Incremental temporal summarization in multiparty meetings

Ramesh Manuvinakurike, Saurav Sahay, Wenda Chen, Lama Nachman
Intel labs, USA
firstname.lastname@intel.com

Abstract

In this work, we develop a dataset for incremental temporal summarization in a multiparty dialogue. We use crowd-sourcing paradigm with a model-in-loop approach for collecting the summaries and compare them with the expert-generated summaries. We leverage the question generation paradigm to automatically generate questions from the dialogue, which can be used to validate the user participation and potentially also draw attention of the user towards the contents that need to be summarized. We then develop several models for abstractive summary generation in the Incremental temporal scenario. We perform a detailed analysis of the results and show that including the past context into the summary generation yields better summaries as measured by ROUGE scores.

1 Introduction

In meetings, distractions by stimuli such as an email, text messages, Slack messages, or in virtual at-home meetings by a child or a pet requiring immediate attention impact the concentration negatively. This exacerbates 'Zoom fatigue' (fatigue caused by participating in too many virtual meetings) (Fosslien and Duffy, 2020) and impacts productivity negatively. One of the approaches suggested to optimize the concentration levels is to take frequent notes, which helps maintain engagement (Peper et al., 2021). However, some distractions require immediate attention and are unavoidable, or the participant may just tune-out during the meetings. A note-taking tool designed to help capture the notes for the time the user was distracted could be useful for the participants. Such a tool that produces notes taking the past notes from the users and incrementally updating the notes for the time missed from the meeting could be useful. The goal of this work is to develop a dataset that helps us move towards the development of such an automatic dialogue summarizer that captures the notes for the chunks of time using the transcriptions and the past notes. The task of incremental temporal summarization in dialogue that is developed in this work has two main aspects to it, i) The content being summarized has a temporal order, meaning the information evolves over time. All conversations are temporal in nature, however, the current datasets on dialogue summarization (Carletta et al., 2005; Janin et al., 2003; Liu et al., 2019a; Gliwa et al., 2019; Lacson et al., 2006; Favre et al., 2015) consist of summaries that are written for the entire dialogue or parts of it (not in a sequence). Thus the summaries are not in temporal order. ii) The summaries build upon or use the past context (transcriptions, summaries, or human notes) to generate the summaries for the current dialogue. To the best of our knowledge, current datasets on dialogue summarization do not possess incremental property.

The incremental temporal summarization task bears a resemblance to the tasks of Temporal summarization (TS) and Incremental Update Summarization (IUS) of news articles (Dang and Owczarzak, 2008; McCreadie et al., 2014; Aslam et al., 2015). These tasks are set up as a summarization task that utilizes news articles/summaries from the past along with the current newly available article to which the summary needs to be generated under the assumption that the user is aware of the past contents. Incremental Temporal Summarization (ITS) for dialogue introduced in our work highlights challenges that are associated with processing human dialogue due to its incremental nature (Poesio and Rieser, 2010; Schlangen and Skantze, 2011; DeVault et al., 2011). For instance, the information (utterances, visual and prosodic signals) comes continuously and in smaller increments of time and at a much faster rate than news

Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 530–541
July 29–31, 2021.

articles. Contents to summarize also depend on dyadic exchanges (Question and answers). Disfluencies and the dynamic nature of dialogue introduces new challenges. To the best of our knowledge, while the corpora for TS and IUS exist for the news/Twitter feed summarization, a corpus for multi-party meeting scenarios does not exist. The first contribution of this work is towards providing a dataset for 'incremental temporal summarization' in a meeting scenario.

Our second contribution is that of providing a model-in-the-loop approach for summary data collection using crowd-sourcing. Crowd-sourcing summaries data collection has proven to be a challenging task as the task is non-trivial, subjective, and often ambiguous. In this work, motivated by a promising multi-step approach developed by Jiang et al. (2018) for crowd-sourcing summary data collection, we extend the literature by developing a model-in-the-loop approach for collecting summaries. The participants first read the context, mark extractives highlighting important utterances, answer automatically generated multiple-choice questions, and then provide an abstractive summary. We evaluate this approach by comparing the summaries generated by crowd-workers with those created by experts.

Our third contribution is towards the development and evaluation of baselines for ITS task and showing that the models, when provided with the context, generate better summaries than the counterparts which do not have access to the past context. While the focus of this work is not to provide new models, we develop the baselines using the recent transformer-based architectures that have performed well in the summarization tasks (Lewis et al., 2020; Zhang et al., 2020; Raffel et al., 2020).

2 Related work

Dialogue summarization corpora (Carletta et al., 2005; Janin et al., 2003; Lacson et al., 2006; Favre et al., 2015; Misra et al., 2015; Barker et al., 2016; Liu et al., 2019a; Gliwa et al., 2019) have helped accelerate the research in the area of conversational summarization and helped identify the differences in the dialogue and news article summarization (Jung et al., 2019). Our dataset could help progress the field by identifying similar differences and developing summarization model for incremental scenarios.

