Generating and Evaluating Summaries for Partial Email Threads: Conversational Bayesian Surprise and Silver Standards

Jordon Johnson, Vaden Masrani, Giuseppe Carenini, and Raymond Ng
Department of Computer Science
University of British Columbia
Vancouver, British Columbia, Canada
{jordon, vadmas, carenini, rng}@cs.ubc.ca

Abstract

We define and motivate the problem of summarizing partial email threads. This problem introduces the challenge of generating reference summaries for partial threads when human annotation is only available for the threads as a whole, particularly when the human-selected sentences are not uniformly distributed within the threads. We propose an oracular algorithm for generating these reference summaries with arbitrary length, and we are making the resulting dataset publicly available[1]. In addition, we apply a recent unsupervised method based on Bayesian Surprise that incorporates background knowledge into partial thread summarization, extend it with conversational features, and modify the mechanism by which it handles redundancy. Experiments with our method indicate improved performance over the baseline for shorter partial threads; and our results suggest that the potential benefits of background knowledge to partial thread summarization should be further investigated with larger datasets.

1 Introduction

Despite the relatively early advent of emails compared to other forms of electronic communication, the continued proliferation of emails make them an ongoing focus of NLP research. With users experiencing an increasing flow of emails and decreasing screen sizes, there has been a growing interest in the *email summarization task*: given an email thread with multiple participants, provide a summary of the contents of the thread. Such summaries should contain the key information in a thread and free a user from having to comb through its entire contents. Also, given that email threads can span days, weeks, or months, and users often participate in multiple threads at once, such summaries can serve as memory aids to users returning to or joining a thread in progress (Ulrich et al., 2008).

Email threads are dynamic document collections, however, and the content of a summary may need to change over time as emails come in. Therefore, while the *full thread summarization problem* (extensively studied in the past as discussed in section 2) provides a single summary of a complete, archived email thread, we are interested in the *partial thread summarization problem* where we generate a succession of summaries, each summarizing the thread at different moments in time. More formally, for each email E_i in a given email thread $\{E_1...E_i...E_n\}$ we wish to generate a summary for the corresponding *partial thread* (PT) $\{E_1...E_i\}$. Given the novelty of the summarization task, in this paper we focus on investigating simple unsupervised extractive approaches, where the summary is a subset of the sentences in the source partial thread, and leave supervised and abstractive approaches for future work.

A *partial thread summary* will provide a summary of the thread so far, including the new email; it is intended to benefit users that may have forgotten the content of the preceding emails in the thread (or may be new to the thread) and need a quick refresh, possibly on the relatively small screen of a mobile device. Additionally, a user may want to "extend" a partial thread summary in order to get more information; and so we also investigate the ability to generate summaries of arbitrary length. The PT summarization problem is thus different from the *update summariza-*

[1] http://www.cs.ubc.ca/cs-research/lci/research-groups/natural-language-processing/Software.html

Proceedings of the SIGDIAL 2017 Conference, pages 263–272,
Saarbrücken, Germany, 15-17 August 2017.

tion problem previously studied for news in the Text Analysis Conferences (Dang and Owczarzak, 2008). The update summarization problem, applied to email threads, would provide a summary of *only* the incoming email with the assumption that the user knows and remembers the content of the preceding emails.

The new NLP task of summarizing email PT is challenging, not only because new algorithms may need to be developed, but also with respect to evaluating the generated summaries. While there are publicly available datasets - including BC3 (Ulrich et al., 2008) and an Enron-derived dataset (Loza et al., 2014) - that provide gold standard summaries for completed email threads, none to our knowledge provides such summaries for PTs; such annotation by humans would be prohibitive, as it would require a summary for each partial thread (i.e., each email) in the corpus. So, a challenge we face in the evaluation of PT summaries is due to the dearth of human annotations. More specifically, given gold standard human annotations of a thread as a whole, how do we generate reference summaries of each PT against which to compare automatically generated extractive summaries?

Most current summarization techniques for full thread summarization rely on the analysis of only the content of the input thread to decide what sentences should be included in the summary. However, since PT can be rather short we hypothesize that the identification of the most informative sentences would benefit from examining the larger informational context in which the PT was generated (eg. all the email generated in an organization). We test this hypothesis by applying and extending a recent summarization method based on Bayesian Surprise that leverages such background information for PT summarization.

