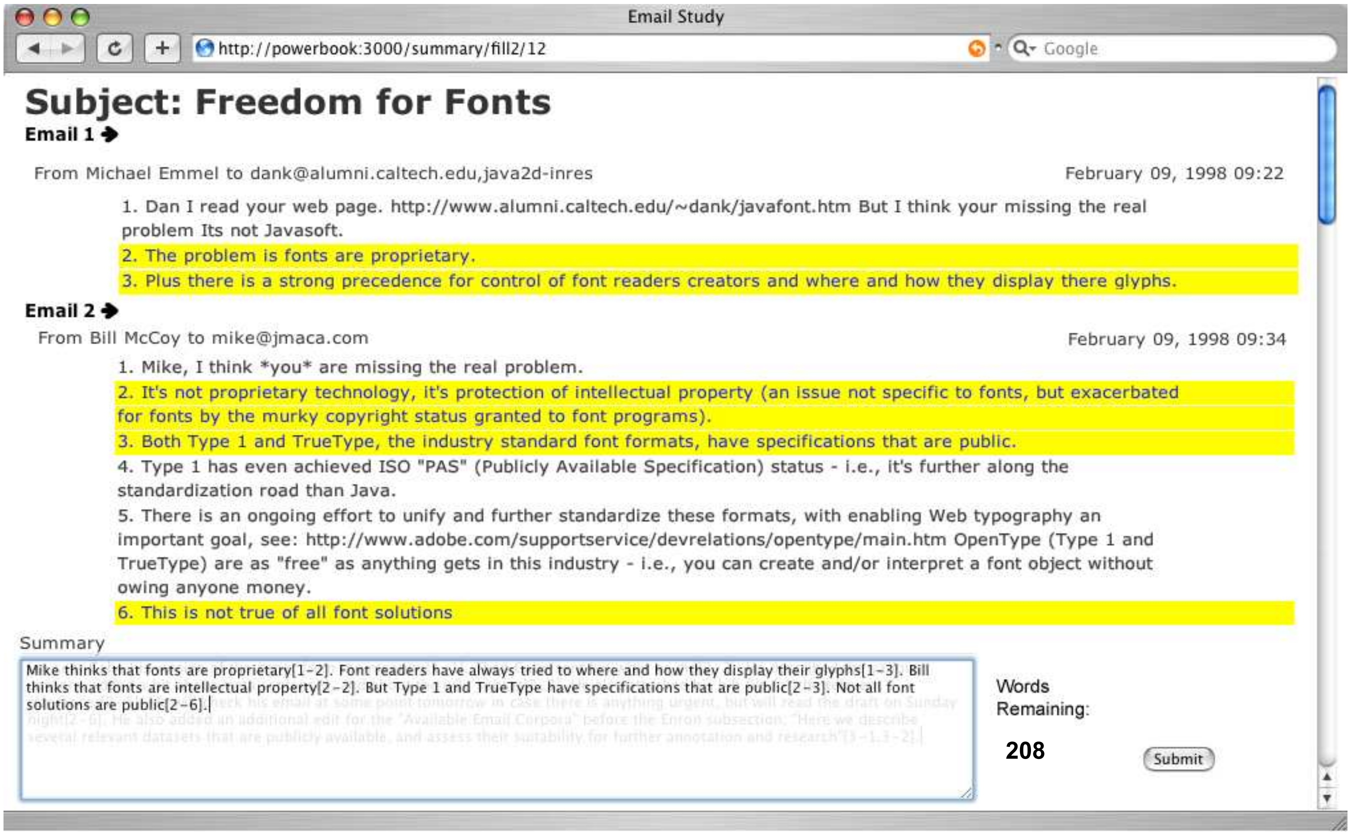
A web based interface is created for the annotation. This is convenient for the annotator whose job of highlighting selected sentences is made easier. It is also good for the researcher since the data is automatically compiled and ready to use for research purposes. The interface is a web database application written in Ruby on Rails which processes the original emails and records the annotation results. The researcher imports emails in raw source or XML format and the application parses the header fields. The researcher then combines the emails into the threads that need to be summarized. This database of emails can then be used to run annotation studies. The researcher selects which threads should be summarized by the current annotator and then starts the annotation study. Now the researcher is logged out of the web application for security purposes and the annotator can take over to perform the annotation. The results are stored in an MySQL database which can transform the results to the format needed by the researcher. The researcher can also log in and view past results from previous annotators. The interface therefore allows for complete control over the annotation process.

