EMAILSUM: Abstractive Email Thread Summarization

Shiyue Zhang[♠] **Asli Celikyilmaz**[♡] **Jianfeng Gao**[♣] **Mohit Bansal**[♠]
[♠]UNC Chapel Hill [♡]Facebook AI Research [♣]Microsoft Research
{shiyue, mbansal}@cs.unc.edu
aslic@fb.com jfgao@microsoft.com

Abstract

Recent years have brought about an interest in the challenging task of summarizing conversation threads (meetings, online discussions, etc.). Such summaries help analysis of the long text to quickly catch up with the decisions made and thus improve our work or communication efficiency. To spur research in thread summarization, we have developed an abstractive **Email** Thread **Sum**marization (EMAILSUM) dataset, which contains human-annotated short (<30 words) and long (<100 words) summaries of 2,549 email threads (each containing 3 to 10 emails) over a wide variety of topics. We perform a comprehensive empirical study to explore different summarization techniques (including extractive and abstractive methods, single-document and hierarchical models, as well as transfer and semi-supervised learning) and conduct human evaluations on both short and long summary generation tasks. Our results reveal the key challenges of current abstractive summarization models in this task, such as understanding the sender's intent and identifying the roles of sender and receiver. Furthermore, we find that widely used automatic evaluation metrics (ROUGE, BERTScore) are weakly correlated with human judgments on this email thread summarization task. Hence, we emphasize the importance of human evaluation and the development of better metrics by the community.[1]

Email Thread:
Subject: lunch this week
Susan: All, Regarding our lunch this week to celebrate the one year anniversaries for Michelle & David, and Mark's birthday, I have a request to make it Wednesday instead of Tuesday. Does anyone have an objection to this? Susan
David: I have another lunch engagement Wed, but I will skip it if everyone else wants to move our lunch. David
Tamra: Susan, Wednesday works out better for me as well. I have a doctor's appointment tomorrow during lunch. Tamra

Short Summary:
Susan emails everyone about an anniversary and offers to change the date. David says he is busy but is willing to go with the majority. Tamra agrees with Susan's date.

Long Summary:
Susan emails everyone about a lunch to celebrate a one year anniversary as well as Mark's birthday. She says she would change the date to a different day. David says he is busy that day with his own appointment but is willing to go with the majority and cancel that appointment to make this one. Tamra agrees with Susan's date as she is busy Tuesday with an appointment.

Table 1: An email thread and human-written short and long summaries from our EMAILSUM Dataset.

1 Introduction

As one of the major natural language generation tasks, automatic summarization has been studied for decades. Most research efforts were focused on single-document summarization tasks, e.g., news document summarization (Hermann et al., 2015; Narayan et al., 2018). However, living in an information era, we are facing with diverse content in different structures. The summarization need is varied along with different application scenarios. Recently, there is an increasing research interest in diverse summarization tasks (Gao et al., 2020), e.g., timeline (Allan et al., 2001), query-based (Li and Li, 2014), multi-modal (Zhu et al., 2018), meeting (Carletta et al., 2006), dialogue or discussion thread (Misra et al., 2015; Gliwa et al., 2019; Rameshkumar and Bailey, 2020), etc. Following the branch of dialogue or thread summarization, we introduce a new abstractive **Email** Thread **Sum**marization (EMAILSUM) dataset.

Email threads are widely used at work. An email thread is a special type of dialogue that usually has a specific structure (sender, receiver, greeting line, main body, and the signature), contains technical information, and involves multiple speakers. Unlike a conversational dialog turn, an email in a

[1] Our code and summary data have been made available at: https://github.com/ZhangShiyue/EmailSum

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 6895–6909
August 1–6, 2021. ©2021 Association for Computational Linguistics

thread is much longer with longer sentences, multiple action items or requests, and stylistically similar to written text. Studies have shown that on average a worker sends/receives 122 business emails (Radicati, 2015) and spends more than 3 hours on those emails (Adobe, 2019) per day. One possible reason is that sometimes people have to read through the entire conversation before replying to the latest email. This happens when you forget the main points of previous discussions or you are newly included in a discussion thread. Therefore, automatically summarizing email threads can improve our work efficiency and provides practical benefits. Email Thread Summarization is not a new task. Carenini et al. (2007) collected extractive summaries of 39 email threads from Enron email corpus (Klimt and Yang, 2004) and proposed to use a fragment quotation graph and clue words to conduct summarization. Ulrich et al. (2008) collected both extractive and abstractive summaries of 40 threads from W3C email corpus (Craswell et al., 2006) plus speech acts, meta sentences, etc. However, this task has been much less studied compared to other summarization tasks, partially due to the lack of large labeled email thread datasets.

In this paper, we collect human-written short ($<$ 30 words) and long ($<$ 100 words) abstractive summaries of 2,549 email threads constructed from Avocado Research Email Collection (Oard et al., 2015), which is 64× the size of previously labeled email thread datasets (Carenini et al., 2007; Craswell et al., 2006). We limit each thread to a minimum of 3 and a maximum of 10 emails, an example is given in Table 1. We also extract 8,594 unlabeled email threads from both Avocado and W3C to facilitate semi-supervised learning.[2] See Section 2 for details of data collection.

Next, we present comprehensive baselines from different learning paradigms as a benchmark for our new email summarization dataset. Specifically, we explore different summarization techniques, including extractive and abstractive summarization methods, single-document and hierarchical models, transfer learning, and semi-supervised learning for both short and long summary generation. Experiments demonstrate that utilizing pretrained language model (e.g., T5 (Raffel et al., 2020)) is critical due to the small size of our data; taking the email thread as a single document sets up a good baseline; transferring from news or dialogue datasets barely improve the performance; using hierarchical encoders only marginally improves it; while semi-supervised learning by using unlabelled email threads significantly ($p < 0.01$) improves ROUGE (Lin, 2004) scores in some cases.

Lastly, to better understand how well the email thread summarization models perform and investigate the correlation between automatic metrics and human judgment, we ask humans to rate the "salience" (how well the model summarizes salient points) and "faithfulness" (how well the model stays true to the email thread) of model-generated summaries, as well as to perform a pairwise comparison between our best and base models. We find that even though semi-supervised learning improves ROUGE scores, human judges still favor the summary generated by the baseline model (T5$_{base}$). Two frequent errors made by the model are (1) *failing to understand the sender's intent* and (2) *failing to identify the roles of the sender and receiver*. Relatedly, human correlation analysis reveals that automatic metrics (ROUGE (Lin, 2004), BERTScore (Zhang et al., 2019)) are poorly correlated with human judgment, which stresses the importance of human evaluation in this task and the requirement for better metrics to be proposed. Overall, in this work, we propose the new EMAILSUM dataset that provides a larger resource for studying the email thread summarization task. We conduct a comprehensive empirical model study and human evaluation analysis, which will serve as an important starting point for future studies.

2 EMAILSUM Dataset

To collect email thread summarization data, we first need to obtain unlabelled email threads. We resort to existing email collections: Enron (Klimt and Yang, 2004), W3C (Craswell et al., 2006), and Avocado (Oard et al., 2015). However, none of them provides explicit thread structure. Therefore, in this section, we will introduce our email thread preprocessing and summary collection procedures.

2.1 Email Thread Preprocessing

We extract email threads from the flat email collections in the following steps: (1) we give every email a "normalized subject" by removing the reply or forward tags (e.g., "Re:", "Fwd:", etc.) from its original subject; (2) we group emails by the normalized subjects and sort emails in the same group (i.e.,

[2] We apply strict criteria for thread extraction (see Section 2). More threads can be extracted by relaxing those constraints.