The main contributions of this paper are as follows:

- We propose an algorithm for exploiting existing extractive gold standard (EGS) summaries of full threads to automatically generate oracular "silver standard" PT summaries of arbitrary length, as discussed in section 3. Further, we are releasing these silver standard summaries for the dataset used in this work.

- For PT summary generation, we propose an unsupervised method extending previous work on full-thread summarization that considers not only the input thread, but also background knowledge synthesized from a large number of other email threads. In particular, we developed a summarization method based on Bayesian Surprise (Louis, 2014) which takes into account conversational features of the partial thread, as discussed in section 4. We then evaluate the system-generated summaries using our silver standards with ROUGE.

- Using our silver standard with ROUGE, we carry out experiments to compare the summaries generated by Bayesian-based methods with summarization techniques that do not take into account background information.

2 Related Work

To generate PT summaries we propose an unsupervised extractive approach. Although to the best of our knowledge no one has studied PT summarization directly, there has been extensive work done in extractive summarization in general, as well as work done on email summarization specifically. Supervised methods have been proposed which turn the extractive summarization task into a binary classification problem where sentences are labeled in/out using standard machine learning classifiers (Rambow et al., 2004; Murray and Carenini, 2008). Variations of this approach include adding sentence compression and using integer linear programming to evaluate candidate summaries and select the best ones (Berg-Kirkpatrick et al., 2011). Sentence classification assumes sentences are independent from one another; and so to capture dependencies between sentences, the extractive summarization problem has also been recast as a sequence labeling problem using hidden Markov models and conditional random fields (Fung et al., 2003; Jin et al., 2012; Oya and Carenini, 2014).

The weakness of supervised approaches is the reliance on human-annotated labeled data, which is often expensive and difficult to acquire due to privacy concerns. Our extractive approach, therefore, will focus on unsupervised extractive techniques which do not require labeled data. Another benefit of unsupervised methods is that they can serve as features for supervised methods, meaning improvements in unsupervised techniques can directly benefit supervised systems.

Many unsupervised extractive summarization methods have been proposed for generic docu-

ments, as well as for conversations. Some make use of textual features such as lexical chains, cue words ("In conclusion", "To summarize", etc.) or conversation structure to select the most informative sentences (Barzilay and Elhadad, 1999; Hatori et al., 2011; Carenini et al., 2008). Others make use of more advanced methods including topic modeling, latent semantic analysis or rhetorical parsing (Nagwani, 2015; Kireyev, 2008; Hirao et al., 2013). Our algorithm for generating silver standard summaries of partial threads incorporates a topic modeling framework that, in turn, makes use of lexical chains and conversational structure.

There is also a large class of methods which build graphs with textual units (words, sentences, paragraphs, etc) as vertices and use similarity measures between the text units to form the edge weights. Once a full graph is created, an extractive summary is generated by using a centrality measure to select central nodes from a cluster and concatenating them to form a summary. Two popular systems are LexRank and TextRank, which both use a variant of the PageRank algorithm (Erkan and Radev, 2004; Mihalcea and Tarau, 2004; Mihalcea and Radev, 2011). Graph methods are popular because of their simplicity and ease of implementation, and their performance has been shown to be competitive with other methods. Our silver standard algorithm and baseline summarizer both incorporate graph-based sentence scoring.

No matter how the information content (or the query relevance) of a sentence is computed, sentences should be included in the final summary not only if they are informative but also if they convey new information with respect to sentences already in the summary. One popular method known as *Maximum Marginal Relevance* (MMR) builds a summary with a scoring function that trades off between the "relevance" and "information-novelty" of a sentence, and builds a summary by selecting sentences which maximize relevance and minimize redundancy with previously selected sentences (Carbonell and Goldstein, 1998). While our silver standard generation system uses vanilla MMR, the Bayesian Surprise-based summarizers described in section 4 have a built-in means of handling information redundancy.

There has also been work on the task of unsupervised email summarization specifically. Carenini et al. (2008) proposed the use of "fragment quotation graphs" (FQGs) to summarize asynchronous conversations. FQGs use the fact that a given email often contains quoted material from previous emails. These quotations, or "fragments", can then be used to create fine-grained representations of the underlying structure of a given email thread, allowing a set of particularly informative *clue words* to be identified. In this paper, we also exploit FQGs in our silver standard generation system, and we use a summarizer based on clue words as a baseline in our evaluations.