The Annotation Study

In our study we plan to annotate the 30 threads we have selected from the W3C corpus. Each thread will be annotated by three different annotators to reduce overall subjectivity. The annotation study will be run on individual computer stations and annotators will be supervised by researchers. The study consists of two steps: a summarization step consisting of both abstractive and extractive summarization followed by a feature collection step.

First the annotator is asked to fill out a profile of their personal information and given instructions on how to annotate. An example is provided to go through all the annotation steps before the annotator starts with annotating actual threads. The annotation starts with the email thread being displayed allowing each email to be collapsed for navigational ease. The annotator is expected to read the entire email conversation to become familiar with the content. The threads are also provided as a hard copy for reference and to make notes. In the first stage annotators are asked to write a 250 word summary describing the information contained in the email. This summary length is chosen because it matches the length of the DUC2007 human written model summaries in the main summarization task. Then each sentence in the written summary has to be linked to at least one original sentence with the same content. The annotator does this by putting the corresponding sentence numbers behind each written sentence. Annotators are then asked to create an extractive summary by selecting important sentences from the original text (Figure 2). There are no restrictions on the number of sentences which can be extracted. Extracted sentences do not necessarily have to be linked nor do linked sentences have to be extracted. After annotators create the summaries, they are asked to label the meta sentences and speech acts which include: *Proposal*, *Request*, *Commit*, and *Meeting* (Figure 3). Therefore the user creates an abstractive and extractive summary with links between them as well as additional labeled features.

The study will be performed with 10 annotators who each annotate 9 different threads. They will be recruited from the University of British Columbia's Departments of Computer Science and Psychology. Annotators will be screened to have satisfactory writing skills. The work will be spread over several days and the annotators will be compensated for their work.



Subject: Freedom for Fonts

Email 1 ➔

From Michael Emmel to dank@alumni.caltech.edu,java2d-inres February 09, 1998 09:22

1. Dan I read your web page. http://www.alumni.caltech.edu/~dank/javafont.htm But I think your missing the real problem Its not Javasoft.
2. The problem is fonts are proprietary.
3. Plus there is a strong precedence for control of font readers creators and where and how they display there glyphs.