Domain	News		Dialogue		Email Thread		
Dataset	CNN/DM	XSum	SAMSum	CRD3	BC3	EMAILSUM$_{short}$	EMAILSUM$_{long}$
# of documents	312,085	226,677	16,369	32,720	40	2,549	2,549
Avg. document length	786.4	409.5	124.1	615.8	550.4	233.2	233.2
# of turns per doc.	-	-	10.2	27.5	6.4	4.5	4.5
Avg. turn length	-	-	11.1	19.4	85.3	50.3	50.3
Avg. summary length	55.2	23.2	23.4	58.3	134.3	27.1	68.5
Ext-Oracle-R1	58.2	23.8	45.3	50.4	36.5	39.0	46.0

Table 2: The statistics of different summarization datasets. Ext-Oracle-R1s are the ROUGE-1 scores of the oracle extractive method, which shows the abstractiveness of the summary (the lower the more abstractive).

thread) by timestamp; (3) we de-duplicate emails in every thread by sender's email plus timestamp; (4) we traverse emails in every thread in temporal order and cut off the thread when none of the senders plus receivers of the current email appears in previous emails; (5) we filter out threads that only contain single repeated content.

To obtain a cleaner dataset, we remove threads that do not comply with the following constraints: (1) 3 $\leq$ the number of emails $\leq$ 10; (2) 5 $<$ the number of words in each email $<$ 200; (3) 30 $<$ the total number of words $<$ 1000; (4) does not contain non-English (e.g., German) tokens; (5) does not contain reply or forward tags in the subject of the first email.

Emails often contain personal information such as full name, email/physical address, phone number, etc. To protect privacy, we anonymize all email threads before annotation: (1) only keep first names; (2) remove threads that have "password", "pwd", "confidential", etc.; (3) replace email address, physical address, phone number, URL, IP address, local path, and other sensitive numbers with USERNAME@DOMAIN.COM, ADDRESS, PHONENUMBER, HTTP://LINK, IPADDRESS, PATH, and NUMBER, respectively.

We conduct an extensive manual quality scan to make sure that the extracted threads are truly threads (instead of random emails grouped) and properly anonymized. Finally, we obtain 8,116 threads from Avocado and 3,478 threads from W3C.[3] We randomly sample 3K Avocado threads for summary annotation, and the remaining threads are used as unlabelled data.

2.2 Thread Summary Collection

We collect summary annotations on Amazon Mechanical Turk. Since summarizing text is not an easy task, to get acceptable English summaries we use several quality control strategies: (1) We select annotators that are located in the US, have an approval rate greater than 97%, and have at least 10,000 approved HITs; (2) During annotation, we periodically sample summaries, manually check their quality, and reject or block poor-quality annotators; (3) After annotation, we randomly sample 2 examples per annotator and manually categorize annotators into "good", "fair", and "bad" groups, then filter examples written by bad annotators.

Email threads oftentimes contain technical information, we instruct annotators not to get stuck on technical details, instead, focus on the major concerns, decisions, and consensus. We collect both short ($<$ 30 words) and long ($<$ 100 words) abstractive summaries per thread. For the short summary, we instruct annotators to write a *concise description of what the thread is mainly talking about*; while for the long summary, we instruct them to write a *a narrative of what happens*. We are intent to provide summaries with two different levels of abstractiveness, length, and concreteness. We show annotators an example written by an expert (a CS graduate student). More summary collection details can be found in Appendix A.

2.3 Final Dataset Description

The summary collection and filtering process yield 2,549 email threads each with a long and a short summary. We randomly sample 500 examples from the "good" annotator group as our testing set and split the remaining examples into training (1,800 threads) and development (249 threads) sets. Table 2 shows the statistics of EMAILSUM.[4]. For ease of benchmarking, we also include statistics on other

[3] We find that the extracted threads from Enron are usually short (fewer than 3 emails) and noisy.

[4] Since comparing the model-generated summary to only one human-written reference may not be fully informative, recently we have also collected *one more reference* for each email thread in our test set, i.e., each test example will have two gold references now in our final dataset. The results in the paper are all still based on the original one-reference setup but we will release the updated two-reference results for our best baselines on Github.

commonly used summarization datasets: CNN/DM (Hermann et al., 2015) and XSum (Narayan et al., 2018) are about news summarization; SAMSum (Gliwa et al., 2019) is about chit-chat summarization; CRD3 (Rameshkumar and Bailey, 2020) is a role-play dialogue summarization dataset; BC3 (Ulrich et al., 2008) is another email thread summarization with 40 threads from W3C. Compared to the other datasets, the average document length in the EMAILSUM dataset is not very long, containing 233 words; long summaries are more than twice as longer than short summaries. "Ext-Oracle-R1" in Table 2 indicates how abstractive the summaries are. It computes the ROUGE-1 scores of an oracle extractive method (see Section 3.1 for details of the oracle extractive method). The lower it is, the more abstractive the dataset is. According to this score, the abstractiveness of the EMAILSUM summaries is lower than the XSum summaries, while higher than the CNNDM summaries. Furthermore, the short summaries of EMAILSUM dataset are more abstractive than its long summaries.

3 Models

The summarization models we explore in this work take the email thread as input and generate the summary as output. We experiment on EMAILSUM$_{short}$ and EMAILSUM$_{long}$ tasks separately.

3.1 Extractive

Oracle. This method maximize an evaluation metric w.r.t. the gold summary. "Ext-Oracle-R1" in Table 2 is computed from an oracle summary that maximizes ROUGE-1 (Lin, 2004).

Lead. This model simply picks the first sentence from the source document as the summary, which has surprisingly good performance on CNN/DM dataset (Narayan et al., 2018). We test two variants by selecting: (1) the first sentence of the email thread, which is usually the subject (see the example in Table 1), referred as **Lead-1**; (2) the first sentence of the email thread (the subject) plus the first sentences of every email, named **Lead-1-Email**.[5]

TextRank. This is a graph-based method (Mihalcea and Tarau, 2004). It first builds a graph between sentences by their embedding similarities; then the PageRank algorithm is applied to obtain the rank scores for each sentence, and top-rank sentences are selected as the summary.

BertSumExt. Liu and Lapata (2019b) propose to build a sentence extractor upon BERT (Devlin et al., 2019) to perform extractive summarization, which achieves a good performance on CNN/DM.

3.2 Abstractive

Fast Abs RL. As the simple non-pretrained abstractive baseline, we use Chen and Bansal (2018), which is a hybrid model that first extracts sentences from the source document, then rewrites the extracted sentences by an abstractive rewriter. They pair summary sentences with the extracted sentences to train the abstractive rewriter. Adapting their model to our email thread summarization task, we make two adjustments: (1) We extract emails instead of sentences, which is a natural unit for email thread; (2) Since summary sentences usually follow the temporal order of the emails, we enhance this pairing procedure by using the Neeleman-Wunsch algorithm (Needleman and Wunsch, 1970; Rameshkumar and Bailey, 2020) to impose the order constraint to the alignment (see description and comparison in Appendix B).

T5. T5 (Raffel et al., 2020) is a Transformer (Vaswani et al., 2017) based seq-to-seq model pretrained with large-scale English data. It achieves state-of-the-art performances on a lot of NLP tasks including the CNN/DM summarization task. As our main baseline, we take the email thread as a single document and finetune a T5 base to generate the summary (**T5**$_{base}$). A similar setup is also used in transfer and semi-supervised learning. Since our training dataset is small, we find that using the pretrained knowledge transfer is crucial. Training a T5 model from scratch performs poorly (see the results in Appendix Table 7).

Transfer Learning. To analyze how information from other summarization datasets (listed in Table 2) can be transferred to this new task and its impact on the performance, we investigate two simple transfer learning methods: (1) *Pre-finetuning*, in which we first finetune T5 on a bigger summarization dataset (e.g., CNN/DM) then continue the finetuning on our dataset, referred as **X**$_{pre}$ (X is the bigger dataset's name, e.g., **CNNDM**$_{pre}$) in our result tables. This is analogous to the continual training method proposed for multilingual transfer learning of machine translation (Kocmi and Bojar,

[5] We also tested some other heuristics: e.g., the first sentence of the last email, the last 3-5 sentences of the email thread, etc. However, none of them perform better than Lead-1-Email.

