TWEETSUMM - A Dialog Summarization Dataset for Customer Service

Guy Feigenblat[*,†], **Chulaka Gunasekara**[*], **Benjamin Sznajder**[*], **Ranit Aaronov**,
David Konopnicki, Sachindra Joshi
IBM Research AI
{guyf@il, chulaka.gunasekara@, benjams@il}.ibm.com
{ranit.aharonov2@, davidko@il[‡], jsachind@in}.ibm.com

Abstract

In a typical customer service chat scenario, customers contact a support center to ask for help or raise complaints, and human agents try to solve the issues. In most cases, at the end of the conversation, agents are asked to write a short summary emphasizing the problem and the proposed solution, usually for the benefit of other agents that may have to deal with the same customer or issue. The goal of the present article is advancing the automation of this task. We introduce the first large scale, high quality, customer care dialog summarization dataset with close to 6500 human annotated summaries. The data is based on real-world customer support dialogs and includes both extractive and abstractive summaries. We also introduce a new unsupervised, extractive summarization method specific to dialogs.

1 Introduction

Text summarization is the task of creating a short version of a long text, retaining the most important or relevant information. In NLP, there are two types of summarization tasks- (1) Extractive summarization, in which segments from the original text are selected to form a summary and (2) Abstractive summarization, in which new natural language expressions are generated for summarizing the text. The past few years have witnessed a tremendous progress in creating both kinds of summaries using seq2seq models. However, these works have largely focused on documents such as news and scientific publications (Lin and Ng, 2019).

In this paper, we focus on summarizing conversational data between customers and human support agents. In many enterprises, once an agent is done with handling a customer request, she is required to create a short summary of the conversation for record keeping purposes. At times, an ongoing conversation may also need to be transferred to another agent or escalated to a supervisor. This also requires creating a short summary of the conversation so far, as to provide the right context to the next handling agent.

Our main contribution is the release of TWEETSUMM, a dataset focused on summarization of dialogs, which represents the rich domain of Twitter customer care conversations [1]. The dataset contains close to 6500 extractive and abstractive summaries generated by human annotators from 1100 dialogs. This is the first dataset released to the research community, which focuses on real dialogs, as opposed to previous works focusing on meeting conversations (McCowan et al., 2005), general *chitchat* summarization (Gliwa et al., 2019), or topic descriptions of interviews (Zhu et al., 2021). Furthermore, the fact that each dialog was annotated by 3 different crowd-workers, resulting in an overall of 6 summaries for each dialog, provides diversity of summaries. We performed quality control and assessment to remove erroneous summaries, and to ensure that the collected TWEETSUMM summaries are of a high quality. We evaluate several summarization baselines and further provide a novel unsupervised extractive summarization algorithm, referred to as *NRP Summ* which outperforms other unsupervised baselines for extractive summarization. Figure 2 shows an example of a TWEETSUMM dialog along with a human-generated abstractive summary and two machine-generated summaries - abstractive and extractive summaries. We propose that the dataset quality and scale, is suitable for developing future models for the dialog summarization task. We hope that releasing TWEETSUMM for the community will foster further research.

[*] With equal contribution
[†] Current address: guy@piiano.com
[‡] Current address: david.konopnicki@booking.com

[1] https://github.com/guyfe/Tweetsumm

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 245–260
November 7–11, 2021.

Original dialog	
Customer	@Company flight1234 from Miami to LaGuardia smells awful. We just boarded. It's really really bad.
Agent	@Customer_id Allie, I am very sorry about this. Please reach out to a flight attendant to address the odor in the aircraft. *TBW
Customer	@Company They're saying it game in from the last flight. They have sprayed and there's nothing else they can do. It's gross!
Agent	@Customer_id I'm very sorry about the discomfort this has caused you for your flight! *TBW
Customer	@Company It's not just me! Every person getting on the flight is complaining. The smell is horrific.
Agent	@Customer_id Oh no, Allie. That's not what we want to hear. Please seek for one of our crew members on duty for further immediate assistance regarding this issue. Please accept our sincere apologies. *AOS
Customer	@Company They've brought maintenance aboard. Not a great first class experience :(
Agent	@Customer_id We are genuinely sorry to hear about your disappointment, Allie. Hopefully, our maintenance crew can fix the issue very soon. Once again please accept our sincere apologies for this terrible incident. *AOS
Customer	@Company Appreciate it. Thank you!
Agent	@Customer_id You are most welcome, Allie. Thanks for tweeting us today. *AOS
Customer	@Company They told us to rebook, then told us the original flight was still departing. We got put back on 1234 but are now in the 1st row instead of the 3rd. Can you get us back in seats 3C and 3D?
Customer	@Company My boyfriend is 6feet tall and can't sit comfortably at the bulkhead.
Agent	@Customer_id Unfortunately, our First Class Cabin is full on our 1234 flight for today, Allie. You may seek further assistance by reaching out to one of our in-flight crew members on duty. *AOS
Ground truth (human) abstractive summary	
	Customer complains about smell in flight. Agent updated the customer to seek further assistance by reaching out to one of their in-flight crew members on duty.
Automated abstractive summary	
	Customer is complaining about bad smell in his flight. Agent informed to contact in-flight crew member on duty for further assistance.
Automated extractive summary	
Customer	Flight1234 from Miami to LaGuardia smells awful.They told us to rebook, then told us the original flight was still departing.
Agent	Unfortunately, our First Class Cabin is full on our 1234 flight for today, Allie. You may seek further assistance by reaching out to one of our in-flight crew members on duty.

Figure 1: TWEETSUMM dialog and its summaries

2 TWEETSUMM Dataset

TWEETSUMM comprises of 1100 dialogs reconstructed from Tweets that appear in the *Kaggle Customer Support On Twitter* dataset[2], each accompanied by 3 extractive and 3 abstractive summaries generated by human annotators. The Kaggle dataset, is a large scale dataset based on conversations between consumers and customer support agents on *Twitter.com* (Hardalov et al., 2018). It covers a wide range of topics and services provided by various companies, from airlines to retail, gaming, music etc. Thus, TWEETSUMM can serve as a dataset for training and evaluating summarization models for a wide range of dialog scenarios.