Collecting such conversational summarization

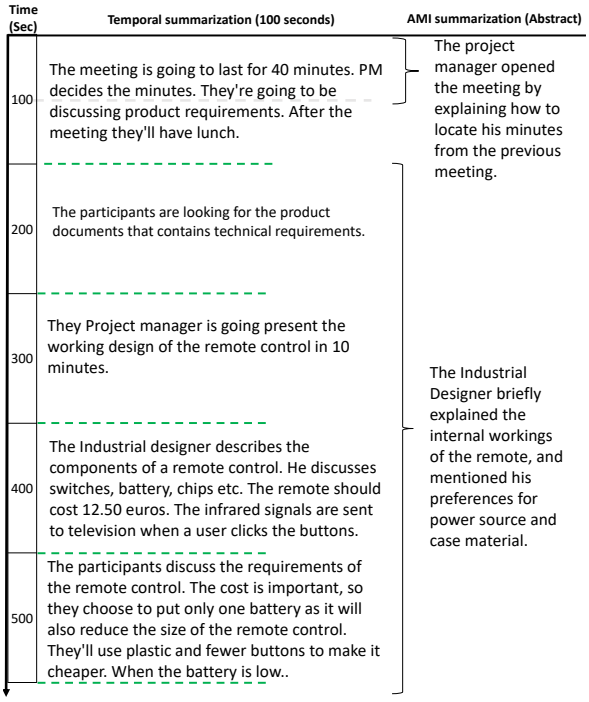

Figure 1: Figure shows a sample extract from our corpus compared to the summary from the AMI corpus

corpora can be expensive and time-consuming. Crowd-sourcing has emerged as a popular approach for collection and evaluation for numerous tasks. The task of summarization is, however, complex and subjective. Researchers in the past have experimented with collecting summarization data by framing the problem as a collection of open-ended descriptions or collecting question-answer pairs on the conversation. These approaches have yielded promising yet mixed results (Lloret et al., 2013). Hence, tasks are often simplified into subtasks automatically and requesting crowd-workers to rate, arrange or rephrase the content (Falke and Gurevych, 2017; Ouyang et al., 2017). In Jiang et al. (2018), the authors describe 'pin-refine' method where the crowd-workers perform the extractive task and abstractive summarization tasks in separate steps. To ensure the workers who provide abstractive summaries are aware of the content being summarized, they request the workers to also provide a justification that is validated by the expert. We extend the literature in this direction by developing a model-in-the-loop semi-automated approach for validation and collecting the summaries.

In recent times, deep learning models (Li et al., 2019; Liu et al., 2019b) and especially transformer-based models, have achieved impressive perfor-

mance in abstractive summarization task (Zhang et al., 2020; Raffel et al., 2020; Lewis et al., 2020; Zhu et al., 2020). Such transformer-based models are typically pre-trained on a large dataset and then fine-tuned on a smaller dataset to achieve impressive performance. In this work, we adopt the current state-of-the-art transformer architecture and utilize and evaluate transfer learning to generate summaries. Our contribution is not to develop a new model architecture for summarization but rather to benchmark and to adapt the training methodology for incremental temporal summarization tasks.

Automatic question-answer (QA) generation in the process of summarization has shown promise in recent times (Guo et al., 2018; Dong et al., 2020). Such an automated QA generation method is used to verify if the generated summary entails the same information as the content by matching the answer generated from the content and the summary. Our corpus also contains a collection of QA pairs for the conversations, which could be useful for training such systems. In our work, we utilize an automated transformer-based QA generation approach (Alberti et al., 2019; Chan and Fan, 2019; Lopez et al., 2020) to generate the QA from the dialogues.

3 Data Collection

In this work, we extend the AMI meetings corpus (Carletta et al., 2005) with the incremental temporal summaries. AMI is a multi-modal corpus consisting of conversations between 4 role-playing participants (Project Manager (PM), Industrial Designer (ID), User Interface expert (UI), and Marketing expert (ME)) in a remote-control design scenario. Each group of four participants meet four times and continue the conversation forward from the previous sessions but often on a new agenda. The AMI corpus also consists of extractive and abstractive summaries for the conversation annotated by experts. One important thing to note is that the summaries are not temporal and incremental. Summaries are often independent and can have overlapping or shared utterances with other summaries and correspond to variable time duration.

For collecting data for ITS scenario, we split the conversation videos into 100 second time duration (called dialogue chunks) and collect extractive and abstractive summaries for each of these dialogue chunks. We use Amazon Mechanical Turk (MTurk) for data collection. Our task on MTurk was available to participants in the US and Canada with an acceptance rate of above 85% in a minimum of 50 tasks. We pay the users $3.00 per dialogue chunk. (Avg. $18.00 per hour) We describe the process of setting the pay in Appendix A.2.

3.1 Data Collection Pipeline

The ITS data collection process of every dialogue chunk is broken down into four steps. The participants are presented with an interface clearly explaining each step (S) that needs to be carried out:

(**S0**) **Read context summaries:** In the first step, the user is asked to read the context, i.e., the summaries of the past 5 minutes (referred to as 'context' henceforth in the paper) of the conversation provided as three paragraphs (abstractive summary of the past 3 dialogue chunks). The users are requested to read the context and asked to tick a check box next to each paragraph acknowledging that they've read the context.

(**S1**) **Mark extractives:** The users are then required to watch the video with a conversation between the participants. The video's transcriptions are presented next to the video, with the current text being conversed highlighted as the video is played back. The users can also select the current transcript while the video is being played back. The instruction is given to the participants that these highlighted texts should help them write a summary of the conversation.

(**S2**) **Answer MCQ:** The users are then requested to answer five multiple-choice questions (MCQ). The first two questions are generic (What is the meeting about? & Did reading context help you understand the conversation better?). The remaining three are automatically generated (Section 3.2). The users can see the utterance for which the question is generated along with the question and the multiple-choice answer candidates.

(**S3**) **Provide abstractive summary:** After answering the MCQs, the users are asked to summarize the conversation in their own words. The transcriptions highlighted by the users in step 2 are shown next to the text area where the users were asked to input the summaries.