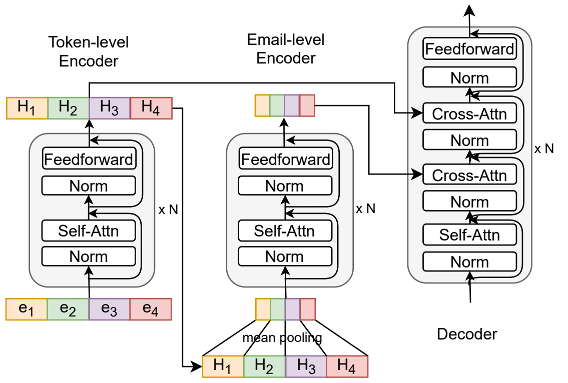


Figure 1: The architecture of our hierarchical T5.

2018). (2) *Joint-training*, in which we upsample EmailSum data and mix it with another dataset, then use the combined data to finetune T5, similarly denoted as $\mathbf{X}_{joint}$. This is analogous to the multilingual joint training method used in machine translation (Johnson et al., 2017).

Semi-supervised learning. Since we only have 2.5K labeled email threads, another important technique to improve the performance is to utilize unlabelled data (i.e., email threads without labeled summaries). As introduced in Section 2.1, in addition to the 3K email threads used for summary collection, we have 8,594 unlabelled email threads (5,116 from Avocado; 3,478 from W3C). We explore semi-supervised learning via the simple self-training technique (Scudder, 1965). We use a trained model (a finetuned T5) to generate summaries for unlabelled threads, then mix the model-labeled and human-labeled data to finetune T5 again, referred as **SemiSup**$_x$ (x stands for the unlabelled data source we use, i.e., W3C, Avocado, or together).

Hierarchical T5. Hierarchical summarization models have been shown to improve the performance of multi-document summarization task (Liu and Lapata, 2019a). Although an email thread can be treated as a single document due to the temporal dependency between consecutive emails, it also has a clear turn structure that encourages using of the hierarchical encoders. Recently, Zhu et al. (2020) proposed a hierarchical model (HMNet) for meeting summarization. Inspired by their work, we propose a hierarchical model that is similar to HMNet in structure but uses T5 as the backbone, therefore, it can take advantage of both the hierarchical structure and the pre-trained knowledge. As shown in Figure 1, this model contains two encoders: the *token-level* encodes the whole email thread (e.g., e_1, e_2, e_3, e_4) while the *email-level* receives mean-pooled email-level representations as input. The decoder has two cross attentions that attend to the outputs of the email-level and the token-level encoders respectively. Both token-level and email-level encoders are sharing the weights of the T5 encoder. We add a small number of new parameters by adding new cross attention between the decoder and the email-level encoder.

4 Experiments

4.1 Evaluation Metrics

ROUGE (Lin, 2004) is a commonly used automatic metric for summarization tasks. It has several variants: (1) ROUGE-1 (R1) measures the unigram overlap between the generated and reference summaries; (2) ROUGE-2 (R2) measures the bi-gram overlap; (2) ROUGE-L (RL) computes the longest common subsequence (LCS); (4) summary-level ROUGE-L (RLsum) computes LCS between each pair of reference and candidate sentences and returns the union-LCS. We use the `rouge_score` package[7] and report F1 scores.

BERTScore (Zhang et al., 2019) goes beyond n-gram overlap to provide contextualized semantic similarity. Specifically, it uses BERT (Devlin et al., 2019) (or RoBERTa (Liu et al., 2019)) representations to "softly" align the words in candidate and reference summaries and then computes a "soft" uni-gram F1 score. We use the `bert_score` package[8] and report rescaled numbers with a baseline.

4.2 Results

Table 3 shows the evaluation results on the testing set of different models (the corresponding results on the development set can be found in Appendix Table 7). It can be observed that the *Oracle* extractive model sets up a high upper bound on all metrics except for BERTScore (BertS). Among non-oracle extractive methods, the *Lead-1-Email* heuristic works best and even better than the deep extractive method, BertSumExt. The hybrid *Fast Abs RL* model outperforms purely extractive methods but works worse than purely abstractive methods with large-scale pretraining (e.g., T5).

[6] The significance test is following the bootstrap test setup (Efron and Tibshirani, 1994) and sample for 100k times.

[7] https://github.com/google-research/google-research/tree/master/rouge

[8] https://github.com/Tiiiger/bert_score

Models	EmailSum$_{short}$					EmailSum$_{long}$				
	R1	R2	RL	RLsum	BertS	R1	R2	RL	RLsum	BertS
Oracle	*39.04*	*12.47*	*30.17*	*35.61*	*22.32*	*45.98*	*15.49*	*32.40*	*42.14*	*26.31*
Lead-1	23.35	5.57	18.22	19.61	12.25	19.75	4.84	14.24	16.88	6.87
Lead-1-Email	26.62	5.60	19.72	23.77	13.00	35.71	8.69	24.70	32.13	16.93
TextRank	22.52	4.54	16.56	20.24	5.89	28.42	6.20	19.08	25.19	5.67
BertSumExt	24.84	5.15	17.81	21.81	7.51	30.23	7.08	19.59	26.68	7.78
Fast Abs RL	31.15	6.59	22.73	29.03	6.49	39.35	10.58	27.01	36.51	10.03
T5$_{base}$	36.57	10.56	28.3	32.76	33.90	43.81	14.08	30.47	39.88	32.09
CNNDM$_{pre}$	35.43	10.75	27.49	32.15	33.61	44.15	14.20	30.84	40.21	32.53
XSum$_{pre}$	36.14	10.26	28.66	33.47	**33.97**	43.48	13.82	30.14	39.80	31.60
SAMSum$_{pre}$	34.68	10.56	26.62	31.22	33.25	42.83	13.54	30.00	39.13	31.82
CRD3$_{pre}$	36.05	10.04	27.21	32.06	33.52	43.60	13.93	30.49	39.97	31.53
CNNDM$_{joint}$	34.38	9.27	27.20	31.30	32.70	43.28	12.37	28.84	39.39	29.95
XSum$_{joint}$	34.18	8.17	25.94	30.68	31.83	42.36	11.85	28.23	38.31	29.22
SAMSum$_{joint}$	35.57	10.07	27.95	32.57	33.55	42.96	13.44	29.99	39.54	31.82
CRD3$_{joint}$	34.66	8.81	26.95	31.59	33.29	42.81	12.96	29.35	39.33	32.14
SemiSup$_{w3c}$	35.43	10.64	28.59	32.31	33.61	**44.56****	14.60**	**31.38****	**40.73****	**32.81****
SemiSup$_{avocado}$	36.73	10.82	28.44	33.25	33.76	43.83	**14.61****	31.21**	40.52*	32.71**
SemiSup$_{together}$	**36.98**	**11.21***	**28.76**	**33.70****	33.91	44.08	14.06	31.17**	40.67**	32.30
Hier. T5$_{base}$	36.17	10.37	28.44	33.34	33.39	44.50*	14.53*	30.89*	40.22	32.30

Table 3: Summarization performance on the testing set of different models. We test the significance[6] of the improvement over T5$_{base}$ ($*$: $p < 0.05$, $**$: $p < 0.01$).

	EmailSum$_{short}$		EmailSum$_{long}$	
	EO-R1↓	LE-R1↓	EO-R1↓	LE-R1↓
Human	39.0	26.62	46.0	35.71
T5$_{base}$	50.27	36.88	55.43	43.65
R1-best	52.50	39.22	60.04	49.14

Table 4: The extractive Oracle (EO) and Lead-1-Email (LE) models' ROUGE-1 by taking human summary, base or best model generated summary as the ground-truth. The lower the scores are, the more abstractive the summaries are (↓).