For creating the 1100 dialogs of TWEETSUMM, we first reconstructed 49,155 unique dialogs from the *Kaggle Customer Support On Twitter* dataset (see section 2.1). Second, we filtered short and long dialogs, containing less than 6 or more than 20 utterances, in order to focus on dialogs that are representative of most cases. This resulted in 45,547 dialogs with an average length of 22 sentences[3]. Next, in order to represent the customer service scenario, in which a single customer interacts with a single agent, dialogs with more than two speakers were removed. From the remaining 32,081 dialogs, we randomly sampled 1100 dialogs. These dialogs were sent for generation of summaries using crowdsourcing on the *Appen.com* platform, as described below.

2.1 Dialog Reconstruction Method

The data is delivered via a CSV file where each record contains the following fields: $text$ - the anonymized text of the Tweet, $tweet\_id$ - unique anonymized Tweet ID, $author\_id$ - unique anonymized author ID, $inbound$ - whether the Tweet is to or from a company, $response\_tweet\_id$ - IDs of Tweets that are responses to this Tweet, $in\_response\_to\_tweet\_id$ - ID of the Tweet this Tweet is in response to, and $created\_at$ - date and time the Tweet was sent.

In order to reconstruct dialogs from Tweets, we traversed the CSV data recursively using the $in\_response\_to\_tweet\_id$ field. At the end of this process, each dialog is a sorted list of Tweets and their metadata fields. In case several Tweets are posted as response to the same Tweet, they are sorted by their $created\_at$ timesamp. This often happens when a message exceeds the length limit for a single Tweet, and has to be split.

2.2 Summaries Generation

Each annotator was asked to generate one extractive and one abstractive summary for a single dialog at a time. When generating the extractive summary, the annotators were instructed to highlight the most salient sentences in the dialog. For the abstractive summaries, they were instructed to write a summary that contains one sentence summarizing what the customer conveyed and a second sentence summarizing what the agent responded. See the supplementary material for a detailed description of the instructions provided to annotators before starting the task. We collected 3 annotations per dialog, such that overall we obtained $\approx$ 6600 summaries: $\approx$ 3300 extractive summaries, termed hereafter the **extractive dataset** and $\approx$ 3300 abstractive summaries, termed hereafter the **abstractive dataset**. As explained in the next section, some summaries were discarded following quality control, and for some dialogs, a second round of summaries collection was done. Overall, TWEETSUMM contains 3056 extractive and 3327 abstractive summaries.

[2] www.kaggle.com/thoughtvector/customer-support-on-twitter

[3] An utterance, sometimes termed turn, usually contains more than one sentence.

2.3 Quality Control and Assessment

2.3.1 Quality Control

To guarantee a high quality level of annotations, multiple measures were taken in advance. We only recruited as crowd-workers, members of an Expert Business Partner channel, who are fluent English speakers. Before an annotator was approved for the task, he or she had to pass a quality control test by annotating 10 dialogs with an acceptable high quality. The quality of those summaries was checked manually. Out of 25 annotators who participated in the test only 10 were approved for the task.

Following completion of the task, several heuristics were applied to identify and discard bad extractive summaries, and statistics were kept on annotators to identify those, if any, that produced erroneous summaries with high frequency. The applied heuristics included removing summaries containing only one sentence, summaries containing only one side (Customer-only or Agent-only), or summaries starting by an Agent turn. We remove summaries starting by an agent turn since tweeter dialogs begin by a customer raising an issue, and hence the summary is expected to begin with a customer turn. By these cleansing steps, we removed from our dataset 286 extractive summaries. None of the annotators exhibited a high frequency of such bad summaries, supporting the assumption that these errors are due to technical annotation problems, such as erroneously pressing submit prematurely, rather than an annotator performing poorly on the task in general.

To further assure the quality of the summaries, we computed on each document and for each annotator the percentage of his selected sentences which were also selected by one of the other annotators. A classical Jacquard score would result in irrelevant low-scores if one of the other annotators selected a large number of sentences, and, thus, we used a slightly adapted version $J{=}|A \cap B|/|A|$ which *punishes* A if he selected a less concise summary. No annotator got an extreme low score and the average scores of the annotators range from 50% to 68%. For extra safety, we manually checked the summaries with low J scores and found that they do not appear to be unequivocally erroneous. Rather, the difference in the selection of the sentences was due to similar sentences in the original dialog and to the inherent subjectivity of the task, which is also consistent with previous research (Daume III and Marcu, 2005)

Summary Type	*Question*	*Average score*
extractive	Provide your rating as to the overall **coverage** of the summary, based on how well it represents important information from the dialog	4.03 ($\pm$0.77)
abstractive	Provide your rating as to the overall **coverage** of the summary, based on how well it represents important information from the dialog	3.96 ($\pm$0.84)
	Provide your rating as to the **readability** of the summary. Please consider fluency, grammatical correctness, and coherence	4.22 ($\pm$0.61)

Table 1: Results of the Quality Assessment

In addition, we looked for cases where annotators used a repeating, or closely-repeating, text for abstrative summaries of different dialogs. We have identified only 9 such abstractive summaries, which were discarded from the dataset.

2.3.2 Quality Assessment

We also used annotators to assess the quality of the summaries generated for TWEETSUMM. To achieve a high quality standard we recruited NLP experts instead of using the same pool of crowd-workers that worked on the summaries generation task. The annotators were instructed to read the dialog carefully and to select a rating between 1 (lowest score) to 5 (highest score) as an answer to three questions focusing on summary *Coverage* and *Readability*. To this end, 100 pairs of extractive and abstractive summaries from different dialogs were randomly sampled from TWEETSUMM, with 3 experts working on each summary. The obtained median score for all 3 questions is 4, with average ratings ranging between 3.96-4.22. The questions that were asked along with their average scores and std, are described in Table 1. In order to evaluate the reliability of this assessment, we followed the approach suggested by (Toledo et al., 2019) to measure agreement between the 3 annotators over ordinal ratings, by reporting average Kappa values among the possible combinations of two annotators. For the extractive and abstractive Coverage questions, the obtained Kappa scores are 0.41 and 0.56 respectively. For the abstractive Readability question the obtained Kappa score is 0.36. While not perfect, the obtained Kappa values are expected due to the inherent subjectivity of the summarization task, as backed up by previous research (Daume III and Marcu, 2005).