3.2 Automatic question-answer generation

In this section, we describe how the question-answers were generated automatically in step **S2**. The 3 MCQs for the data collection pipeline are generated automatically using the text from the conversation transcriptions that the users are currently annotating. We utilize a BERT-based model to train the question generator (QGen). The model is a sequence-to-sequence BERT-base model[1] implemented in the Huggingface library (Wolf et al., 2019). The model is trained to generate questions given the input utterance and the answer span. The QGen model is pretrained on the SQUAD dataset (Rajpurkar et al., 2016) and then fine-tuned on 400 QA pairs data created from a randomly sampled AMI dialogue for this work. These QA pairs were generated by an expert annotator using the utterances that have INFORM, ELICIT-INFORM, SUGGEST, and ELICIT-OFFER-OR-SUGGESTION dialogue acts. These dialogue acts were chosen due to their longer utterance length (# tokens). These dialogue-acts are annotated in the original AMI dataset. Since we use only 400 QA from a single dialogue, the evaluation of the model is not informative of the performance. We found that fine-tuning the models on these 400 QA pairs generated questions with better surface forms. However, we leave further evaluation of QGen models for future work.

E.g utterances and questions are shown below. A sample utterance from AMI with the span (within <hl> tags) is the annotated answer:

1. Utterance: "<hl> everybody <hl> found his place again ? yeah ?".

Question generated: "Who found his place again?".

2. Utterance: "there 's <hl> our ghost mouse <hl> again .".

Question generated: "What is there again?"

When generating the questions for the crowd-sourcing task, the model takes the utterance with the answers marked within the span (within <hl> tags) as input and generates the question. During run-time, we extract the answers from utterances using out-of-the-box BERT-based Semantic Role labeler (SRL) from Allennlp toolkit (Gardner et al., 2018). The approach to utilize SRL entities for generating questions has yielded promising results (Dhole and Manning, 2020). For each verb that is predicted by the SRL model, we extract the ARG0, ARG1, ARG2 (Propbank labels (Bonial et al., 2010), these are usually the noun entities) entities and wrap these arguments within <hl> tags to indicate the answers for which the QGen model generates the question. Typically, each utterance produces more than one question (due to multiple ARGs in an utterance). We pick a question randomly from the generated questions for the MCQ (in step S2). If no ARG entities were extracted for the utterances, we do not generate the questions for the utterance. As the choices for the MCQ, we provide the ARG corresponding to the question, a random SRL entity sampled from the conversation, 'Question doesn't make sense' and 'Other' (with a text box next to it for the users to type in the answer) as the four options. 5.8% of the answers were marked with 'Questions made no sense' while 18.9% of the users marked 'Others' and chose to type the answers to the questions, indicating that the questions made sense, but the answer span selected automatically was incorrect. We point out that the contribution of this work is rather the application of the automatic question-generation model to the process of data collection and not the model itself. We now briefly discuss the effect of question-answering (step **S2**) on the summaries generated by the users.

3.3 Effects of Question-Answering

In order to verify if the step **S2** (MCQ Question-answering) had any effect on the quality of the summary generated, we perform a preliminary analysis of the Crowd-worker (CW) summaries. It is important to note that the purpose of this analysis is not to verify if the step **S2** improves the correctness of the summary provided but rather to see if it affected the summaries. We collected summaries following the steps mentioned in Section 3.1 data from 50 dialogue chunks but without Step **S2** for this analysis. We compare the ROUGE, and BERTScores (Zhang et al., 2019) between the CW-CW summaries with and without step **S2**. We find that there is a significant difference (Pairwise t-test, $p < 0.05$) between the ROUGE (R-1, R-2, R-L) scores. In Table 3 we can observe that the ROUGE and BERTScore is lower in conditions with the step **S2** and without step **S2**. From this, we can imply that the summaries provided by the users when subjected to step **S2** agree more with other CW than those who provided a summary for the same dialogue without step **S2**. However, from this analysis, we cannot

[1] https://huggingface.co/bert-base-uncased

infer that the summaries from CW without step **S2** were incorrect. We then look at the rejection rate of the participants with step **S2** and without step **S2**. However, since the answers to the MCQs were not available to the expert conducting the data collection, it resulted in slightly lower rejection in the non-step **S2** part of the study (8.3%) compared to the study with **S2** (8.9%). Some examples were missed during the validation but not relevant to the dialogue "The remote design conversation. It was really good at design and all art works. ", "the conversation is industrial designer and tv size and on/off settings and inderier colours and designs always", "how to improve marketing and tips and most important ideas and success project.some meterial form desidn and more collected ideas"(sic). We leave it to future work to analyze how the **S2** influences the users in providing the summaries. We also compare the ROUGE scores between the question presented to the users and the CW summaries. We found higher R-1, R-L, and BERTScore with the questions than the summaries provided by the CW, who were not shown **S2**. This shows some preliminary evidence of **S2** influencing the summaries provided. We leave further analysis of this for future work.

Comparison	**R-1**	**R-2**	**R-L**	**BERTScore**
CW (QA - No QA)	30.01	7.20	18.84	0.81
CW - Questions	31.33	5.52	20.31	0.82

Table 1: Row 1 contains the comparison between the crowd-workers who participated with QA and without QA step. Row-2 contains comparisons between the CW and the questions.

4 Data collection results

	# sessions	# chunks	Hours
# Total Dialogues	49	924	25.67
# Train dialogues	32	566	15.72
# Dev dialogue	9	191	5.31
# Test dialogues	8	161	4.64

Table 2: Shows the statistics of the data collected.