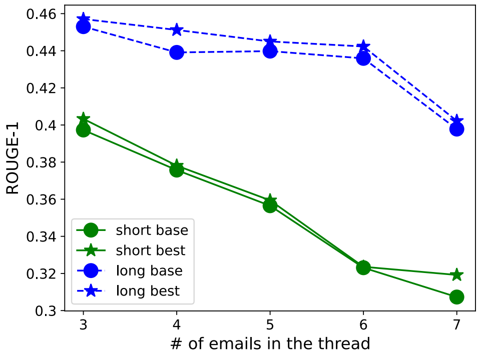


Figure 2: The impact of the number of emails in the thread on summarization performance (ROUGE-1). The results are on the testing set. short/long denotes EmailSum$_{short}$/EmailSum$_{long}$; base/best denotes the baseline/best model.

Taking the email thread as one single document and finetuning T5 (i.e., T5$_{base}$ in Table 3) sets up a strong baseline. Upon this baseline model, we test the transfer learning from four different summarization datasets (CNN/DM, XSum, SAMSum, and CRD3). However, as shown in Table 3, transfer learning barely improves over baseline, and transferring by *pre-finetuning* always works better than *joint-training*. Since our EmailSum has a quite different domain as existing news or dialogue datasets, we conjecture that it is hard to transfer knowledge between them or better transferring techniques need to be applied. Similarly, we test the semi-supervised learning with unlabelled data from W3C, Avocado, and both of them (together). This method can mostly (or significantly in some cases) outperform the baseline's performance for both EmailSum$_{short}$ and EmailSum$_{long}$. Lastly, the hierarchical T5$_{base}$ model only marginally outperforms the non-hierarchical baseline for EmailSum$_{long}$ task. It is notable that overall EmailSum$_{long}$ has higher ROUGE scores but lower BERTScore than EmailSum$_{short}$.

Since we focus on generating abstractive summaries for email threads and the human-written summaries are fairly abstractive (as shown in Table 2), we further investigate the abstractiveness of model-generated summaries. We take summaries generated by the baseline (T5$_{base}$) and the best ROUGE-1 models (SemiSup$_{together}$ for EmailSum$_{short}$, SemiSup$_{w3c}$ for EmailSum$_{long}$) as the pseudo ground-truth, respectively. Then, we evaluate the ROUGE-1 of extractive *Oracle* and *Lead-1-Email* models; higher scores means more extractive summaries. As shown in Table 4, compared

	EMAILSUM$_{short}$			EMAILSUM$_{long}$		
	SemiSup$_{together}$ vs T5$_{base}$			SemiSup$_{w3c}$ vs T5$_{base}$		
	Win	Lose	Tie	Win	Lose	Tie
Salience	109	133	55	109	130	50
Faithfulness	116	123	58	126	122	41
Overall quality	120	138	39	125	140	24

Table 5: Pairwise comparison between summaries generated by the best ROUGE-1 models and T5$_{base}$.

to humans, models generate much more extractive summaries. Moreover, the semi-supervised models (R1-best) are even more extractive than the baseline, which is probably because the self-training procedure amplifies the extraction tendency. Lastly, for both base and best models as well as for both short and long summaries, the model performance (ROUGE-1) decreases as the number of emails in the thread increases (shown in Figure 2).

5 Human Evaluation

5.1 Human Rating Collection

To better understand where the model still falls short and investigate if the automatic metrics correlate well with human judgments, we conduct a human evaluation on Amazon Mechanical Turk. Initially, by manually checking the quality of model-generated summaries, we find that models can mostly generate grammatical, relevant, and fluent summaries; however, they often fail to be salient and faithful, i.e., models tend to be over-detailed or do not stay true to the source thread. Therefore, we ask human annotators to rate the "salience" and "faithfulness" of model-generated summaries. We choose the best ROUGE-1 models, SemiSup$_{together}$ for EMAILSUM$_{short}$, SemiSup$_{w3c}$ for EMAILSUM$_{long}$, to evaluate, then we sample 100 examples, and collect 3 responses for each example. Human judges are asked to rate on a 5-point Likert scale for salience and faithfulness respectively and annotate which summary sentences are not salient or unfaithful. We explain the meaning of "salience" and "faithfulness" to annotators and instruct them how to rate from 1 to 5. Meanwhile, to verify the improvement obtained by best R1 models over T5$_{base}$, we ask them to compare the summaries generated by these models and those from T5$_{base}$, and judge which one is more salient, more faithful, and has overall higher quality. More collection details can be found in the Appendix D.

We check the average inter-rater agreement (Krippendorff's alpha (Krippendorff, 2011)) of "salience" and "faithfulness" ratings. It is around 0.09 to 0.23, i.e., slight to fair agreement (Fleiss and Cohen, 1973). However, when we convert the ratings to 3-point by taking {3}, {4 and 5}, {1 and 2} as 3 classes, the agreement increases to 0.36 to 0.63, i.e., fair to substantial agreement. This indicates that humans' subjectivity affects the ratings and people have a hard time distinguishing 'bad' from 'very bad' as well as 'good' from 'very good'. Meanwhile, the ratings for short summaries are always less agreed across raters (0.36-0.38) than that for long summaries (0.58-0.63). This indicates that there might be multiple different ways of summarizing an email thread into a short summary. The agreement of pairwise comparison is around 0.20 to 0.24 (fair agreement), which is because the baseline and the best models have non-distinguishable performance (shown in Table 5). Finally, we take the 3-rater average as the final human rating for each example.

In addition, we evaluate the correlations (*Pearson Correlation* (Benesty et al., 2009)) among different human ratings. The correlation between salience and faithfulness ratings is 0.36/0.45 for short/long summarization. And the correlations among salience, faithfulness, and overall quality pairwise preferences are around 0.53 to 0.79. Overall, moderate to large (Cohen, 2013) correlations are observed.

5.2 Generated Summary's Quality Analysis

Surprisingly, human evaluators are mostly satisfied with the salience and faithfulness of model-generated summaries, ratings are around 4 out of 5. On average, humans rate 3.89 and 4.04 for the salience and faithfulness of SemiSup$_{together}$ generated short summaries, respectively; and they rate 4.22 and 4.29 for the salience and faithfulness of SemiSup$_{w3c}$ generated long summaries, respectively. Examples with low or high ratings are shown in Table 6 or Appendix Table 8. Humans rate higher for model-generated long summaries, which is correlated to the trend of ROUGE, and they are more satisfied with faithfulness than salience.

Table 5 presents the human pairwise compari-

Fail to understand the sender's intent.
Thread: Subject: minutes of meeting: 3.5 plan ||| Om: 1. Nihar mentioned that we spent about 3 weeks in redefining the language, which was not originally planned. This is the major reason for moving the code freeze date from 8/24 to 9/21. 2. For phase-I code drop to QA on 8/28 The confidence in date is : 90% The confidence in statbility of build is : 80% 3. ... ||| Sharon: Hi Om - We also need to lock down the date for: 1 - service pack merge 2 - bug fix freeze and, Javascript library testing (Offline) resource thanks, sharon ||| Rajeev: Thanks for the meeting minutes. Nihar, Sharon can you list the Risks to the phase 1 & Phase II schedules and what we are doing to manage the risk. Rajeev
Generated Summary: Om tells Nihar that he spent 3 weeks redefining the language. Sharon tells Om that she needs to lock down the date for 1 - service pack merge 2 - bug fix freeze and Javascript library testing. (**salience=4, faithfulness=3.3**)
Ground-truth: Om gives everyone minutes for a meeting. Sharon updates Om on some other plans and Rajeev asks Nihar/Sharon for some technical details.
Fail to identify the roles of the sender and receiver.
Thread: Subject: latest 4.0 ga palladium install for biogen ||| Nilesh: PATH/patchinstaller I tested this with build version 377 and it works fine. ||| Diana: This one looks good. I have verified that the 2 fixes in 382 are in the patch installer. Just to clarify, this is really a 382 patch installer that falls under the 377 directory? ... ||| Nilesh: Wilhan, I have deleted build 382 as there was no space to create patch installer. (as we discussed in the lab) And as we specified the build version to be 377 when creating the patch installer I thought we will need to put it under build 377 and use the jar files for that. Can you please clarify this. ...
Generated Summary: Nilesh tells Diana that the 2 fixes in 382 are in the patch installer. Nileshe also asks Wilhan to clarify the definition of the build. (**salience=3.3, faithfulness=3.3**)
Ground-truth: Nilesh says he tested something with a build. Diana thinks it looks good after verifying it but asks some questions. Nilesh updates Wilhan and has some questions.