We thus conclude, based on our quality control and assessment, that the TWEETSUMM **dataset contains high quality summaries generated by high quality annotators.**

	Full dialog	*Customer utterances*	*Agent utterances*
#utterances	10.17(±2.31)	5.48(±1.84)	4.69(±1.39)
#sentences	22(±6.56)	10.23(±4.83)	11.75(±4.44)
#tokens	245.01(±79.16)	125.61(±63.94)	119.40(±46.73)

Table 2: Average lengths of dialogs

2.4 Dataset Analysis

Table 2 details the average length of the dialogs in TWEETSUMM, including the average lengths of the customer and agent utterences. The average length of the summaries is reported in Table 3. Comparing the dialog lengths to the summaries lengths indicates the average compression rate of the summaries. For instance, on average, the abstractive summaries compression rate is 85% (i.e. the number of tokens is reduced by 85%), while the extractive summaries compression rate is 70%. The number of customer and agent sentences selected in the extractive summaries were relatively equally distributed with 7445 customer sentences and 7844 agent sentences in total.

	Overall	*Customer*	*Agent*
Abstractive	36.41(±12.97)	16.89(±7.23)	19.52(±8.27)
Extractive	73.57(±28.80)	35.59(±21.3)	35.80(±18.67)

Table 3: Average lengths (in # tokens) of summaries

Next, the positions of the sentences selected for the extractive summaries were analyzed. In 85% of the cases, sentences from the first customer utterance were selected, compared to 52% of the cases in which sentences from the first agent utterances were selected. This corroborates the intuition that customers immediately express their need in a typical customer service scenario, while agents do not immediately provide the needed answer: agents typically greet the customer, express empathy, and ask clarification questions. For the abstractive summaries, inherently, the utterance from which annotators selected information cannot be directly deduced, but can be approximated. Following (Nallapati et al., 2017), for each abstractive summary, we evaluated the ROUGE distance (using ROUGE-L Recall) between the agent (resp. customer) part of the summary, with each of the actual agent (resp. customer) utterances in the original dialog. We then considered the utterance with the maximal score to be the utterance from which the summary is mainly based-on. By averaging over all the dialogs, we obtained that 75% of the customer summary part are based-on the first customer utterance vs. only 12% of the agent's part.

3 Next Response Prediction Summarizer

We introduce a novel, unsupervised extractive summarization method (coined *NRP Summ*) aimed at identifying the sentences that influence the entire dialog the most.

The Next Response Prediction Model - To identify the influence of each sentence on the entire conversation, we utilize the next response prediction (NRP) task (Gunasekara et al., 2019) in dialog systems. The NRP task is defined as follows: given a dialog context, i.e., the list of sentences in the dialog up to a certain point ($C = \{s_1, s_2, ..., s_k\}$), predict the next response sentence (c_r) from a given set of candidates $\{c_1, ..., c_r, ..., c_n\}$. To train the NRP model, we used a binary classifier commonly used for GLUE tasks (Wang et al., 2018). We process the dialogs to construct triples of <*dialog context (C), candidate (c_i), label* (1/0)> from each dialog context. For each C, we create a set of $k + 1$ (k=5 in this study) triples: one triple containing the correct response (c_r) (label=1), and k triples containing incorrect responses randomly sampled from the dataset (label=0). The dialog context C and a candidate response c_i are fed together to BERT as a sequence ([CLS] C [SEP] c_i [SEP]). The hidden state of the [CLS] token was used as the representation of the pair. Training is done using positive and negative examples with cross-entropy loss. A model trained on the NRP task associates a probability (p_r) for the response (c_r), given the context C. We trained two NRP models, (1) a model predicting the next response given the prior sentences *(NRP-FW)*, and (2) a model predicting the prior utterance given subsequent utterances *(NRP-BW)*.

Salient sentence identification- The intuition behind this approach is that the removal of the critical sentences from a dialog context will entail a larger drop in probability in predicting a subsequent and prior responses. We follow the hypothesis that the critical sentences for the NRP task will also be salient sentences for the summary. The sentence removal occurs in two steps. In the initial step, we feed the entire context to the NRP model and identify the probability of predicting the next (or prior) sentence. In the next step, we remove one sentence at a time from the context, and input the new context to the NRP model and identify the probability of predicting the same next (or prior) utterance. Then, we assign the drop in probability as a score to the removed sentence.

To identify the salient sentences in predicting the next response, we remove one sentence at a time from the dialog context ($C \backslash s_i$) and use that as the input to a trained *NRP-FW* model and identify the probability (p_r^{fw}) for the corresponding response (c_r). Then, we assign the drop in probability ($p_r - p_r^{fw}$) as a score to the removed sentence s_i in the context. We follow the same process to identify the drop in probability in predicting the prior sentence, given the same dialog context and masked sentence (using *NRP-BW* model), and assign that as another score for the masked sentence. The averaged score for each sentence is used during salient sentence identification. For the evaluation, we use the top two customer sentences and the two top agent sentences as the extractive summary of the dialog.

4 Experiments and Results

We aim to confirm that TWEETSUMM is suitable as a ground-truth dataset for the dialog summarization task. To this end, we apply and analyze several baseline summarization models as well as *NRP Summ*, to the dataset, as detailed below. We randomly split the dialogs and their associated summaries into three sets: 80% for the training set, 10% for the validation and the rest 10%, for the test set.