In this section, we'll describe the results from the data collection experiments. The data collection tasks can only be launched one dialogue chunk per conversation at a time. This is because the context for the current time chunk to be summarized by the user requires the past 5 minutes of summaries from other crowd-workers. This means that a dialogue chunk can be launched for the crowd-workers only if the past three dialogue chunks are summarized. The task had to be monitored for and the tasks launched in increments by a human operator as the data kept coming in. The ITS data collection took 35 days. The statistics of the data collected are shown in Table 2.

We answer the following question in this section, 'How do the summaries generated by the experts and the crowd-workers (CW) compare?'. We use human/CW evaluations and automated comparisons between the summaries generated by the expert to answer this question.

4.1 Summaries comparison

Human evaluation of summaries is a popular approach to evaluate the summaries. Such evaluations are either done by an expert or through crowd-sourcing (Iskender et al., 2020; Dang, 2006; Khashabi et al., 2021). For human evaluation of the summaries generated by a CW, we use a comparative approach similar to those used in the Genie dashboard (Khashabi et al., 2021). We wanted to ensure that the participants (evaluators, crowd-workers as raters) had listened to the conversations before they provided the ratings. The evaluators were informed that the conversation is about 'designing of the remote control'. The evaluators were first requested to listen to the conversation and write a summary in their own words. Upon writing the summary, the evaluators comparatively rated the CW and the expert-written summaries. The expert-written summaries were authored before launching the crowd data collection, and hence, the experts were not aware as to how the summaries from CW look like. We asked the evaluators to rate the summaries on Coverage, Informativeness, Fluency, and Overall score. The evaluators were presented with two summaries and were asked to choose one of these summaries across the metrics. For each of the questions, the users had to choose "Strongly prefer A", "Weakly prefer A". "No preference", "Weakly prefer B" and "Strongly prefer B". 8% of the CW evaluators were found not following the instructions or providing generic/nonsensical summaries (e.g., This was a good conversation, Very good, They are talking about remote, Good conversation etc.) or copy-pasting contents from the conversations (They were told explicitly multiple times not to do). The workers for the evaluation task were compensated $3.00 (Average time: 10

minutes, Average hourly wage: $18.00 USD).

We performed the comparison on 27 dialogue chunks (∼45 minutes of dialogue). Each of these 27 dialogue chunks was summarized by two different crowd-workers. This allowed us to compare Expert-Crowd (Expert-CW) and Crowd-Crowd (CW-CW) conditions. For these evaluations between the dialogue-chunks, we also ran ROUGE score (Lin, 2004) comparisons, treating the Expert authored summary as the reference summary. When running evaluations between Crowd workers (CW-CW), we treated one of the summaries randomly as the reference. We also use BERTScores (Zhang et al., 2019) to do compare the summaries.

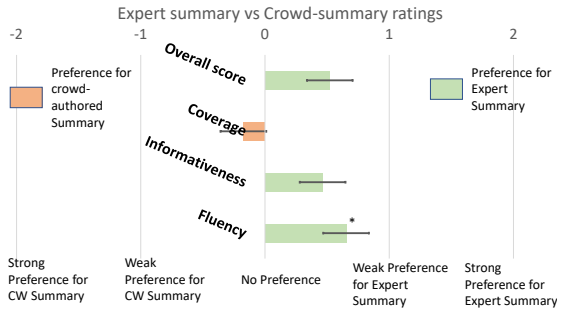

Figure 2: Shows the mean and standard error lines for the responses from the crowd evaluators. * p <0.05

Expert vs Crowd worker summaries: In the human evaluations between Expert and CW summaries, we found no 'strong' preference for either. The workers slightly preferred the expert-authored summary for their overall quality, informativeness, and fluency. The workers rated crowd-authored summaries as having slightly more coverage than the expert-authored summaries. Figure 2 shows the ratings from the evaluators. Our analysis of the One-sample t-test (mu=0) yielded no significance (p>0.05) for the overall scores indicating no major difference between the samples. Fluency scores were better for the expert-authored summaries (p<0.05). Coverage and informativeness yielded no significant difference. The average number of tokens in the crowd-authored summary (61.61) was slightly greater than the expert-authored summary (59.8). For these 27 pairs of summaries (Expert-CW, CW-CW), we then computed the ROUGE scores and performed the pairwise t-test to see if the ROUGE scores varied significantly. We found that there was no significant difference (Pairwise t-test, p>0.1) between the ROUGE scores and BERTScore for the summaries generated between crowd-workers (CW-CW) and the expert (Expert-CW). The BERTScores between the CW-CW and Expert-CW were the same up to two decimal places. Table 3 shows the result. In other words, we observed a similar variation between the summaries written by the CW when compared to other CW and the expert. This, combined with the human evaluations, seems to indicate variability in the summaries, yet no major difference in the human preferences for either of the summaries. We believe this is due to the nature of the open-ended abstractive summarization task.

Comparison	R-1	R-2	R-L	BERTScore
Expert - CW	39.86	12.15	26.56	0.88
CW - CW	38.46	13.01	28.25	0.88

Table 3: The Rouge score comparisons between the summaries by the expert and the crowd-workers are shown in Rows 1 and 2.

5 Models for summarization

We also develop models for abstractive summarization in our work. Our primary focus is on abstractive summarization for the incremental temporal scenario. The Incremental temporal summarization module takes as input the utterances in the current time window along with the past summaries (Context) to generate the summaries. However, it is not clear how important these contexts are. We thus mainly set out to answer this question as we develop the abstractive summarization models.