Table 6: Error analysis examples. Emails are separated by '|||' and some content is omitted by '...'. (**salience=xx, faithfulness=xx**) gives the average human rating for that summary.

son between the best ROUGE-1 models and T5$_{base}$. Except for the faithfulness of EMAILSUM$_{long}$, the best ROUGE-1 models mostly lose to the baseline (though the loss and win are mostly marginal). Together with Table 4, we conjecture that the improvement obtained by semi-supervised learning exploits n-gram matching accuracy by making the summary more extractive, while humans prefer more abstractive summaries.

Lastly, we analyze the non-salient and unfaithful sentences labeled by the human evaluators. We find that two errors are frequently made by the summarization model: **(1) Failing to understand the sender's intent.** Usually, when we send an email, there is a high-level intention behind the detailed content we write, e.g., start up a discussion, bring up a concern, broadcast a decision, etc. However, models are oftentimes unable to capture the intention and thus overly focus on details. As shown in the first example of Table 6, *Om* intends to summarize the important points from a meeting, while the model only picks the first piece of detail in that email as the summary. This problem is also related to the over-extractive issue (shown in Table 4). The model tends to extract details from the source thread and the extraction is biased to the first sentence of each email. **(2) Failing to identify the roles of the sender and receiver.** An email thread is a special type of conversation with multiple speakers involved. One important task for the model is to identify the roles of different speakers and their relations, i.e., who does what to whom. As shown in the second example of Table 6, the model wrongly takes "2 fixes in 382 are in the patch installer" as information provided by *Nilesh*, whereas it is supposed to be by *Diana*. The same issue can also be observed in the first example: *Om* is just summarizing what *Nihar* said instead of telling *Nihar*. This is considered as a type of unfaithfulness, which has been widely identified as a common issue of abstractive summarization models (Wang et al., 2020; Durmus et al., 2020; Maynez et al., 2020).

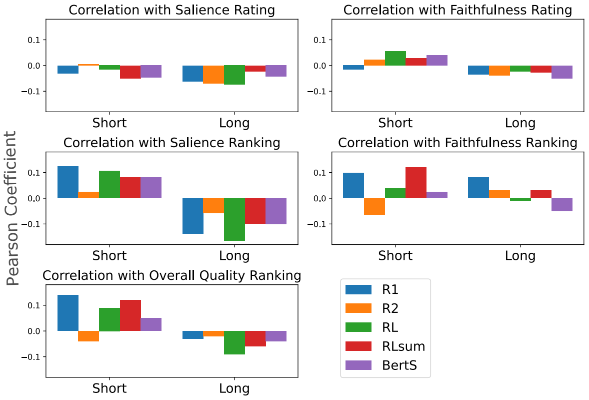


Figure 3: Correlation between automatic metrics and human judgements. Short and Long refer to EMAILSUM$_{short}$ and EMAILSUM$_{long}$ tasks, respectively.

5.3 Correlation with Human Judgement

ROUGE (Lin, 2004) measures n-gram overlap and BERTScore (Zhang et al., 2019) is essentially based on "soft" uni-gram matching. However, according to our analysis presented above, the email thread summarization models mainly fail to be abstractive, salient, and faithful, which are hard to be evaluated by n-gram overlap. Furthermore, as pointed out by Bhandari et al. (2020), different datasets usually require different evaluation metrics. Therefore, here, we study the correlation between automatic metrics and human judgments.

Specifically, we evaluate the *Pearson Correlation* between human ratings and automatic metric scores on the 100 examples used in the human evaluation. Besides, as described above, we conduct a pairwise model comparison between the best ROUGE-1 models and T5$_{base}$ for "salience", "faithfulness", and "overall quality". We convert them to a pairwise ranking score, i.e., -1 if T5$_{base}$ is better; 1 if T5$_{base}$ is worse; 0 if two models are non-distinguishable. In the same way, we convert different metric scores to ranking scores. Then, we also evaluate the *Pearson Correlation* between human and metric ranking scores. Figure 3 illustrates the results. Overall, the correlations are fairly poor. The best correlation is between ROUGE-1 and human overall quality ranking for short summary generation (coefficient=0.14, p=0.16). There is little or negative correlation between metrics and human judgment for the long summary generation. Therefore, we emphasize the importance of human evaluation and better automatic proxies need to be proposed in the future.

6 Conclusion

In this work, we propose an abstractive email thread summarization dataset, EMAILSUM, that contains 2,549 email threads with human-written short and long summaries. We explore different summarization paradigms and find that taking the email thread as a single document and finetuning T5 (Raffel et al., 2020) sets up a good baseline. Transferring from other summarization datasets barely improves it. Using hierarchical structure also only marginally improves the performance. Semi-supervised learning by using unlabelled email threads improves automatic metrics (ROUGE) but still loses to the baseline in human evaluation. Finally, our human evaluation reveals that the model fails to understand the sender's main intention and the roles of different speakers. Automatic metrics are poorly correlated with human judgment, which emphasizes the importance of human evaluation and designing new metrics for this task in the future.

7 Broader Impact Statement

We use two email collections in this work: Avocado (Oard et al., 2015) and W3C (Craswell et al., 2006). W3C is derived from W3C Public Mailing List that is open-source available online. Avocado consists of emails and attachments taken from 279 accounts of a defunct information technology company referred to as "Avocado". Its copyright is protected by Linguistic Data Consortium. Based on the license agreement, we will only open-source our collected summaries and provide scripts to obtain email threads from the original Avocado email collection. To further protect copyright and the privacy of the persons involved in the emails, as introduced in Section 2, we carefully anonymize all the email threads we construct from both email collections. We fairly pay crowd-source workers \$1.37 (for threads with 5 or fewer emails) or \$2 (for threads with more than 5 emails) for writing the short and long summaries and \$0.6 for human rating such that the pay rate is higher than the federal minimum wage requirement.

Acknowledgments

We thank the reviewers for their helpful comments and Xiang Zhou for useful discussions. We thank Saadia Gabriel, Yichen Jiang, Tom McCoy, and Yixin Nie for helping write summary examples (to show as initial examples to MTurk annotators) and estimate the workload for deciding the fair payment. This work was partially done while SZ was interning at MSR and later extended at UNC, where it was supported by NSF-CAREER Award 1846185, ONR Grant N00014-18-1-2871, and a Microsoft Investigator Fellowship. The views contained in this article are those of the authors and not of the funding agency.

References

Adobe. 2019. Adobe email usage study.

James Allan, Rahul Gupta, and Vikas Khandelwal. 2001. Temporal summaries of new topics. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 10–18.

Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. 2009. Pearson correlation coefficient. In *Noise reduction in speech processing*, pages 1–4. Springer.