4.1 Baselines

The baselines evaluated as part of this study are:
Random (extractive) - Two random sentences from the agent utterances and two from the customer utterances.
LEAD-4 (extractive) - The first two sentences from the agent utterances and the first two from the customer utterances. This approach is considered a very competitive baseline (see (Kryscinski et al., 2019) when considering news summarization).
LexRank (extractive) - This unsupervised summarizer (Erkan and Radev, 2004) casts the summarization problem into a fully connected graph, in which nodes represent sentences and edges represent similarity between two sentences. Pair-wise similarity is measured over the bag-of-words representation of the two sentences. Then, *PowerMethod* is applied on the graph, yielding a centrality score for each sentence. We take the two top central customer and agent sentences (2+2).
Cross Entropy Summarizer (extractive)- *CES* is an unsupervised, extractive summarizer (Roitman et al., 2020; Feigenblat et al., 2017), which considers the summarization problem as a multi-criteria optimization over the sentences space, where several summary quality objectives are considered. The aim is to select a subset of sentences optimizing these quality objectives. The selection runs in an iterative fashion: in each iteration, a subset of sentences is sampled over a learned distribution and evaluated against quality objectives. We introduced some minor tuning to the original algorithm, to suit dialog summarization. First, query quality objectives were removed since we focus on generic summarization. Then, since dialog sentences tend to be relatively short, when measuring the coverage objective, each sentence was expanded with the two most similar sentences, using Bhattacharyya similarity. Finally, Lex-Rank centrality scores were used as an additional quality objective, by averaging the centrality scores of sentences in a sample.
***PreSumm* (extractive/abstractive)** - This model (Liu and Lapata, 2019b) applies BERT (Devlin et al., 2019) for text summarization in both extractive and abstractive settings. In the extractive setting, *PreSumm* treats the summarization task as a sentence classification problem: a neural encoder creates sentence representations and a classifier predicts which sentences should be selected for the summary. We used a pre-trained model[4] and fine-tuned the model using the TWEETSUMM. In the abstractive setting, the model uses the same encoder as the extractive model while the decoder is a 6-layered Transformer initialized randomly.
BART (abstractive) - A denoising autoencoder (Lewis et al., 2019) that uses the seq2seq transformer architecture. It consists of two parts: an encoder and a decoder. The encoder is a bidirectional encoder which corresponds to the structure of BERT, and the decoder is an auto-regressive decoder following the settings of GPT (Radford et al., 2019). We use a lightweight variant of *BART* (coined *DistilBART*) that is fine-tuned on the XSum task (Narayan et al., 2018b). We further fine-tuned the model using the TWEETSUMM. Different variants of the BART model that were evaluated are discussed in the results section. The hyper-parameters are described in the supplemental material.

4.2 Automatic Evaluation

We first use automatic measures to evaluate the summaries generated by the models described above, using the reference summaries of TWEETSUMM. We measured summarization quality using

[4] `https://github.com/nlpyang/PreSumm`

Table 4: ROUGE F-Measure evaluation on the test set, supervised baselines are marked with †

Length Limit	Method Name	R-1	R-2	R-SU4	R-L
	Abstractive Dataset				
35 tokens	*Random*	22.970	6.370	8.340	20.601
	Lead	26.666	10.098	11.690	24.360
	LexRank	27.661	10.448	12.249	24.900
	CES	29.105	11.483	13.344	26.281
	NRP Summ	**30.197**	**12.219**	**13.911**	**27.111**
	BART - without fine-tuning	20.365	4.110	6.188	16.019
	PreSumm extractive †	30.821	12.972	14.633	27.909
	PreSumm abstractive †	33.468	9.284	13.115	31.003
	BART - without ext †	36.395	18.015	18.346	32.280
	BART - with ext †	**38.237**	**19.449**	**19.594**	**33.818**
70 tokens	*Random*	26.930	8.870	10.980	24.337
	Lead	28.913	11.489	13.053	26.395
	LexRank	30.457	12.379	14.202	27.486
	CES	**31.465**	13.152	**14.954**	**28.464**
	NRP Summ	31.416	**17.365**	14.043	27.623
	BART - without fine-tuning	20.378	4.127	6.200	16.028
	PreSumm extractive †	33.220	14.288	15.986	30.305
	PreSumm abstractive †	33.010	9.493	12.974	30.667
	BART - without ext †	36.076	17.844	18.161	31.939
	BART - with ext †	**37.938**	**19.263**	**19.417**	**33.508**
unlimited	*Random*	26.865	8.848	10.946	24.269
	Lead	29.061	11.560	13.106	26.470
	LexRank	30.459	12.652	14.423	27.563
	CES	**31.569**	13.334	15.118	**28.552**
	NRP Summ	31.209	**17.265**	**17.956**	28.541
	BART - without fine-tuning	20.378	4.127	6.200	16.028
	PreSumm extractive †	32.815	14.149	15.799	30.026
	PreSumm abstractive †	33.001	9.494	12.971	30.650
	BART - without ext †	36.076	17.844	18.161	31.939
	BART - with ext †	**37.938**	**19.263**	**19.417**	**33.508**
	Extractive Dataset				
35 tokens	*Random*	32.761	17.843	17.794	30.518
	Lead	53.156	42.944	40.549	52.045
	LexRank	48.584	36.758	36.125	46.847
	CES	55.328	45.032	43.841	54.182
	NRP Summ	**58.410**	**49.490**	**47.404**	**57.428**
	PreSumm extractive †	**60.957**	**52.478**	**50.908**	**60.142**
70 tokens	*Random*	47.868	32.978	32.693	46.035
	Lead	57.491	47.199	45.388	56.531
	LexRank	55.773	43.365	42.563	54.290
	CES	58.984	47.713	46.387	57.889
	NRP Summ	**61.114**	**51.381**	**49.558**	**60.292**
	PreSumm extractive †	**65.158**	**55.813**	**53.517**	**64.370**
unlimited	*Random*	48.943	35.074	34.548	47.333
	Lead	54.995	44.425	42.796	53.943
	LexRank	57.018	45.332	44.459	55.772
	CES	59.872	49.126	47.722	58.874
	NRP Summ	**62.971**	**55.411**	**54.614**	**62.596**
	PreSumm extractive †	**65.659**	**56.628**	**54.327**	**64.943**

the ROUGE measure (Lin, 2004) compared to the ground truth. We use the official toolkit with its standard parameters setting[5]. For the limited length variants, we run ROUGE with its limited length constraint. Table 4 reports ROUGE F-Measure results. We evaluate all summarization models (extractive and abstractive, where the extractive summarizers are set to extract 4 sentences) against the abstractive and extractive datasets. Supervised baselines are marked with the † symbol. Based on the average length of the summaries, reported in Table 3, we evaluate ROUGE with three length limits: 35 tokens (the average length of the abstractive summaries), 70 tokens (the average length of the extractive summaries) and *unlimited*. Below we discuss these results in detail.