5.1 Abstractive summarizer

Recent advances in deep learning, such as the transformer-based models have yielded promising results in the abstractive summarization tasks. For instance, BART, Pegasus, and T5 models (Lewis et al., 2020; Zhang et al., 2020; Raffel et al., 2020) have outperformed the previous models in abstractive summarization tasks for news articles. We thus consider these 3 model architectures are the baselines for our task. We use a machine with Intel(R) Xeon(R) Platinum 8180 processor and NVIDIA(R) RTX 2080 GPU. For the models, we use the BART-large, PEGASUS-large and T5-large models from Huggingface (Wolf et al., 2019) library. We retain the default model configurations. The models can generate summaries of the max length of 142 tokens.

We then conduct experiments to answer whether these models generate better summaries if they're

provided with the past context? Hence, for each of the 3 (BART, PEGASUS, T5) models, we create 2 model variants, namely without context (no past summaries) and with human context (with summaries from the past 5 minutes of the conversation). The model architectures are the same across both conditions. We only vary the input in these two variants. In the 'without context' condition, we only input the speaker roles and the transcriptions of the extractives marked by the CW. The speakers and the transcriptions are separated by a separator token. In 'with context' condition, we additionally concatenate the past summaries of the three dialogue chunks context separated by '<EOS>' separator token.

Pre-training the models with large datasets and then finetuning the models on a smaller task-specific dataset has yielded promising results in the past for numerous tasks. It is, however, not clear if the finetuning approach will yield better models mainly due to overfitting on the smaller dataset (Aghajanyan et al., 2021). We also explore the question of whether the finetuning approach yields better results for our task. For each of the 6 model variants (BART, PEGASUS, T5 each with context and without context), we pretrain and finetune in 4 different ways, i) No pretraining (Trained only on ITS data), ii) Pre-training on CNN/Dailymail (Hermann et al., 2015; Nallapati et al., 2016) and then finetune on ITS data, iii) Pretraining on CNN/Dailymail, followed by finetuning the model on a related domain summary from non-incremental AMI corpus summaries (Carletta et al., 2005) iv) We also experiment if the 'speaker role' improves the summary compared to just the transcriptions input. In this variant, we use the same training process as in iii) but change the input during training by removing the speaker role information. Thus we compare the results from 24 models summarized in Table 4.

For training the models for abstractive summarization, we use the following configuration for all the 24 models, learning rate=0.0001, training batch size = 2, label smoothed Negative log-likelihood loss. We run the training for 25 epochs and choose the model resulting in the best R1[2].

[2]Rouge scores were calculated using the rouge-score version 0.0.4 https://pypi.org/project/rouge-score/

5.2 Results

In this section, we're interested in answering three main research questions: i) Which model architecture generates better summaries overall? ii) Does context help generate better summaries? iii) Does pre-training, and fine-tuning help improve the model performance consistently across all the conditions?

For the statistical analysis of the results from abstractive summarization models, we compare the ROUGE Recall metrics as they've been shown to be good indicators of the quality (Owczarzak et al., 2012) compared the ROUGE precision. We compare the ROUGE scores generated on the test set samples. For each dialogue-chunk we obtain the model prediction, then compute the ROUGE scores per sample across all the models for comparison. We perform the Two-way ANOVA analysis (with independent variables: Model and Pretraining method) for R-1, R-2 and R-L recall scores separately.

Which model architecture generates better summaries with better ROUGE recall for ITS task? From the Two-way ANOVA analysis, We find that there are significant differences in the model performance on R1 ($F(2,2997)=6.243$, $p=0.00197$) and R2 ($F(2,2997)= 3.848$, $p=0.0214$) recall metrics. We do not find any significant differences in models for RL metrics ($F(2,2997)=1.658$, $p=0.1907$). We run Tukey's Honestly Significant Difference (Tukey's HSD) posthoc test for pairwise comparison to further answer how models compare to each another. We find that the BART model significantly outperforms PEGASUS ($p = 0.03$) and T5 ($p = 0.001$) on R1 recall metrics. For R2, BART outperforms PEGASUS ($p = 0.01$) while there was no significant difference between BART and T5 ($p = 0.25$). For RL, we find no significant differences between the models. We also found no significant differences in R1, R2, and RL between PEGASUS and T5 models. Figure 3 shows the results. The answer to the question depends on the metrics being used to compare the results, i.e., if R1 and RL are considered, then we can expect to see better performance for the BART model.

Do models trained and inferred with context generate summaries with better recall? We then answer whether the context (during training and inference steps) helps the model generate better summaries than the models without the con-

Model		Without context			With context		
	Pre-trained data	**R1**	**R2**	**RL**	**R1**	**R2**	**RL**
BART	-	37.26/42.74	11.38/12.80	22.83/26.07	**37.70/43.82**	**12.29/14.30**	**23.06/27.48**
	CNN-DM	**39.10**/39.70	11.80/12.06	23.51/24.59	37.84/**44.12**	**12.86/14.94**	**23.70/28.19**
	CNN-DM → AMI *	33.67/**45.93**	9.46/12.81	19.72/**27.69**	**36.43**/43.17	**10.86/13.02**	**21.67**/26.43
	CNN-DM → AMI	**38.06**/39.05	11.59/11.56	**22.73**/23.99	37.57/**41.16**	**11.73/13.21**	22.63/**25.36**
Pegasus	-	40.04/39.76	12.27/11.91	25.64/25.74	**40.10/39.79**	**12.32/11.92**	**25.67/25.76**
	CNN-DM	**40.97**/37.23	**12.81**/11.53	**26.25**/24.45	37.69/**43.02**	11.84/**13.13**	23.69/**27.34**
	CNN-DM → AMI *	**40.89**/37.43	**13.07**/11.46	**26.21**/24.37	39.20/**42.16**	11.14/**12.11**	23.67/**25.88**
	CNN-DM → AMI	39.28/41.33	11.92/12.06	24.56/26.17	**39.72/41.57**	**12.23/12.76**	**24.94/26.37**
T5	-	**44.67**/36.83	**15.06**/11.98	**28.74**/23.77	39.48/**41.59**	12.11/**12.44**	25.00/**26.42**
	CNN-DM	**42.97**/38.79	**14.51**/13.01	**27.05**/24.73	40.30/**40.56**	12.27/**13.13**	24.47/**24.92**
	CNN-DM → AMI *	**42.89**/36.65	**13.61**/11.05	**27.59**/23.82	39.09/**42.41**	11.77/**12.42**	24.03/**26.36**
	CNN-DM → AMI	**42.87**/38.75	**14.52/12.30**	**26.98**/24.74	40.37/**40.61**	12.30/**12.30**	24.50/**24.92**