Email 2 ➔

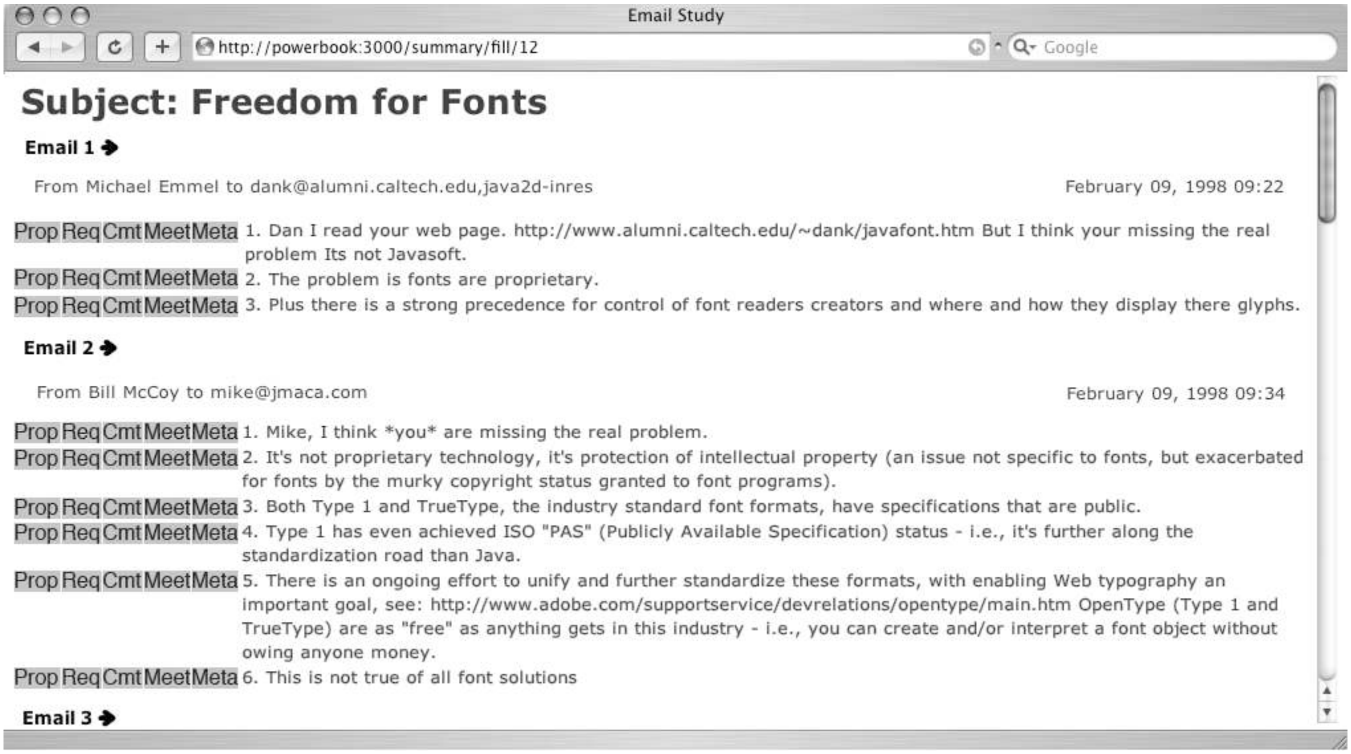
From Bill McCoy to mike@jmaca.com February 09, 1998 09:34

1. Mike, I think *you* are missing the real problem.
2. It's not proprietary technology, it's protection of intellectual property (an issue not specific to fonts, but exacerbated for fonts by the murky copyright status granted to font programs).
3. Both Type 1 and TrueType, the industry standard font formats, have specifications that are public.
4. Type 1 has even achieved ISO "PAS" (Publicly Available Specification) status - i.e., it's further along the standardization road than Java.
5. There is an ongoing effort to unify and further standardize these formats, with enabling Web typography an important goal, see: http://www.adobe.com/supportservice/devrelations/opentype/main.htm OpenType (Type 1 and TrueType) are as "free" as anything gets in this industry - i.e., you can create and/or interpret a font object without owing anyone money.
6. This is not true of all font solutions

Summary

Mike thinks that fonts are proprietary[1-2]. Font readers have always tried to where and how they display their glyphs[1-3]. Bill thinks that fonts are intellectual property[2-2]. But Type 1 and TrueType have specifications that are public[2-3]. Not all font solutions are public[2-6].

Figure 2: Summarizing the email thread with both abstractive and extractive summaries.



Subject: Freedom for Fonts

Email 1 ➔

From Michael Emmel to dank@alumni.caltech.edu,java2d-inres February 09, 1998 09:22

1. Dan I read your web page. http://www.alumni.caltech.edu/~dank/javafont.htm But I think your missing the real problem Its not Javasoft.
2. The problem is fonts are proprietary.
3. Plus there is a strong precedence for control of font readers creators and where and how they display there glyphs.