[5] ROUGE-1.5.5.pl -a -c 95 -m -n 2 -2 4 -u -p 0.5

4.2.1 TweetSumm Abstractive Dataset

Quality of extractive summarization models- We start by analyzing how well extractive summarization models perform on the abstractive reference summaries. As described in Table 4, we note that in most cases, except 70 tokens summary, *NRP Summ* outperforms other unsupervised, extractive baselines. Interestingly, the performance of the simple *Lead-4* baseline is not far from that of the more complex unsupervised baselines. For instance, considering the 70 tokens results of the abstractive dataset, *LexRank* outperforms *Lead-4* by only 4%-8%. This is backed up by the statistics we report in section 2.4, namely that salient content conveyed by the customer appears at the beginning of the dialog. To rule out any potential overfitting, we also present results of the unsupervised, extractive, summarizers against the validation set. Table 5 shows a similar trend: in most cases, *NRP Summ* outperforms other models.

Quality of abstractive summarization models- We analyze three variants of the BART model: **(1)** *BART* with no fine-tuning on TweetSumm (*BART-without-fine-tuning*), **(2)** BART fine-tuned on TweetSumm (*BART-without-ext*), **(3)** BART fine-tuned on TweetSumm with the extractive summary provided as input in addition to the dialog (*BART-with-ext*). For training the *BART-with-ext*, the ground truth extractive summaries were appended to the dialog (with a dedicated separator). For validation and testing, the extractive summaries generated by the *NRP Summ* model were used. All BART models were pre-trained on the XSum summarization dataset (Narayan et al., 2018a) (see the specific system models settings in the supplemental material). As described in Table 4, the BART models fine-tuned on TweetSumm obtain the best results by far, compared to all other models. *BART-without-fine-tuning* model performs poorly, compared to all the other models. **From this analysis we learn that, pre-training on the general summarization task is not sufficient, fine-tuning is required to help the model learn the specifics of the dialog summarization task**. Interestingly, *BART-with-ext* outperforms *BART-without-ext*, suggesting that the extractive summary helps the model to attend to salient content. Although the PreSumm model was also similarly fine-tuned on TweetSumm, its performance is inferior to BART.

4.2.2 TweetSumm Extractive Dataset

Here we focus on evaluating the extractive summarizaion models on the extractive dataset. We first note that the average length of ground truth extractive summaries in TweetSumm is 4 sentences out of 22 sentences, on average, in a dialog. The lower compression rate of the extractive summaries compared to the abstractive summaries leads to higher ROUGE scores of the extractive summaries. The *NRP Summ* model outperforms all unsupervised methods, while the supervised *PreSumm extractive* model outperforms all other models.

4.3 Human Evaluation

We conducted two human evaluation studies to assess the quality of the summarization models. The first focuses on the **Informativeness** and **Saliency** of the summaries generated by the models. Following (Liu and Lapata, 2019c,a), we used the QA paradigm to test whether the summarization models retain key information. We chose to evaluate the two abstractive models *BART-without ext* and *PreSumm-abs* and four extractive models - *NRP Summ*, *CES*, *PreSumm-ext* and *LEAD* (limited to 4 sentences). We randomly selected 20 dialogs and recruited 4 NLP expert annotators for the task. One was asked to create a set of questions based on the three ground truth abstractive summaries from TweetSumm, and the other three were asked to read the generated summaries and answer the questions. Using the abstractive rather than the extractive summaries allows the questions to focus on the most salient information, since the extractive summaries are constrained by having a limit of sentences selected as-is from the dialog. For each dialog, $4-10$ yes/no questions regarding the information included in the summary (e.g. *"Does the summary specify that ..."*), were created by the human annotator. Following (Nenkova and Passonneau, 2004), we assigned each question a weight, w_j which is the ratio of ground-truth summaries containing an answer for question j. Clearly, important information should be included in several human summaries. Then, the other three annotators, $i \in \{1, 2, 3\}$ were given the set of questions and one summary at a time (without knowing which model generated the summary) , and were asked to indicate whether the summary contained an answer to the question. Denote the indicator I_{ij} to be 1 if annotator i determined that the summary contained an answer to question j, and 0 otherwise. The score of a summary generated by a model per dialog d is calculated as $S_d = (100/(3 * \sum_{j=1}^{K_d} w_j)) \sum_{i=1}^{3} \sum_{j=1}^{K_d} w_j * I_{ij}$, where K_d is the number of questions given d. The highest score a summary can get is 100 which occurs when all annotators agreed that the summary includes the information in all questions. Refer to the supplemental material for examples of questions that were created as part of this evaluation.

Table 6 reports the evaluation results, when calculating the summary scores separately for questions pertaining to the agent and customer utterances. The Lead-4 baseline outperforms other methods for summarizing customer utterances, which is expected as remarked in sub-section 4.2.1. In this case, the simple baseline is hard to beat. However, for summarizing agent utterances, the more advanced models are better, but even the supervised *PreSumm* and *BART* models leave much room for improvement.

Following (Liu and Lapata, 2019a), we further assess the quality of the summaries along the two dimensions of *Readability* and *Informativeness*. We chose to evaluate only the abstractive models (*BART-without ext* and *PreSumm*) since a high level of *Readability* is not expected with extractive summaries. The annotators were asked to indicate which summary is better with respect to their *Readability* and *Informativeness*, without knowing which system was used to generate which summary. In more than 90% of the cases *BART* outperforms *PreSumm* on both dimensions, consistent with the results in Table 6.