Manik Bhandari, Pranav Narayan Gour, Atabak Ashfaq, Pengfei Liu, and Graham Neubig. 2020. Re-evaluating evaluation in text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9347–9359, Online. Association for Computational Linguistics.

Giuseppe Carenini, Raymond T Ng, and Xiaodong Zhou. 2007. Summarizing email conversations with clue words. In *Proceedings of the 16th international conference on World Wide Web*, pages 91–100.

Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, Guillaume Lathoud, Mike Lincoln, Agnes Lisowska, Iain McCowan, Wilfried Post, Dennis Reidsma, and Pierre Wellner. 2006. The ami meeting corpus: A pre-announcement. In *Machine Learning for Multimodal Interaction*, pages 28–39, Berlin, Heidelberg. Springer Berlin Heidelberg.

Yen-Chun Chen and Mohit Bansal. 2018. Fast abstractive summarization with reinforce-selected sentence rewriting. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–686.

Jacob Cohen. 2013. *Statistical power analysis for the behavioral sciences*. Academic press.

Nick Craswell, Arjen P Vries, and Ian M Soboroff. 2006. Overview of the trec-2005 enterprise track. In *Text Retrieval Conference (TREC)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Esin Durmus, He He, and Mona Diab. 2020. Feqa: A question answering evaluation framework for faithfulness assessment in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5055–5070.

Bradley Efron and Robert J Tibshirani. 1994. *An introduction to the bootstrap*. CRC press.

Joseph L Fleiss and Jacob Cohen. 1973. The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. *Educational and psychological measurement*, 33(3):613–619.

Shen Gao, Xiuying Chen, Zhaochun Ren, Dongyan Zhao, and Rui Yan. 2020. From standard summarization to new tasks and beyond: Summarization with manifold information. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 4854–4860. International Joint Conferences on Artificial Intelligence Organization. Survey track.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. Samsum corpus: A human-annotated dialogue dataset for abstractive summarization. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 70–79.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in neural information processing systems*, pages 1693–1701.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Bryan Klimt and Yiming Yang. 2004. The enron corpus: A new dataset for email classification research. In *European Conference on Machine Learning*, pages 217–226. Springer.

Tom Kocmi and Ondrej Bojar. 2018. Trivial transfer learning for low-resource neural machine translation. *WMT 2018*, page 244.

Klaus Krippendorff. 2011. Computing krippendorff's alpha-reliability.

Yanran Li and Sujian Li. 2014. Query-focused multi-document summarization: Combining a topic model with graph-based semi-supervised learning. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1197–1207.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Yang Liu and Mirella Lapata. 2019a. Hierarchical transformers for multi-document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5070–5081.

Yang Liu and Mirella Lapata. 2019b. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3721–3731.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411.

Amita Misra, Pranav Anand, Jean E Fox Tree, and Marilyn Walker. 2015. Using summarization to discover argument facets in online idealogical dialog. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 430–440.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807.

Saul B Needleman and Christian D Wunsch. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453.

Douglas Oard, William Webber, David Kirsch, and Sergey Golitsynskiy. 2015. *Avocado Research Email Collection LDC2015T03*. DVD. Philadelphia: Linguistic Data Consortium.

Radicati. 2015. Email statistics report, 2015-2019.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Revanth Rameshkumar and Peter Bailey. 2020. Storytelling with dialogue: A critical role dungeons and dragons dataset. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5121–5134.

H Scudder. 1965. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11(3):363–371.

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*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5008–5020.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

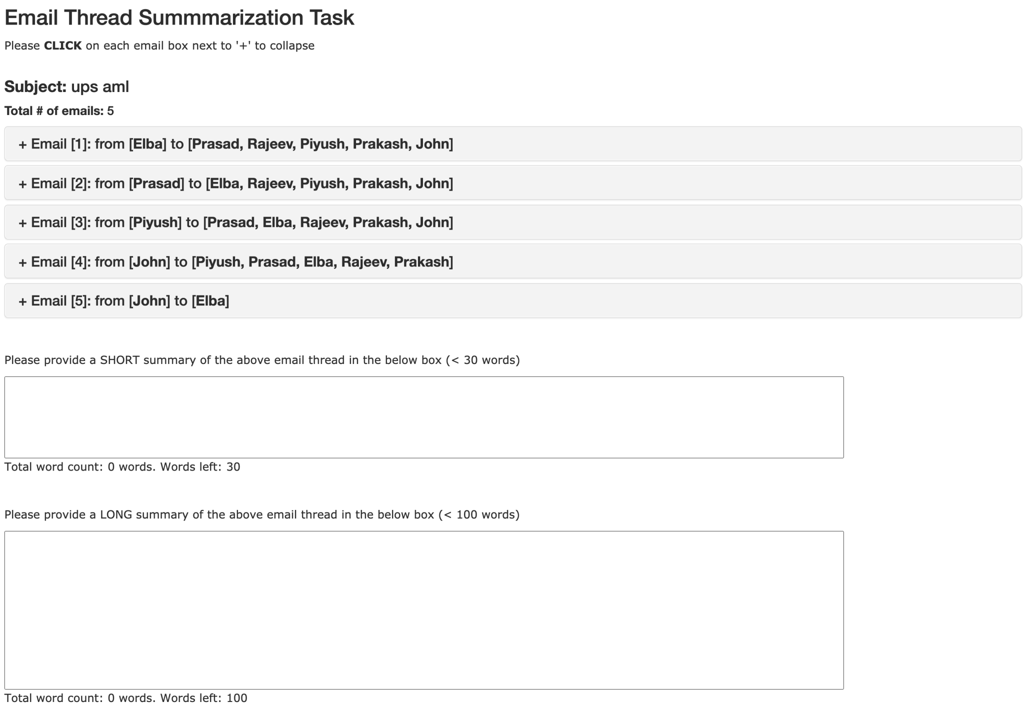
Chenguang Zhu, Ruochen Xu, Michael Zeng, and Xuedong Huang. 2020. A hierarchical network for abstractive meeting summarization with cross-domain pretraining. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 194–203.

Junnan Zhu, Haoran Li, Tianshang Liu, Yu Zhou, Jiajun Zhang, and Chengqing Zong. 2018. Msmo: Multimodal summarization with multimodal output. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 4154–4164.

Appendix

A Summary Collection

Figure 4 illustrates the questions we asked human annotators on Amazon Mechanical Turk during summary collection. Before these questions, here are some important instructions we listed on the webpage: (1) Long summary MUST be longer than short summary; (2) Summary length can be dynamically decided based on the content of the thread; (3) Short summary should be a *concise and abstractive* description of what the thread is mainly talking about; (4) Long summary can be a narrative of what happens. But do NOT simply summarize each email separately. The summary should be *coherent*; (5) It is NOT necessary to summarize every email in the long summary, i.e., it is OK to skip



Email Thread Summmarization Task

Please **CLICK** on each email box next to '+' to collapse

Subject: ups aml

Total # of emails: 5

+ Email [1]: from **[Elba]** to **[Prasad, Rajeev, Piyush, Prakash, John]**

+ Email [2]: from **[Prasad]** to **[Elba, Rajeev, Piyush, Prakash, John]**

+ Email [3]: from **[Piyush]** to **[Prasad, Elba, Rajeev, Prakash, John]**

+ Email [4]: from **[John]** to **[Piyush, Prasad, Elba, Rajeev, Prakash]**

+ Email [5]: from **[John]** to **[Elba]**

Please provide a SHORT summary of the above email thread in the below box (< 30 words)

Total word count: 0 words. Words left: 30

Please provide a LONG summary of the above email thread in the below box (< 100 words)

Total word count: 0 words. Words left: 100

Figure 4: A part of the Amazon Mechanical Turk webpage used for collecting summaries.

unimportant ones and merge similar ones if needed; (6) You are *encouraged* to include important sender and/or receiver names in long summary; (7) You are *disencouraged* to copy a lot from emails for both short and long summaries; You are supposed to write in your own words as much as you can; (8) You may find some content are technical. We do NOT expect any background knowledge. Just focus on the major concerns, decisions, and consensus. (9) In the thread, emails are ordered by time. However, one email does NOT necessarily reply to the previous one. It can reply to an earlier email OR forward to new receivers. In other words, the structure is NOT always continuous, so please be careful when you read.