Furthermore, a key limitation of (Carenini et al., 2008), common to other approaches to full-thread summarization, is to consider only the input thread in the summarization process; in contrast, a user's email history (or that of the user's organization) can provide valuable background knowledge. The summarizer we propose in this paper addresses this limitation by taking into account background knowledge synthesized from a large number of other email threads, which we argue is especially beneficial to PT summarization as the PT can be rather short and consequently unable to provide much ground for sentence selection.

3 Generating Silver Standard Summaries for Partial Email Threads

In order to automatically evaluate PT summaries (e.g., with ROUGE), human-generated EGS summaries are needed for comparison. However, because producing such EGS summaries is a time-consuming and often difficult task, all publicly available email corpora we are aware of only provide human-annotated EGS summaries for each email thread as a whole (Loza et al., 2014; Ulrich et al., 2008). Given a partial thread PT and a gold standard summary EGS of the corresponding full thread, an intuitive solution might be to simply use $EGS \cap PT$ as the silver standard. In this section we discuss potential problems with that approach as well as our solution.

3.1 Distribution of Summary Sentences

The distribution of EGS sentences across emails in a thread cannot be assumed to be uniform in all (or even most) cases; indeed, this is not the case in the dataset used in this work (a collection of 62 email threads, described further in section 5). As shown in Figure 1, while many threads in the dataset have highly ranked EGS sentences in the first part of the conversation, others have important EGS sentences in the middle or even at the end of the con-

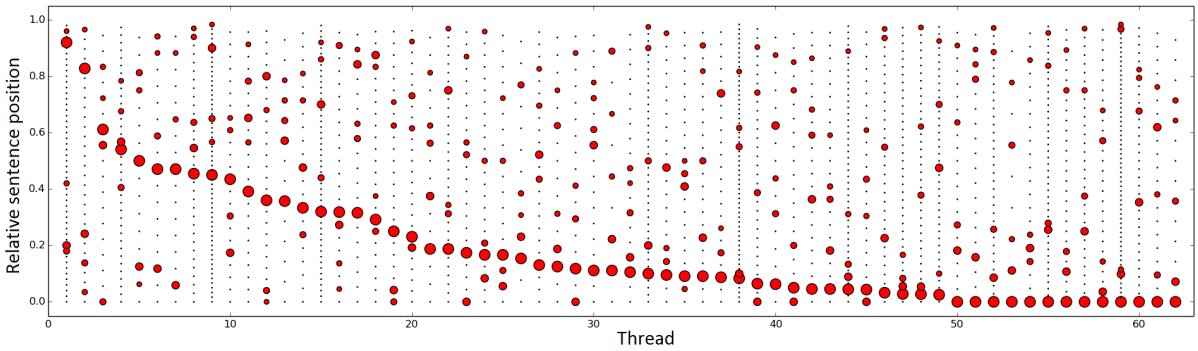

Figure 1: Distribution of EGS sentences in full threads. Each vertical column of dots represents a thread, with each dot representing a sentence at its relative position within the thread (beginning at 0, ending at 1). Non-EGS sentences are black dots, while EGS sentences are red circles; and larger circles indicate that a human annotator considered those sentences more important. The threads are sorted in descending order of the relative position of the highest-ranked sentences.

versation. Variations in EGS sentence distribution become a concern when generating silver standard PT summaries. In some cases, there may not be enough EGS sentences in a given PT to form a silver standard summary; in extreme cases, the PT may have no EGS sentences at all. In other cases, there may be too many EGS sentences in a PT to fit into the silver standard; and not all datasets rank EGS sentences by importance as part of the annotation. In other words, unless exactly the desired number of EGS sentences are present in each PT, some sentence selection is necessary; and this issue is exacerbated when generating silver standard summaries of arbitrary length. Our silver standard generation algorithm handles all these possibilities as described in the next section.

3.2 The Silver Standard Algorithm

We propose an oracular algorithm for generating silver standard extractive reference summaries of arbitrary length for partial threads; in other words, it references the existing gold standard for the full thread to generate silver standard summaries for the partial threads. Our silver standard system incorporates graph-based sentence scoring, which has been used extensively for summarization as discussed in section 2. Both the graph-based aspect of the algorithm and its redundancy minimization mechanism rely on word embeddings trained using a large email corpus.