4.4 Further Analysis of BART summaries

In section 4.2.1 we showed that fine tuning BART on TweetSumm significantly improves the summaries compared to using BART with no fine tuning. Here we examine, whether using TweetSumm for fine tuning improves BART's ability to learn an important characteristic of dialog summarization, namely, that a summary should convey text from both speakers (agent and customer). We consider three variants of BART: **(1)** BART fine tuned on TweetSumm, **(2)** BART fine tuned as in (1) for which additional speaker tags (agent or customer) were added during fine tuning, **(3)** original BART variant, with no fine-tuning on TweetSumm. We generate summaries for each dialog in the test set using each of the aforementioned variants (1)-(3). Following (Nallapati et al., 2017),

Table 5: ROUGE F-Measure on validation set

Length Limit	*Method Name*	*R-1*	*R-2*	*R-SU4*	*R-L*
		Abstractive Dataset			
35 tokens	*Random*	24.459	7.719	9.504	22.157
	Lead	28.569	11.623	13.058	26.088
	LexRank	27.039	10.120	12.030	23.990
	CES	30.693	13.129	14.752	27.606
	NRP	**30.889**	**13.410**	**14.901**	**27.890**
70 tokens	*Random*	28.249	10.480	12.277	25.721
	Lead	31.127	13.536	14.867	28.542
	LexRank	30.302	12.444	14.161	27.191
	CES	**32.769**	14.125	**15.650**	**29.516**
	NRP	32.453	**14.694**	15.316	29.119

Table 6: System scores based on questions answered

Model	**Type**	**Customer**	**Agent**
LEAD	ext.	77.9	39.2
CES	ext.	69.6	49.9
NRP Summ	ext.	71.3	40.8
PreSumm†	ext.	74.3	51.2
PreSumm†	abs.	16.0	12.5
BART-without-ext†	abs.	58.5	31.7

for each generated summary, we find the two dialog utterances which are most similar to it, using ROUGE-L Recall, and ask whether they represent both speakers, or only one of them. We find that in 78% and 79% of cases, both speakers are represented for variants **(1)** and **(2)** respectively, but in only 46% of the cases for variant **(3)**. These should be compared to the baseline of choosing two random utterances, where in 58% of the cases both speakers are represented. The differences of distribution between variants **(1)** and **(2)**, compared to variants **(3)** (as well as the random baseline) are statistically significant (Welch Two Sample t-test, $p{<}10^{-6}$). **This analysis strengthens the confidence we have in TweetSumm and the ability to use it for the dialog summarization tasks**.

5 Related Work

Document Summarization- Text summarization has been studied for many years and several public datasets have been published in this domain. One central problem in summarization research is the high cost of generating ground truth data.Whereas, in some datasets, such as DUC (Dang, 2005) and Xsum (Narayan et al., 2018a), reference summaries were created specifically for the dataset, in other works different strategies are employed to identify existing texts that can be used as reference summaries. For example, in the case of single-document summarization, the CNN/Dailymail the key points associated with published news articles as part of the editorial process (Nallapati et al., 2016), are taken to be the reference summary of the news article. Other datasets, such as NewsRoom, Gigaword, NYT, (Grusky et al., 2018; Rush et al., 2015; Sandhaus, 2008) also focus on the news domain, leveraging existing texts as reference summaries. Summarization of scientific articles has also been studied as in (Yasunaga et al., 2019), treating abstracts as well as sentences describing another paper, as potential reference summaries.

Data Driven Dialog Systems- Many aspects of data driven dialog systems have undergone a revolution in recent years with the advent of ever more powerful techniques based on deep learning (Serban et al., 2016; Henderson et al., 2019; Zhang et al., 2019; Wu et al., 2020). Most of the available dialog datasets support dialog tasks such as next response prediction (Kadlec et al., 2015; Bordes et al., 2016; Byrne et al., 2019), conversational question answering (Reddy et al., 2019; Choi et al., 2018; Saeidi et al., 2018) and dialog state tracking (Budzianowski et al., 2018; Rastogi et al., 2019).

Dialog Summarization Datasets- On the other hand, summarization of two-party dialogs is relatively unexplored due to the lack of suitable large scale benchmark data. Most of the previous works on abstractive dialog summarization (Banerjee et al., 2015; Mehdad et al., 2014; Goo and Chen, 2018; Li et al., 2019) focus on the AMI meeting corpus dataset (McCowan et al., 2005). This dataset has multiple deficiencies including, its size (only 141 summaries are available), and the quality of the ground truth summaries, since the meeting description is treated as the summary. The Argumentative Dialog Summary Corpus (Misra et al., 2015), a small dataset of 45 dialogs, is based on political debates from the Internet Argument Corpus (Walker et al., 2012) where summaries are constructed by crowd-workers. More recently, CRD3 (Rameshkumar and Bailey, 2020) was introduced, a spoken conversation dataset that consists of 159 conversations and summaries. The SAMSum dialog corpus (Gliwa et al., 2019) contains over 16k chat conversations with manually annotated abstractive summaries. However, this dataset contains role-playing open domain, *chichat* dialogs, and does not provide ground truth for extractive summarization. In contrast, TweetSumm involves different summarization challenges, e.g, identifying problems and provided solutions. (Yuan and Yu, 2019) studied the problem of abstractive dialog summarization using a dataset constructed from the MultiWOZ-2.0 dataset (Budzianowski et al., 2018). This dataset considers the instructions provided to crowd-workers as part of the Wizard-of-OZ setting as the ground truth summary. Hence, the dataset does not contain "real" summary annotations for dialogs. (Liu et al., 2019) worked on the problem of automatic summary generation for customer service dialogs, but the dataset is not publicly available. Recently, MediaSum (Zhu et al., 2021) was released, suggesting the use of overview and topic descriptions as summaries of 460k interview transcripts from NPR radio channel.

6 Conclusion

In this paper, we release TWEETSUMM, the first open large-scale dataset focused on summarization of customer-support dialogs. We conducted automatic and human evaluation studies to ensure the high-quality of the human-generated extractive and abstractive summaries. To test the applicability of the dataset, we evaluated various baselines, as well as a new extractive summarization method, *NRP Summ*, and showed that while automatically generated abstractive summaries achieve high quality, there is still much room for improvement. We believe TWEETSUMM will help foster research in this real-world scenario, which was previously little studied due to lack of suitable datasets.

7 Ethics

We constructed TWEETSUMM dialogs using the publicly available *Customer Support on Twitter* dataset (`www.kaggle.com/thoughtvector/customer-support-on-twitter`). The summaries generation task was executed on Appen.com platform; we only recruited crowd-workers that are members of an Expert Business Partner channel, fluent English speakers, with a very high approved task acceptance rate. We have set the task payment, so that crowd-workers are expected to earn 9$ per hour.

References

Siddhartha Banerjee, Prasenjit Mitra, and Kazunari Sugiyama. 2015. Abstractive meeting summarization using dependency graph fusion. In *Proceedings of the 24th International Conference on World Wide Web*, pages 5–6.