Table 4: Results table shows the R1, R2 and RL (Precision/Recall) scores for the 24 models evaluated. * indicates trained with no speaker information.

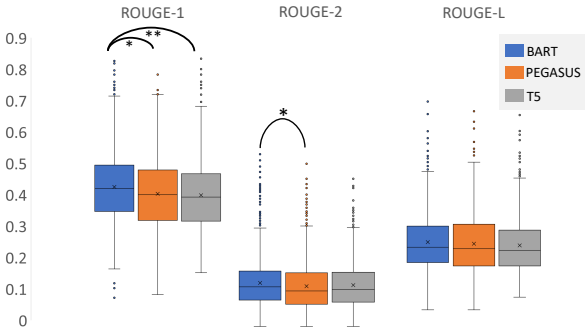

Figure 3: Shows the box plot of recall scores of the samples from the test set of all the models for model architecture comparison. (2 way ANOVA, pairwise Tukey HSD, ** p <0.01, * <0.05)

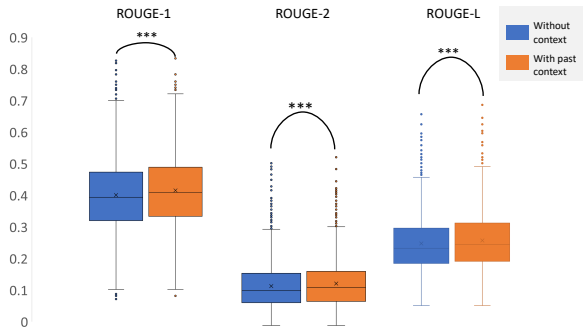

Figure 4: Shows the box plot of recall scores of the samples from the test set of all the models for context comparison. *** p <0.001

text. From Table 4, we can observe that the models, when trained with the context, perform better overall across the model architecture and different pre-training and finetuning methods. For this comparison, we take the R1, R2, and RL scores across all the models with and without context and perform an independent 2-group Mann-Whitney-U test. We found that the models with context have better recall scores for R1, R2, and RL (p <0.001). We can thus infer that the models with context as input generate summaries with better recall. Figure 4 shows the box plot of the R1, R2 and RL with and without context.

Does pre-training and fine-tuning approach yield consistent improvement across models? We found no significant differences in the R1, R2, and RL recalls resulting from the Pre-training/fine-tuning process alone. However, we found interaction effect between the models and pre-training and found significant differences between models and pre-training processes for R1 (F(4,2997)=4.923,p=0.00059) and RL (F(4,2997)=2.378, p=0.0498). This implies that the gains in performance for models resulting from the pre-training and fine-tuning procedure is different for different model architectures.

Finally, We also found that adding speaker info increases the R1 performance of recall across models (Mann-Whitney test, p=0.05). Training summarization with speaker roles (even if just concatenated with the text input) helps improve the summarization models' performance significantly.

6 Discussion & Future work

In this work, we developed a corpus for incremental temporal summarization in dialogue using crowdsourcing. We showed that our approach to collect summaries yields summaries of comparable quality to experts. The dataset also contains >5000 questions generated automatically and the answers from the crowd-workers. Recent developments in the

summarizations have developed approaches that utilize such Q-A (Question-Answer) approaches to facilitate summary generation (Guo et al., 2018; Dong et al., 2020). In this work, we use the Q-A pairs for validating the CW summaries; however, the dataset developed in this work could help facilitate the development of similar approaches for conversational summarization.

We developed models for automatic abstractive summarization and showed that models, when provided with past context summaries, helps generate better summaries. The crowd-workers in the study also indicated 94.6% times that the context helped them better understand the context of dialogue. We showed through the statistical tests that the BART model generated better summaries (measured in terms of R-1 and R-L scores) and showed that pre-training interacts with different models differently. Hence, we could not conclude that the pre-training alone will help achieve better performance. This information could benefit model builders to test different combinations of a model with the training procedures to get the best performance.

Yet another avenue for the future work is the development and evaluation of the summaries using metrics that capture the incremental nature of the summaries generated.

6.1 Extractive summarizer

In this work, until now, for the development of the abstractive summaries, we assume a perfect extractive summarizer. However, this will not be the case during the real-time scenario. Towards this, we also develop a baseline for an extractive summarizer. The extractive classifier model is a binary classification model, with 1 if the current user utterance (Transcribed user speech separated by a silence of $>$ 300 ms) is an 'extractive' i.e. if it needs to be included in the summary, 0 if it is not. We use BERT (Devlin et al., 2018) model for building the extractive summarizer. We extract the BERT embeddings and build a linear layer on top of it to create an extractive classifier. The model is the same as that described in Liu (2019). The model has a test set accuracy = 70.55%, R-1 (recall) = 38.19, R-1 (Precision) = 82.19, R-2 (recall) = 31.59, R-2 (Precision) = 70.92, R-L (recall) = 28.92, R-L (Precision) = 61.91 For future work, we aim to integrate the extractive summarizer and develop models, especially incremental multi-modal models for ITS that could help with the summarization tasks. Integrating the information as the information evolves is an interesting area for future work that corpus supports.