Our silver standard system also makes use of topic modeling. We expect the discussions in email threads to be topically coherent (though, for both individual emails and threads, multiple topics may be covered). The topical coherence of a sentence with both the PT and the gold standard are thus related to that sentence's importance in the discussion in the context of the PT as well as the thread as a whole. The topic modeling system we used exploits conversational structure.

The pseudocode for silver standard generation is given in Algorithm 1. The first step (lines 6-14) is to seed the silver standard with EGS sentences in the PT. If there are more EGS sentences than the desired silver standard length, then a sentence selection method (using human-annotated rankings if available) is applied. This first step is oracular because it directly references the gold standard. If there are fewer EGS sentences in the PT than desired for the silver standard, then the algorithm proceeds to the second step (lines 15-18), where the sentence selection method is applied to the rest of the PT sentences.

For this work, we have chosen an intuitive sentence selection method that can be used in both steps as needed. To maximize sentence importance while minimizing redundancy, the selection method uses maximal marginal relevance (MMR) (Carbonell and Goldstein, 1998). For a given candidate sentence s for inclusion in a summary S, its MMR score is

$$MMR(s) = \lambda I(s) - (1 - \lambda) Sim(s, S) \quad (1)$$

where $I(s)$ is an *importance function*, and *Sim(s,S)* is a similarity function comparing s to the sentences of S. For this work we set λ to 0.5.

The importance function used here incorporates graph centrality and topic segmentation. We first define $PR_{PT}(s)$ as the PageRank score of s in the

Algorithm 1: Silver Summary Generation

Result: SLV_m

- Let $EGS = \{egs_1 ... egs_j ... egs_k\}$ be set of gold standard sentences;
- Let $PT_m = \{pt_1 ... pt_i ... pt_n\}$ be set of sentences in the partial thread up to email m;
- Let $EGS_m^{PT} = \{egs_1^{PT} ... egs_i^{PT} ... egs_n^{PT}\} = (EGS \cap PT_m)$;
- Let $SLV_m = \emptyset$ be silver summary of PT_m;
- Let len be desired length of silver summary;

while $|SLV_m| <$min$(len, |EGS|)$ **do**
  if EGS has annotated sentence ranking **then**
    - score each egs_i^{PT} using ranking
  end
  else
    - score each egs_i^{PT} using $scoring\_function(egs_i^{PT})$;
  end
  - add highest scoring $egs_i^{PT} \notin SLV_m$ to SLV_m
end
while $|SLV_m| < len$ **do**
  - score each pt_i using $scoring\_function(pt_i)$;
  - Add highest scoring $pt_i \notin SLV_m$ to SLV_m
end
return SLV_m

fully-connected graph whose vertices are the sentences of the partial thread. We choose PageRank over LexRank in order to incorporate topic modeling designed for conversational data. For each sentence in PT, a vector representation is obtained by averaging 100-dimensional Word2Vec embeddings of its words (Goldberg and Levy, 2014). The edge weights are then set to the cosine similarity of the vector representations of the relevant sentences. The Word2Vec model was trained on the entire Enron email corpus of ∼500K emails.

We then define $T(s)$ as the topic of sentence s; in this work, we apply a topic segmentation method that uses fragment quotation graphs to represent conversational structure and that has been shown to work well on asynchronous conversations (Joty et al., 2013). We then define $Prom_{PT}(T(s))$, or the *prominence* of $T(s)$ within the partial thread, as the fraction of PT sentences that have that topic; so if a PT containing five sentences has a total of three whose topic is $T(s)$, then $Prom_{PT}(T(s))$ is 0.6. Similarly, $Prom_{EGS}(T(s))$ is the prominence of $T(s)$ within the gold standard summary. Together, the two prominence scores form the overall topic prominence score of s:

$$Prom(T(s)) = \frac{1}{2}\Big(Prom_{PT}(T(s)) + Prom_{EGS}(T(s))\Big) \quad (2)$$

Note that $Prom_{EGS}(T(s))$ is a second oracular component of the silver standard algorithm, since it references the importance of a topic in the context of the entire thread as represented by the EGS. By increasing the likelihood of choosing sentences from the same topics as the EGS, this ensures the silver standard is oracular even in cases where there are no EGS sentences in the PT of interest.