Antoine Bordes, Y-Lan Boureau, and Jason Weston. 2016. Learning end-to-end goal-oriented dialog. *arXiv preprint arXiv:1605.07683*.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 2018. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026.

Bill Byrne, Karthik Krishnamoorthi, Chinnadhurai Sankar, Arvind Neelakantan, Ben Goodrich, Daniel Duckworth, Semih Yavuz, Amit Dubey, Kyu-Young Kim, and Andy Cedilnik. 2019. Taskmaster-1: Toward a realistic and diverse dialog dataset. 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 4506–4517.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. Quac: Question answering in context. *arXiv preprint arXiv:1808.07036*.

Hoa Trang Dang. 2005. Overview of duc 2005. In *Proceedings of the document understanding conference*, volume 2005, pages 1–12.

Hal Daume III and Daniel Marcu. 2005. Bayesian summarization at duc and a suggestion for extrinsic evaluation. In *Proceedings of the Document Understanding Conference, DUC-2005, Vancouver, USA*.

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.

Günes Erkan and Dragomir R. Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Int. Res.*, 22(1):457–479.

Guy Feigenblat, Haggai Roitman, Odellia Boni, and David Konopnicki. 2017. Unsupervised query-focused multi-document summarization using the cross entropy method. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, Shinjuku, Tokyo, Japan, August 7-11, 2017*, pages 961–964. ACM.

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.

Chih-Wen Goo and Yun-Nung Chen. 2018. Abstractive dialogue summarization with sentence-gated modeling optimized by dialogue acts. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 735–742. IEEE.

Max Grusky, Mor Naaman, and Yoav Artzi. 2018. Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 708–719.

Chulaka Gunasekara, Jonathan K Kummerfeld, Lazaros Polymenakos, and Walter Lasecki. 2019. Dstc7 task 1: Noetic end-to-end response selection. In *Proceedings of the First Workshop on NLP for Conversational AI*, pages 60–67.

Momchil Hardalov, Ivan Koychev, and Preslav Nakov. 2018. Towards automated customer support. In *International Conference on Artificial Intelligence: Methodology, Systems, and Applications*, pages 48–59. Springer.

Matthew Henderson, Ivan Vulić, Daniela Gerz, Iñigo Casanueva, Paweł Budzianowski, Sam Coope, Georgios Spithourakis, Tsung-Hsien Wen, Nikola Mrkšić, and Pei-Hao Su. 2019. Training neural response selection for task-oriented dialogue systems. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5392–5404.

Rudolf Kadlec, Martin Schmid, and Jan Kleindienst. 2015. Improved deep learning baselines for ubuntu corpus dialogs. *arXiv preprint arXiv:1510.03753*.

Wojciech Kryscinski, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Neural text summarization: A critical evaluation. 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 540–551, Hong Kong, China. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

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: Proceedings of the ACL-04 workshop*, volume 8. Barcelona, Spain.

Hui Lin and Vincent Ng. 2019. Abstractive summarization: A survey of the state of the art. In *AAAI*.

Chunyi Liu, Peng Wang, Jiang Xu, Zang Li, and Jieping Ye. 2019. 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 and Mirella Lapata. 2019a. Hierarchical transformers for multi-document summarization. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5070–5081. Association for Computational Linguistics.

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.

Yang Liu and Mirella Lapata. 2019c. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 3728–3738. Association for Computational Linguistics.

I. McCowan, J. Carletta, W. Kraaij, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, M. Kronenthal, G. Lathoud, M. Lincoln, A. Lisowska, W. Post, Dennis Reidsma, and P. Wellner. 2005. The ami meeting corpus. In *Proceedings of Measuring Behavior 2005, 5th International Conference on Methods and Techniques in Behavioral Research*, pages 137–140. Noldus Information Technology.

Yashar Mehdad, Giuseppe Carenini, and Raymond Ng. 2014. Abstractive summarization of spoken and written conversations based on phrasal queries. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1220–1230.

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, Bing Xiang, and Bowen Zhou. 2016. Sequence-to-sequence rnns for text summarization. *CoRR*, abs/1602.06023.

Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 3075–3081. AAAI Press.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018a. 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*, Brussels, Belgium.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018b. 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.

Ani Nenkova and Rebecca J. Passonneau. 2004. Evaluating content selection in summarization: The pyramid method. In *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, HLT-NAACL 2004, Boston, Massachusetts, USA, May 2-7, 2004*, pages 145–152. The Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).

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.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2019. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. *arXiv preprint arXiv:1909.05855*.

Siva Reddy, Danqi Chen, and Christopher D Manning. 2019. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Haggai Roitman, Guy Feigenblat, Doron Cohen, Odellia Boni, and David Konopnicki. 2020. Unsupervised dual-cascade learning with pseudo-feedback distillation for query-focused extractive summarization. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, pages 2577–2584. ACM / IW3C2.

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389.

Marzieh Saeidi, Max Bartolo, Patrick Lewis, Sameer Singh, Tim Rocktäschel, Mike Sheldon, Guillaume Bouchard, and Sebastian Riedel. 2018. Interpretation of natural language rules in conversational machine reading. *arXiv preprint arXiv:1809.01494*.

Evan Sandhaus. 2008. The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia*, 6(12):e26752.

Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Assaf Toledo, Shai Gretz, Edo Cohen-Karlik, Roni Friedman, Elad Venezian, Dan Lahav, Michal Jacovi, Ranit Aharonov, and Noam Slonim. 2019. Automatic argument quality assessment-new datasets and methods. 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 5629–5639.

Marilyn Walker, Jean E Fox Tree, Pranav Anand, Rob Abbott, and Joseph King. 2012. A corpus for research on deliberation and debate. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 812–817.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355.

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*, pages arXiv–1910.

Zeqiu Wu, Michel Galley, Chris Brockett, Yizhe Zhang, Xiang Gao, Chris Quirk, Rik Koncel-Kedziorski, Jianfeng Gao, Hannaneh Hajishirzi, Mari Ostendorf, et al. 2020. A controllable model of grounded response generation. *arXiv preprint arXiv:2005.00613*.

Michihiro Yasunaga, Jungo Kasai, Rui Zhang, Alexander R Fabbri, Irene Li, Dan Friedman, and Dragomir R Radev. 2019. Scisummnet: A large annotated corpus and content-impact models for scientific paper summarization with citation networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7386–7393.