Acknowledgements

We want to thank Maike Paetzel-Prüsmann (University of Potsdam) and Nese Alyuz Civitci (Intel labs) for their feedback on statistical tests. We want to thank John Sherry for his feedback at various stages of the project. We also wish to thank anonymous reviewers for their helpful comments and feedback.

References

Armen Aghajanyan, Akshat Shrivastava, Anchit Gupta, Naman Goyal, Luke Zettlemoyer, and Sonal Gupta. 2021. Better fine-tuning by reducing representational collapse. In *International Conference on Learning Representations*.

Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. 2019. Synthetic qa corpora generation with roundtrip consistency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6168–6173.

Javed Aslam, Fernando Diaz, Matthew Ekstrand-Abueg, Richard McCreadie, Virgil Pavlu, and Tetsuya Sakai. 2015. Trec 2014 temporal summarization track overview. Technical report, NATIONAL INST OF STANDARDS AND TECHNOLOGY GAITHERSBURG MD.

Emma Barker, Monica Lestari Paramita, Ahmet Aker, Emina Kurtić, Mark Hepple, and Robert Gaizauskas. 2016. The sensei annotated corpus: Human summaries of reader comment conversations in on-line news. In *Proceedings of the 17th annual meeting of the special interest group on discourse and dialogue*, pages 42–52.

Claire Bonial, Olga Babko-Malaya, Jinho D Choi, Jena Hwang, and Martha Palmer. 2010. Propbank annotation guidelines. *Center for Computational Language and Education Research Institute of Cognitive Science University of Colorado at Boulder*.

Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, et al. 2005. The ami meeting corpus: A pre-announcement. In *International workshop on machine learning for multimodal interaction*, pages 28–39. Springer.

Ying-Hong Chan and Yao-Chung Fan. 2019. A recurrent bert-based model for question generation. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 154–162.

Hoa Trang Dang. 2006. Duc 2005: Evaluation of question-focused summarization systems. In *Proceedings of the Workshop on Task-Focused Summarization and Question Answering*, pages 48–55.

Hoa Trang Dang and Karolina Owczarzak. 2008. Overview of the tac 2008 update summarization task. In *TAC*.

David DeVault, Kenji Sagae, and David Traum. 2011. Incremental interpretation and prediction of utterance meaning for interactive dialogue. *Dialogue & Discourse*, 2(1):143–170.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Kaustubh Dhole and Christopher D Manning. 2020. Syn-qg: Syntactic and shallow semantic rules for question generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 752–765.

Yue Dong, Shuohang Wang, Zhe Gan, Yu Cheng, Jackie Chi Kit Cheung, and Jingjing Liu. 2020. Multi-fact correction in abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9320–9331.

Florian Eyben, Martin Wöllmer, and Björn Schuller. 2010. opensmile - the munich versatile and fast open-source audio feature extractor. In *Proc. ACM Multimedia (MM), ACM, Florence, Italy, 2010*, pages 1459–1462.

Tobias Falke and Iryna Gurevych. 2017. Bringing structure into summaries: Crowdsourcing a benchmark corpus of concept maps. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2951–2961.

Benoit Favre, Evgeny Stepanov, Jérémy Trione, Frédéric Béchet, and Giuseppe Riccardi. 2015. Call centre conversation summarization: A pilot task at multiling 2015. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 232–236.

Liz Fosslien and Mollie West Duffy. 2020. How to combat zoom fatigue. *Harvard Business Review*, 29.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. Allennlp: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6.

Bogdan Gliwa, Iwona Mochol, Maciej Biesek, and Aleksander Wawer. 2019. Samsum corpus: A human-annotated dialogue dataset for abstractive summarization. *EMNLP-IJCNLP 2019*, page 70.

Han Guo, Ramakanth Pasunuru, and Mohit Bansal. 2018. Soft layer-specific multi-task summarization with entailment and question generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 687–697.

KM Hermann, T Kočiskỳ, E Grefenstette, L Espeholt, W Kay, M Suleyman, and P Blunsom. 2015. Teaching machines to read and comprehend. *Advances in Neural Information Processing Systems*, 28.

Neslihan Iskender, Tim Polzehl, and Sebastian Möller. 2020. Towards a reliable and robust methodology for crowd-based subjective quality assessment of query-based extractive text summarization. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 245–253.

Adam Janin, Don Baron, Jane Edwards, Dan Ellis, David Gelbart, Nelson Morgan, Barbara Peskin, Thilo Pfau, Elizabeth Shriberg, Andreas Stolcke, et al. 2003. The icsi meeting corpus. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).*, volume 1, pages I–I. IEEE.

Youxuan Jiang, Catherine Finegan-Dollak, Jonathan K Kummerfeld, and Walter Lasecki. 2018. Effective crowdsourcing for a new type of summarization task. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 628–633.

Taehee Jung, Dongyeop Kang, Lucas Mentch, and Eduard Hovy. 2019. Earlier isn't always better: Sub-aspect analysis on corpus and system biases in summarization. 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 3315–3326.

Daniel Khashabi, Gabriel Stanovsky, Jonathan Bragg, Nicholas Lourie, Jungo Kasai, Yejin Choi, Noah A Smith, and Daniel S Weld. 2021. Genie: A leaderboard for human-in-the-loop evaluation of text generation. *arXiv preprint arXiv:2101.06561*.