Putting graph centrality and topic prominence together, we have:

$$I(s) = \frac{1}{2}\Big(PR_{PT}(s) + Prom(T(s))\Big) \quad (3)$$

It is worth noting that the PageRank score takes values in [0,1], as do both of the prominence scores. The weights in equations 2 and 3 are set to match the simplifying assumption that the centrality of a sentence in its PT is as important as the overall prominence of its topic, and that the prominence of a topic within a PT is as important as its prominence within the larger context of the

full thread. Taken together, the importance function takes values in [0,1], which is appropriate for MMR.

The similarity function $Sim(s, S)$ in equation 1 is the maximum cosine similarity of the candidate sentence s and the sentences of the in-progress summary S, using the aggregated Word2Vec representations described for the PageRank score.

4 Generating Partial Thread Summaries

While previous work on unsupervised full-thread summarization essentially takes as input only the thread to be summarized, Louis (2014) has shown that background knowledge can be effectively taken into account in the summarization process by applying the idea of Bayesian Surprise.

The Bayesian Surprise method is based on the intuition that, given a collection of background knowledge (such as the email history of a user or organization), the most "surprising" new information is the most significant for inclusion in a summary.

Presumably, while background knowledge should be useful for summarization in general as an additional source of information from which to infer salience, it should be especially useful for PT summarization, since the partial threads can be rather short, and thus there is relatively little information available to a given summarizer. For this reason, our PT summarization method is based on Bayesian Surprise, but it extends the existing technique to consider conversational features and incorporates a less harsh redundancy management mechanism.

4.1 Bayesian Surprise

Let H be some hypothesis about a background corpus that is represented by a multinomial distribution over word unigrams. The prior probability of H is a Dirichlet distribution:

$$P(H) = Dir(\alpha_1, ...\alpha_V) \tag{4}$$

where α_i is the count of word i in the background corpus, and V is the size of the background corpus vocabulary.

Suppose word w_i appears c_i times in the PT being summarized. We can then obtain the posterior

$$P(H|w_i) = Dir(\alpha_1, ..., \alpha_i + c_i, ...\alpha_V) \tag{5}$$

The Bayesian Surprise score for w_i due to the PT is then the KL divergence between $P(H|w_i)$ and $P(H)$. Then, to obtain the Bayesian Surprise score of a sentence, one simply aggregates the scores of its words; and the sentence with the highest score is added to the summary. In order to minimize redundancy during summarization in the original proposal (Louis, 2014), once a sentence is added to a summary, the Bayesian Surprise scores of its words are set to zero. The process is repeated until the desired summary length has been reached.

4.2 Conversational Features

As discussed in section 2, conversational features have proved useful in summarizing asynchronous conversations such as email threads. We have extended the Bayesian Surprise method to include a number of these conversational features as additional concentration parameters in the Dirichlet distributions. In order to maintain consistency with the original Bayesian Surprise method, we limit our extensions to features that can be expressed as counts of word w_i; specifically, we use the number of times w_i was used:

- by the creator of the thread (whether in the initial email or afterwards)
- by the dominant participant in the thread (who may or may not be the thread creator)
- in emails where it also appears in the email subject line
- as a clue word

The prior for the extended Bayesian Surprise method then becomes

$$P(H) = Dir(\alpha_{1..V}, \beta_{1..V}, \gamma_{1..V}, \delta_{1..V}, \epsilon_{1..V}) \tag{6}$$

where $\alpha_{1..V}$ are the original concentration parameters, and $\beta, \gamma, \delta, \epsilon$ are the corresponding feature counts.

4.3 Surprise Decay

As discussed in section 4.1, once a sentence containing a word is added to the summary, the Bayesian Surprise score of that word is set to zero in order to minimize redundancy. While this accomplishes that goal, it may impact the measured importance of words in the larger context of the PT too harshly. In order to mitigate this effect, we propose an alternative we call *surprise decay*, where

each time a sentence is added to the summary, the Bayesian Surprise scores of its words are multiplied by some *decay factor* < 1. Intuitively, this corresponds to making these words "less surprising," rather than removing the surprise entirely; this allows salient words to continue to contribute to the overall surprise of sentences in a limited way as the summary is generated. The simplest decay factor would be a constant $df \in [0, 1)$, resulting in exponential decay of a given word's Bayesian Surprise score.