Email 2 ➔

From Bill McCoy to mike@jmaca.com February 09, 1998 09:34

1. Mike, I think *you* are missing the real problem.
2. It's not proprietary technology, it's protection of intellectual property (an issue not specific to fonts, but exacerbated for fonts by the murky copyright status granted to font programs).
3. Both Type 1 and TrueType, the industry standard font formats, have specifications that are public.
4. Type 1 has even achieved ISO "PAS" (Publicly Available Specification) status - i.e., it's further along the standardization road than Java.
5. There is an ongoing effort to unify and further standardize these formats, with enabling Web typography an important goal, see: http://www.adobe.com/supportservice/devrelations/opentype/main.htm OpenType (Type 1 and TrueType) are as "free" as anything gets in this industry - i.e., you can create and/or interpret a font object without owing anyone money.
6. This is not true of all font solutions

Email 3 ➔

Figure 3: Annotating the email thread for speech act and meta features.

Conclusion

Due to the scarcity of annotated corpora for email thread summarization, we have decided to annotate a subset of the W3C corpus, a large publicly available email dataset. Having learned from previous corpora annotations, we have chosen to annotate the corpus with both extractive and abstractive summaries. Annotators will have to link sentences containing the same information in the abstractive summary and extractive summary. We have also chosen to label additional features that can be used in machine learning based summarization. These include the basic speech acts of individual sentences as well as whether a sentence is a meta sentence, i.e. referring to the current conversation. We will be reporting the results of this study at the workshop.

Availability

We plan to release this annotated corpus for research purposes. The annotation framework will also be made publicly available to promote further annotation. W3C is a huge dataset which means there are many more threads that can be annotated.

References

Carenini, G.; Ng, R. T.; and Zhou, X. 2007. Summarizing email conversations with clue words. *16th International World Wide Web Conference (ACM WWW'07)*.

Carletta, J.; Ashby, S.; Bourban, S.; Flynn, M.; Guillemot, M.; Hain, T.; Kadlec, J.; Karaiskos, V.; Kraaij, W.; Kronenthal, M.; Lathoud, G.; Lincoln, M.; Lisowska, A.; McCowan, I.; Post, W.; Reidsma, D.; and Wellner, P. 2005. The AMI meeting corpus: A pre-announcement. In *Proc. of MLMI 2005, Edinburgh, UK*, 28–39.

Carvalho, V. R., and Cohen, W. W. 2005. On the collective classification of email ”speech acts”. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, 345–352. New York, NY, USA: ACM.

Cohen, W.; Carvalho, V.; and Mitchell, T. 2004. Learning to classify email into “speech acts”. In Lin, D., and Wu, D., eds., *Proc. of EMNLP 2004*, 309–316. Barcelona, Spain: Association for Computational Linguistics.

Corston-Oliver, S.; Ringger, E.; Gamon, M.; and Campbell, R. 2004. Task-focused summarization of email. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*.

Janin, A.; Baron, D.; Edwards, J.; Ellis, D.; Gelbart, D.; Morgan, N.; Peskin, B.; Pfau, T.; Shriberg, E.; Stolcke, A.; and Wooters, C. 2003. The ICSI meeting corpus. In *Proc. of IEEE ICASSP 2003, Hong Kong, China*, 364–367.

Klimt, B., and Y.Yang. 2004. *The enron corpus: A new dataset for email classification research.* 217–226.

Lin, C.-Y., and Hovy, E. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. *Proceedings of Language Technology Conference (HLT-NAACL)*.

Murray, G., and Renals, S. 2007. Term-weighting for summarization of multi-party spoken dialogues. In *Proc. of MLMI 2007, Brno, Czech Republic*, 155–166.

Murray, G., and Renals, S. to appear, 2008. Meta comments for summarizing meeting speech. In *Proc. of MLMI 2008, Utrecht, Netherlands*.

Murray, G.; Renals, S.; Carletta, J.; and Moore, J. 2006. Incorporating speaker and discourse features into speech summarization. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, 367–374. Morristown, NJ, USA: Association for Computational Linguistics.

Nenkova, A.; Passonneau, R.; and McKeown, K. 2007. The pyramid method: Incorporating human content selection variation in summarization evaluation. *ACM Trans. on Speech and Language Processing (TSLP)*.

Rambow, O.; Shrestha, L.; Chen, J.; and Lauridsen, C. 2004. Summarizing email threads. *Proceedings of HLT-NAACL 2004*.