B Fast Abs RL

The original Fast Abs RL method (Chen and Bansal, 2018) uses ROUGE-L$_{recall}$ to align extracted source sentences and target summary sentences. In our case, we extract emails and align them with summary sentences. Since the emails and summary sentences usually follow the same temporal order, we enhance the alignment procedure by the Neeleman-Wunsch algorithm (Needleman and Wunsch, 1970; Rameshkumar and Bailey, 2020) to imposing strict order constraints, e.g., there should not be "email$_i$ is aligned to sentence$_j$ while email$_{i+1}$ is aligned to sentence$_{j-1}$" cases. Meanwhile, we modify it to allow one email to be aligned with multiple summary sentences but avoid one summary sentence aligning with multiple emails. Specifically, we first obtain the similarity matrix M of size $n_e \times n_s$ between each email and summary sentence by ROUGE-L$_{recall}$ (n_e is the number of emails, n_s is the number of summary sentences); then the alignment score matrix H of size $(n_e\!+\!1)\!\times\!(n_s\!+\!1)$ is initialized as all-zero then computed as follows for $1 \le x \le n_e$, $1 \le y \le n_s$:

$$H_{x,y} = \max \begin{cases} H_{x-1,y-1} + M_{x-1,y-1} \\ H_{x,y-1} + M_{x-1,y-1} \\ H_{x-1,y} \end{cases}$$

Then we traceback from H_{n_e,n_s} to $H_{0,0}$ to obtain the final alignment. As shown in Table 7, the "Fast Abs RL (default)" model refers to this method with the default setting which works mostly worse than our enhanced Fast Abs RL.

C Experimental Details & Additional Results

We implement the TextRank (Mihalcea and Tarau, 2004) model via the `summa` python package[9] and set the summarization ratio as the average $\frac{summary\ length}{thread\ length}$ ratio in the training set, which is 0.22 for short summary and 0.38 for long summary.

[9] `https://github.com/summanlp/textrank`

We test Fast Abs RL (Chen and Bansal, 2018) via the author's open-source code.[10] Most of our models are built on T5 (Raffel et al., 2020) and we use the base version that has 220 million parameters. Our hierarchical T5 shares the same T5 encoder parameters between the token-level and email-level encoders. The only new parameters added are from the first cross attention between decoder and email-level encoder. We use `Transformers` (Wolf et al., 2020)[11] to run all the T5 based models. We run experiments on a single Tesla V100 GPU. We set the max input sequence length as 512 tokens and max output length as 56 tokens during training (200 tokens during evaluation). The total batch size (with gradient accumulation) is 128. The learning rate is 5e-4, except for training the T5_{base} from scratch, we use 1e-4 instead. Since our training set only contains 1.8K examples, it only takes 2-4 minutes per epoch. We train models for 70 epochs.

Our model selection is based on each of the five evaluation metrics, ROUGE-1/ROUGE-2/ROUGE-L/summary-level ROUGE-L/BERTScore. We select the best checkpoints for each of the five metrics on our development set, then test those checkpoints on the testing set to report the final numbers for each metric. Table 7 shows all the results on our development set. Table 8 shows two examples that have high-rating model-generated summaries.

D Human Evaluation

Figure 5 shows the questions we asked to human judges to evaluate the quality of model-generated summaries. Before these questions, we instruct annotators how to rate on a 5-point Likert scale for "salience" and "faithfulness": (1) Rate *salience* from 1 to 5: 1 is the worst, none of the points in the summary is important enough to be summarized; 5 is the best, all of the points mentioned in the summary are important and worth to be summarized; (2) Rate *faithfulness* from 1 to 5: 1 is the worst, all of the sentences in the summary are either wrong or not existing in the email thread; 5 is the best, all of the points mentioned in the summary are true to the thread. Plus, we also prompt examples of "non-salient" and "unfaithful" summaries on the webpage. We pay annotators $0.60 per HIT.

[10] https://github.com/ChenRocks/fast_abs_rl

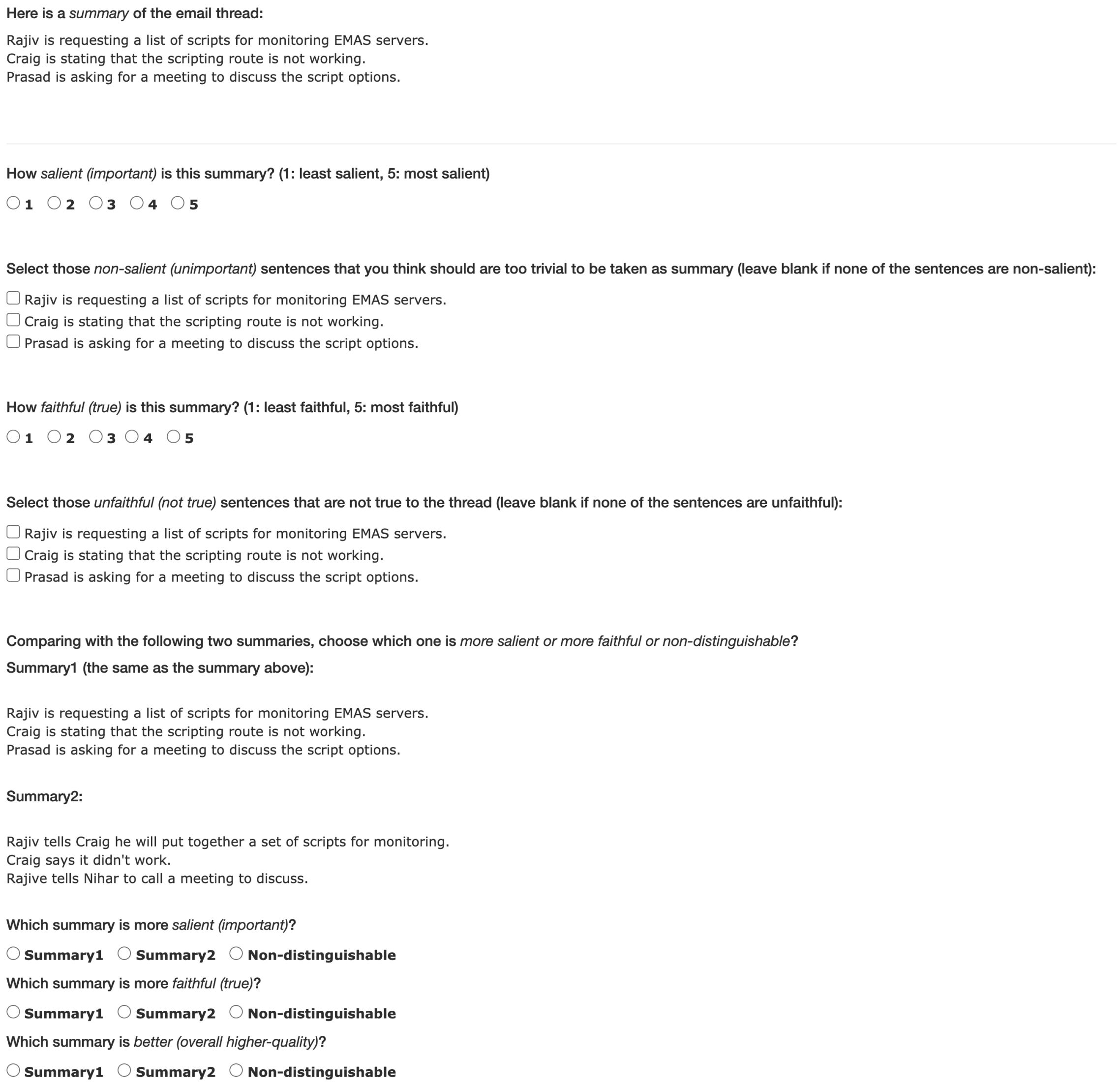
[11] https://github.com/huggingface/transformers