5 Dataset

We used the "corporate thread" subset of the publicly available annotated email dataset produced by Loza et. al., which was derived from the Enron email dataset (Loza et al., 2014). The data consists of 62 email threads (from which 282 PTs can be extracted) containing a total of 354 emails and 1654 sentences. Each thread is manually annotated with abstractive and extractive summaries, as well as five ranked keyphrases. This work focuses on extractive summarization, so only those annotations were used. The keyphrases were not used here, because it is not expected that most gold standard annotations will include keyphrases.

Each thread was annotated by two annotators, so for each thread we have two sets of extractive sentences. The annotators were asked to select up to five sentences *"that contained the most important information in the email, and also rank the sentences in reverse order of their importance"*.

To serve as a background corpus that could be used for both Bayesian Surprise methods, we used a publicly available collection of threads extracted from the Enron corpus (Jamison and Gurevych, 2013), of which threads ∼43k had the metadata required (sender, recipient(s) and subject line in all emails) in order to extract the desired conversational features.

6 Experimental Setup and Results

We generated a number of summaries for each full thread as well as for its corresponding PTs. First, we generated summaries using both the original and our extended Bayesian Surprise methods (**BS** and **BSE**) discussed in sections 4.1 and 4.2. We then generated additional summaries for each method using the exponential surprise decay (**-d**) discussed in section 4.3 with $df = 0.5$.

In addition, we generated summaries using a method (**CWS**) that scores sentences based on the number of clue words they contain (Carenini et al., 2008). This method was shown to perform well in email summarization, and we use it here as a baseline.

6.1 Evaluation over Full Threads

Initially, we evaluated the system summaries over the full threads against the human-annotated EGS. The evaluation was carried out using ROUGE-1 F-scores. In the ROUGE evaluation, stemming was performed, but stopwords were not removed, consistent with previous evaluations of summarization based on Bayesian Surprise (Louis, 2014). The system summaries were truncated to the length (in words) of the corresponding EGS. As a baseline we used a PageRank-based summarizer (**PR-MMR**) that scores sentences using the same sentence graphs as the silver standard algorithm and employs MMR to minimize redundancy. The results for this evaluation over full threads are given in Table 1.

Method	Full threads
BS	0.573
BS-d	0.582
BSE	0.566
BSE-d	0.573
CWS	**0.598**
PR-MMR	0.509

Table 1: ROUGE-1 mean F-scores for full threads as compared to gold standard summaries.

The results of this experiment over full threads suggest that the Bayesian Surprise-based methods perform comparably to the clue words-based summarizer, and that they all significantly outperform the PR-MMR baseline ($p<0.005$)[2]. In addition, there appears to be some benefit to the more gradual redundancy handling provided by surprise decay, though the differences in these cases do not appear to be significant.

6.2 Evaluation over Partial Threads

To evaluate the summarizers over partial threads, we generated two silver standard summaries (one for each annotator) per PT using the algorithm in section 3. The silver standard and system summaries for each PT were truncated to a fraction of

[2]Significance for all reported results was verified using ANOVA followed by paired t-tests (with Bonferroni corrections as needed).

the PT length (in words). Since the silver standard algorithm generates summaries of arbitrary length, we evaluated the summarizers at both 20% and 30% of the PT length.

The hypothesis behind our use of Bayesian Surprise-based methods is that they should work particularly well for PT summarization, because PTs can be rather short, and the identification of the most informative sentences would benefit from examining a larger informational context. To test this hypothesis we sorted the 282 PTs being summarized by length and binned them into quartiles (see Table 2). Since BSE-d is the Bayesian Surprise-based method incorporating all of our extensions, we focus our statistical analysis on comparing it to CWS. The results of this evaluation are given in Table 3.

	min	25%	median	75%	max
Length	22	104	197	329	1236

Table 2: Length (in words) of the partial threads in the dataset used to define the quartile bins.

We observe a number of trends in Table 3 from the experiments over PTs; however, only some cases exhibit at least marginal significance. This may be due in part to limited sample size; and so we argue that further work in applying background knowledge to PT summarization over larger datasets is warranted.

Over the shorter PTs (i.e. first and second quantiles) and at both summary lengths, we observe a trend favoring our hypothesis, namely that Bayesian Surprise-based methods seem to perform better than CWS; for example, the performance improvement of BSE-d over CWS is at least marginally significant ($p<0.1$) for the second quartile at both summary lengths. Conversely, for the longest PTs (i.e., the fourth quantile), we see that the effectiveness of clue words is more fully realized, allowing CWS to outperform the Bayesian Surprise-based summarizers. While this difference is significant ($p<0.05$) for summaries of 30% PT length, it is not significant at 20% PT length; this suggests that Bayesian Surprise-based summarizers may be more robust against changes in PT summary length than CWS.