Ronilda C Lacson, Regina Barzilay, and William J Long. 2006. Automatic analysis of medical dialogue in the home hemodialysis domain: structure induction and summarization. *Journal of biomedical informatics*, 39(5):541–555.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Manling Li, Lingyu Zhang, Heng Ji, and Richard J Radke. 2019. Keep meeting summaries on topic: Abstractive multi-modal meeting summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2190–2196.

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

Chunyi Liu, Peng Wang, Jiang Xu, Zang Li, and Jieping Ye. 2019a. Automatic dialogue summary generation for customer service. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1957–1965.

Yang Liu. 2019. Fine-tune bert for extractive summarization. *arXiv preprint arXiv:1903.10318*.

Zhengyuan Liu, Angela Ng, Sheldon Lee, Ai Ti Aw, and Nancy F Chen. 2019b. Topic-aware pointer-generator networks for summarizing spoken conversations. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 814–821. IEEE.

Elena Lloret, Laura Plaza, and Ahmet Aker. 2013. Analyzing the capabilities of crowdsourcing services for text summarization. *Language resources and evaluation*, 47(2):337–369.

Luis Enrico Lopez, Diane Kathryn Cruz, Jan Christian Blaise Cruz, and C. Cheng. 2020. Transformer-based end-to-end question generation. *ArXiv*, abs/2005.01107.

Richard McCreadie, Craig Macdonald, and Iadh Ounis. 2014. Incremental update summarization: Adaptive sentence selection based on prevalence and novelty. In *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*, pages 301–310.

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.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290.

Jessica Ouyang, Serina Chang, and Kathleen McKeown. 2017. Crowd-sourced iterative annotation for narrative summarization corpora. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 46–51.

Karolina Owczarzak, John Conroy, Hoa Trang Dang, and Ani Nenkova. 2012. An assessment of the accuracy of automatic evaluation in summarization. In *Proceedings of workshop on evaluation metrics and system comparison for automatic summarization*, pages 1–9.

Erik Peper, Vietta Wilson, Marc Martin, Erik Rosegard, and Richard Harvey. 2021. Avoid zoom fatigue, be present and learn. *NeuroRegulation*, 8(1):47–47.

Massimo Poesio and Hannes Rieser. 2010. Completions, coordination, and alignment in dialogue. *Dialogue & Discourse*, 1(1).

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*, 21:1–67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *EMNLP*.

David Schlangen and Gabriel Skantze. 2011. A general, abstract model of incremental dialogue processing. *Dialogue & Discourse*, 2(1):83–111.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

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.

A Appendix

A.1 Prosodic Features

The dataset also contains prosodic features for each utterance. We extracted the 1582-dimensional audio prosodic feature embedding representations for all the 100s audio chunks of the dataset using openSMILE toolkit (Eyben et al., 2010). We randomly selected 500 embeddings and plotted them

in t-SNE two-dimensional space. The red '*' dots in Figure 5 are representing extracted utterances for the summaries, and the green '+' dots are representing the utterances that were not extracted. The figure shows that the two extractive classes could have a reasonable linear separation by the prosodic features related to emotion recognition, which indicates and agrees with the intuitive assumption that the extracted utterances for the summaries are the more emotional utterances in the conversations.

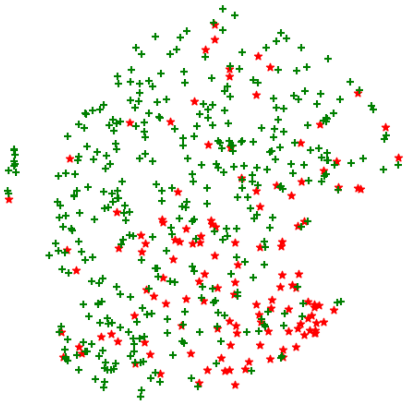

Figure 5: Prosodic feature embeddings for the audio chunks: red '*' dots are extracted utterances; green '+' dots are utterances not extracted.

A.2 Pay for Turker

To decide the pay, the task was simulated with 2 users for an entire dialogue and the time taken was recorded. The users had domain knowledge. We then doubled our time estimate for the crowdworker and deployed the task on MTurk. For each data collection task for a dialogue chunk of 100 seconds, we compensated the workers $3.00 USD (Approx. $20 USD per hour). No limitation was placed on the number of times the users could participate. Hence, their average pay increased more they participated [3]. The participants were informed of the task at every step and the expectations were clearly mentioned. The development of the data collection interface was iterative and the data collected during the development of the interface was discarded.

[3]Highest amount earned was equivalent of $54 per hour.

A.3 R-1 comparisons for models and pretraining

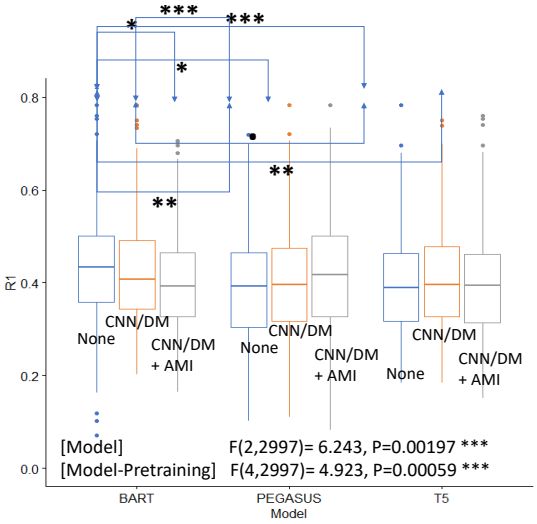

Figure 6: Shows the ROUGE recall scores of the samples from the test set of all the models resulting from pretraining. p <0.001