Models	EMAILSUM$_{short}$					EMAILSUM$_{long}$				
	R1	R2	RL	RLsum	BertS	R1	R2	RL	RLsum	BertS
Oracle	*39.8*	*13.05*	*31.17*	*36.15*	*28.50*	*46.74*	*17.13*	*33.92*	*43.1*	*28.38*
Lead-1	25.63	6.56	19.97	21.51	13.55	20.72	5.87	15.23	18.01	8.09
Lead-1-Email	26.37	5.88	19.68	23.61	12.98	36.65	10.44	26.00	33.27	18.11
TextRank	21.91	4.20	16.12	19.57	6.56	29.00	7.15	20.00	25.92	10.44
BertSumExt	25.76	6.02	18.74	22.59	8.34	30.90	8.29	20.91	27.55	8.92
Fast Abs RL (default)	29.67	6.08	22.68	27.66	6.92	39.43	11.08	25.78	36.81	7.14
Fast Abs RL	31.56	6.52	23.01	29.51	5.59	39.24	11.25	27.77	36.72	9.63
T5$_{base}$ (from scratch)	19.71	1.95	14.88	16.75	22.52	24.51	3.72	15.72	21.91	9.70
T5$_{base}$	36.78	11.93	29.50	33.58	34.92	44.94	15.94	32.33	41.22	33.67
CNNDM$_{pre}$	37.00	11.26	28.97	33.49	35.09	44.83	15.88	32.02	41.25	33.89
XSum$_{pre}$	36.63	11.43	29.43	33.75	35.29	44.55	15.29	31.50	40.87	33.47
SAMSum$_{pre}$	36.72	11.1	28.73	33.21	35.82	44.31	15.36	31.45	40.63	33.60
CRD3$_{pre}$	36.84	11.57	29.19	33.38	35.37	44.57	15.73	31.87	40.91	33.47
CNNDM$_{joint}$	35.89	10.41	28.02	32.41	34.02	43.92	14.48	30.54	39.99	31.67
XSum$_{joint}$	35.07	9.26	27.18	31.53	34.27	43.36	13.35	29.44	39.45	30.97
SAMSum$_{joint}$	36.59	11.20	29.20	33.49	35.44	44.38	15.23	31.68	40.69	33.65
CRD3$_{joint}$	36.24	10.43	28.55	32.72	35.52	44.25	14.87	31.24	40.38	33.57
SemiSup$_{w3c}$	37.03	11.92	29.30	33.78	35.60	45.03	16.09	32.50	41.52	33.95
SemiSup$_{avocado}$	37.78	12.56	30.09	34.50	34.88	45.49	16.21	32.97	41.82	34.42
SemiSup$_{together}$	37.43	12.26	29.84	34.32	35.08	45.73	16.27	32.65	41.91	34.09
Hier. T5$_{base}$	36.67	11.79	29.13	33.58	35.71	45.26	16.13	32.62	41.55	33.99

Table 7: Summarization performance of different models on the development set.

Examples of high-quality summaries generated by the model.
Thread: Subject: faa demos ||| Dan: PM Team, Attached are some general ideas and issues around developing new demos for our new target markets. Please review and provide feedback. Also, please provide links where we can learn more about various FAA applications. Thanx, Dan. ||| Dan, Thanks for putting the high level descriptions together. My questions are: *Is it practical to do an EAI demo given the inherent complexity of application integration? ... *Should we delay looking at Outlook for now?... *What do you think that timelines are developing these demos? ... Alex ||| Alex, Thanks for the feedback, please see my comments below: *Generated Short Summary*: Dan asks the PM team to review and provide feedback on FFA demos. Alex responds with questions. Dan thanks Alex and gives his feedback. (**salience=4.3, faithfulness=4.7**) *Ground-truth*: Dan talks about general ideas about demos to his PM team. Alex provides some feedback and asks questions. Dan thanks Alex for the feedback and adds comments.
Thread: Subject: sun performance report ||| Mahesh: Hi, I am attaching the draft of the performance/sizing report for EMAS on Sun. Please send me your comments. I am also attaching a list of features that would be good to have. Thanks, Mahesh ||| Amitabh: do we have a side-by-side comparison of solaris, hp-ux, and nt? also, a price-performance comparison might also be useful ||| Rajeev: Dan, Please consider Amitabh's suggestions for the sizing requirement document that you are prepaing... ||| Mahesh: we do not have comparison stats. It would be good to have them. ||| Dan: Good points, we should have side-by-side comparisons and also price/performance... *Generated Long Summary*: Mahesh is attaching a draft of the performance/sizing report for EMAS on Sun and asking for comments. Amitabh asks if there is a side-by-side comparison of solaris, hp-ux, and nt. Rajeev asks Dan to consider Amibh's suggestions for the sizing requirement document. Mahesesh says there are no comparison stats, but it would be good to have them. Dan says there should be side- by-side comparies and also price/performance. (**salience=4.3, faithfulness=5**) *Ground-truth*: Mahesh shows everyone a performance report for a future meeting and attaches his feedback. Amitabh gives feedback which Rajeev asks Dan to consider in a different task. Mahesh and Dan make suggestions about comparisons.

Table 8: Examples of high-quality summaries generated by model. Emails are separated by '|||' and some content are omit by '...'. (**salience=xx, faithfulness=xx**) gives the average human rating for that summary.



Here is a *summary* of the email thread:

Rajiv is requesting a list of scripts for monitoring EMAS servers.
Craig is stating that the scripting route is not working.
Prasad is asking for a meeting to discuss the script options.

How *salient (important)* is this summary? (1: least salient, 5: most salient)

○ **1** ○ **2** ○ **3** ○ **4** ○ **5**

Select those *non-salient (unimportant)* sentences that you think should are too trivial to be taken as summary (leave blank if none of the sentences are non-salient):

□ Rajiv is requesting a list of scripts for monitoring EMAS servers.
□ Craig is stating that the scripting route is not working.
□ Prasad is asking for a meeting to discuss the script options.

How *faithful (true)* is this summary? (1: least faithful, 5: most faithful)

○ **1** ○ **2** ○ **3** ○ **4** ○ **5**

Select those *unfaithful (not true)* sentences that are not true to the thread (leave blank if none of the sentences are unfaithful):

□ Rajiv is requesting a list of scripts for monitoring EMAS servers.
□ Craig is stating that the scripting route is not working.
□ Prasad is asking for a meeting to discuss the script options.

Comparing with the following two summaries, choose which one is *more salient or more faithful or non-distinguishable*?

Summary1 (the same as the summary above):

Rajiv is requesting a list of scripts for monitoring EMAS servers.
Craig is stating that the scripting route is not working.
Prasad is asking for a meeting to discuss the script options.

Summary2:

Rajiv tells Craig he will put together a set of scripts for monitoring.
Craig says it didn't work.
Rajive tells Nihar to call a meeting to discuss.

Which summary is more *salient (important)*?

○ **Summary1** ○ **Summary2** ○ **Non-distinguishable**

Which summary is more *faithful (true)*?

○ **Summary1** ○ **Summary2** ○ **Non-distinguishable**

Which summary is *better (overall higher-quality)*?

○ **Summary1** ○ **Summary2** ○ **Non-distinguishable**

Figure 5: A part of the Amazon Mechanical Turk webpage used for human evaluation.