Surprisingly, the conversational features used to extend the Bayesian Surprise method have not improved summarizer performance. It may be that treating these features as equivalent to word counts is inappropriate for this task, in which case some other means of extracting these features as background knowledge should be devised. Alternatively, the inclusion of additional features, such as the number of times a word is used in the first sentence of each email in the thread, may improve the performance of the extended Bayesian Surprise summarizer.

As with the full threads, the inclusion of surprise decay seems to provide some benefit, though it appears to hamper the summarizers for the shortest PTs; this trend can be seen at 30% PT length, where BS-d outperforms BS in all quartiles except the first. This suggests that applying surprise decay factors derived from PT length and desired summary length may improve overall performance; we leave this endeavor for future work.

7 Conclusions and Future Work

In this work, we have defined and motivated the partial thread summarization problem. We have proposed an algorithm that uses gold standard summaries of complete threads in order to build oracular silver standard extractive summaries of arbitrary length for partial email threads. We have also applied an intuitive unsupervised summarization method to PT summarization, extended it with conversational features, and modified the mechanism by which it handles redundancy. Although in our experiments we did not find consistently significant improvements using Bayesian Surprise-based methods on partial threads, we argue that in light of the observed trends, the potential benefit of background knowledge to PT summarization (and email summarization in general) should be further investigated with larger datasets.

There are multiple directions of future work. While an obvious direction is the continued development of extractive PT summarization algorithms (eg. by applying recent summarization techniques such as ILP (Murray et al., 2010) or neural network-based summarizers (Cao et al., 2015)), another is the abstractive summarization of partial threads. Yet another is the application of the silver standard algorithm to other asynchronous conversations, such as discussion forums, as well as other domains where some human annotation is available but reference summaries for different portions of the source document(s) are desired.

Future work may also include finding additional

30% PT length	BS	BS-d	BSE	BSE-d	CWS	*p*
Q1	**0.666**	0.643	0.632	0.622	0.582	*0.310*
Q2	0.558	**0.576**	0.560	0.571	0.503	*0.041*
Q3	0.552	0.565	0.540	0.535	**0.568**	*0.088*
Q4	0.504	0.516	0.510	0.510	**0.548**	*0.011*
all PTs	0.570	**0.575**	0.560	0.559	0.550	*0.519*
20% PT length	BS	BS-d	BSE	BSE-d	CWS	*p*
Q1	**0.600**	0.577	0.558	0.557	0.512	*0.402*
Q2	0.504	**0.513**	0.495	0.494	0.424	*0.078*
Q3	**0.476**	0.470	0.469	0.469	0.467	*0.933*
Q4	0.435	0.441	0.439	0.448	**0.452**	*0.808*
all PTs	**0.504**	0.500	0.490	0.493	0.464	*0.114*

Table 3: ROUGE-1 mean F-scores over partial threads (binned into quartiles by length in words) as compared to silver standard summaries. Values are given for both summary lengths (20% and 30% of PT length). Bolded ROUGE scores are the highest for their quartile and summary length category. P-values are given for the comparisons between BSE-d and CWS; underlined p-values indicate at least marginal significance ($p<0.1$).

ways to incorporate background knowledge into email summarization. For example, Bayesian Surprise scores may be used in tandem with other features to develop summarizers that are more robust against changes in document length.

An advantage to the study of PT summarization is that it may reveal whether current summarization techniques perform differently on in-progress threads than on complete, archived ones. For example, if a summarizer uses features that may depend on the entire email thread (eg. the relative positions of sentences in the thread, completed dialog acts, etc.), then those features may have a different significance when applied to PTs than they do for complete threads. Similarly, PT summaries may give insights into the development of email threads over time. For example, the summaries generated for an earlier PT may have features that are useful in summarizing a later PT or in predicting aspects of a thread's future development. To further the study of PT summarization, another direction of future work is a thorough categorization of the differences between full and partial threads, as well as differences between PTs at different stages of development. Such differences may be found, for example, in lexical and topic diversity, as well as dialog act initiation and/or completion.

Acknowledgments

This research was funded in part under a grant from the Yahoo Faculty Research and Engagement Program (FREP).

References

Regina Barzilay and Michael Elhadad. 1999. Using lexical chains for text summarization. Advances in automatic text summarization pages 111–121.

Taylor Berg-Kirkpatrick, Dan Gillick, and Dan Klein. 2011. Jointly learning to extract and compress. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1. Association for Computational Linguistics, pages 481–490.

Ziqiang Cao, Furu Wei, Li Dong, Sujian Li, and Ming Zhou. 2015. Ranking with recursive neural networks and its application to multi-document summarization. In AAAI. pages 2153–2159.

Jaime Carbonell and Jade Goldstein. 1998. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval. ACM, pages 335–336.

Giuseppe Carenini, Raymond T Ng, and Xiaodong Zhou. 2008. Summarizing emails with conversational cohesion and subjectivity. In ACL. volume 8, pages 353–361.

Hoa Trang Dang and Karolina Owczarzak. 2008. Overview of the tac 2008 update summarization task. In Proceedings of text analysis conference. pages 1–16.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. Journal of Artificial Intelligence Research 22:457–479.

Pascale Fung, Grace Ngai, and Chi-Shun Cheung. 2003. Combining optimal clustering and

hidden markov models for extractive summarization. In Proceedings of the ACL 2003 workshop on Multilingual summarization and question answering-Volume 12. Association for Computational Linguistics, pages 21–28.

Yoav Goldberg and Omer Levy. 2014. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. arXiv preprint arXiv:1402.3722 .

Jun Hatori, Akiko Murakami, and Junichi Tsujii. 2011. Multi-topical discussion summarization using structured lexical chains and cue words. In International Conference on Intelligent Text Processing and Computational Linguistics. Springer, pages 313–327.

Tsutomu Hirao, Yasuhisa Yoshida, Masaaki Nishino, Norihito Yasuda, and Masaaki Nagata. 2013. Single-document summarization as a tree knapsack problem. In EMNLP. volume 13, pages 1515–1520.

Emily Jamison and Iryna Gurevych. 2013. Headerless, quoteless, but not hopeless? using pairwise email classification to disentangle email threads. In RANLP. pages 327–335.

Wei Jin, Shafiq Joty, Giuseppe Carenini, and Raymond Ng. 2012. Detecting informative blog comments using tree structured conditional random fields. NW-NLP, Microsoft Research, Redmond.

Shafiq Joty, Giuseppe Carenini, and Raymond T Ng. 2013. Topic segmentation and labeling in asynchronous conversations. Journal of Artificial Intelligence Research 47:521–573.

Kirill Kireyev. 2008. Using latent semantic analysis for extractive summarization. In Proceedings of text analysis conference. volume 2008.

Annie P Louis. 2014. A bayesian method to incorporate background knowledge during automatic text summarization. Association for Computational Linguistics.

Vanessa Loza, Shibamouli Lahiri, Rada Mihalcea, and Po-Hsiang Lai. 2014. Building a dataset for summarization and keyword extraction from emails. In LREC. pages 2441–2446.

Rada Mihalcea and Dragomir Radev. 2011. Graph-based natural language processing and information retrieval. Cambridge University Press.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into texts. Association for Computational Linguistics.

Gabriel Murray and Giuseppe Carenini. 2008. Summarizing spoken and written conversations. In Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, pages 773–782.

Gabriel Murray, Giuseppe Carenini, and Raymond Ng. 2010. Generating and validating abstracts of meeting conversations: a user study. In Proceedings of the 6th International Natural Language Generation Conference. Association for Computational Linguistics, pages 105–113.

NK Nagwani. 2015. Summarizing large text collection using topic modeling and clustering based on mapreduce framework. Journal of Big Data 2(1):1.

Tatsuro Oya and Giuseppe Carenini. 2014. Extractive summarization and dialogue act modeling on email threads: An integrated probabilistic approach. In 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue. page 133.

Owen Rambow, Lokesh Shrestha, John Chen, and Chirsty Lauridsen. 2004. Summarizing email threads. In Proceedings of HLT-NAACL 2004: Short Papers. Association for Computational Linguistics, pages 105–108.

Jan Ulrich, Gabriel Murray, and Giuseppe Carenini. 2008. A publicly available annotated corpus for supervised email summarization. In Proc. of aaai email-2008 workshop, chicago, usa.