Lin Yuan and Zhou Yu. 2019. Abstractive dialog summarization with semantic scaffolds. *arXiv preprint arXiv:1910.00825*.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2019. Dialogpt: Large-scale generative pre-training for conversational response generation. *arXiv preprint arXiv:1911.00536*.

Chenguang Zhu, Yang Liu, Jie Mei, and Michael Zeng. 2021. Mediasum: A large-scale media interview dataset for dialogue summarization. *arXiv preprint arXiv:2103.06410*.

A TweetSumm Dataset - Summaries Generation

As described in the main paper, TweetSumm dialogs were sent for generation of summaries using crowd-sourcing on the Appen.com platform. Figure 3 shows the instructions provided to annotators working on TweetSumm summary generation task. Figure 4 shows how dialogs were presented to annotators as part of the annotation interface. Figure 5 shows the dialog annotation interface: annotators are asked to highlight the salient sentences (extractive summary) in the dialog. In the following sub-sections we describe in details the instructions crowd-workers received while working on this task.

A.1 Extractive Summaries

The annotators were asked to select 2 to 3 entire sentences that describe the most important messages the customer conveyed. They were asked to focus on sentences presenting a problem, complaint, or a request the customer expressed. Then, they were asked to select between 2 to 3 entire sentences representing the agent response to the customer, with focus on actual solutions and not on apologies or gratitude expressions. Clearly, the analysis of the emotional part of customer interactions is also important. However, this is associated with other NLP tasks such as sentiment analysis. The same decision was taken in (Liu et al., 2019). As a final step, the annotators were asked to go over the selected summary sentences and make sure that they represent the full dialog as much as possible. In addition, several examples of uninformative sentences, that should not appear in summaries, were given to help annotators understand the requirements better (e.g. *"We're sorry to hear that."*, *"Poor customer service."*, *"Hi again, we'd like to investigate this behavior."*, *"I hate X company"*).

A.2 Abstractive Summaries

Here, the annotators were instructed to write two sentences summarizing the whole dialog, one summarizing the customer questions/requests and the second one summarizing the agent responses. We limited ourselves to two sentences to simplify the task of the crowd-workers. In addition, having separate summary sentences allow an automated summarizer to (potentially) generate two summaries, one for the customer and one for the agent. Similarly to the extractive summarization, annotators were asked to write an informative summary, that focuses on requests, problem descriptions and solutions excluding personal opinions, insults or apologies.

B Model Training and Hyperparameter Details

In this section, we elaborate the training processes and the hyperparameters used in the supervised trained models used in this study. Each experiment was run on 2 V100 GPUs (on a single machine).

B.1 Next response prediction model for NRP Sum

As introduced in the main paper, the NRP Sum model uses a BERT based binary classifier. The code will be open-sourced in a public git page upon paper acceptance. For this task, we used the *BertForSequenceClassification* model of HuggingFace (Wolf et al., 2019), commonly used for GLUE tasks (Wang et al., 2018). We process the dataset to construct triples of <dialog context (C), candidate (c_i), label ($1/0$)> from each dialog context. For each C, we create a set of 10 triples: one triple containing the correct response (label=1), and 9 triples containing incorrect responses randomly sampled from the dataset (label=0). Training is done using positive and negative examples with cross-entropy loss.

The hyperparameters used for training the model are as follows:

```
model=bert-base-cased
do_lower_case=True
max_seq_length=512
per_gpu_eval_batch_size=24
per_gpu_train_batch_size=24
learning_rate=2e-5
num_train_epochs=5
adam_epsilon=1e-8
max_grad_norm=1.0
```

We trained two models with this approach, one for predicting the next response given a dialog context and, another to predict the previous sentence given the dialog context. The results of the two models on the validation set are shown in Table 1.

B.2 PreSumm model

The PreSumm (Liu and Lapata, 2019b) model was used as a baseline in this study. We used the

Model	R@1	R@2	R@5
NRP	56.09	75.95	98.08
PRP	51.91	73.51	95.64

Table 7: The results of the next response prediction task. The model NRP refers to the task of predicting the next response given a dialog context, and the model PRP refers to the task of predicting the previous response given a dialog context.

PreSumm extractive summarization model which was pre-trained on the CNN/DM summarization dataset, and fine-tuned the model on the TWEETSUMMdataset. All the code and pre-trained models used in this study are publicly available[6].

The hyperparameters used for training the extractive summarization model are as follows:

```
ext_dropout=0.1
lr=2e-3
save_checkpoint_steps=5000
batch_size=3000
train_steps=50000
accum_count=2
warmup_steps=10000
max_pos=512
```

The checkpoint which produced the best performance on the validation dataset (checkpoint at step 35000) was used to initialize the PreSumm abstractive summarization model. The hyperparameters used for training the abstractive summarization model are as follows:

```
dec_dropout=0.2
sep_optim=true
lr_bert=0.002
lr_dec=0.2
save_checkpoint_steps=5000
batch_size=140
train_steps=100000
accum_count=5
use_bert_emb=true
use_interval=true
warmup_steps_bert=20000
warmup_steps_dec=10000
max_pos=512
beam_size=5
```

The checkpoint which produced the best performance on the validation dataset (checkpoint at step 55000) was used to generate summaries on the test dataset.

B.3 BART models

As a fully abstractive summarization algorithm, we used the BART model (Lewis et al., 2019) in this study. We use a lightweight variant of BART, named DistilBART provided by HuggingFace (Wolf et al., 2019) library[7]. This instance of DistilBART is fine-tuned on the extreme summarization (XSum) task, and we fine-tune this model on the TWEETSUMMdataset. The code used for the fine-tuning is publicly available[8].

The hyperparameters used for training the DistilBART model are as follows:

```
train_batch_size=4
eval_batch_size=4
num_train_epochs=6
model_name_or_path=sshleifer/distilbart
-xsum-12-6
learning_rate=3e-5
val_check_interval=0.1
max_source_length=512
max_target_length=80
```

C Sample summaries with corresponding QA questions

Figure 2 shows an example of a TWEETSUMM human-generated abstractive summary along with machine-generated summaries and their corresponding QA questions. Upon acceptance of the paper, TWEETSUMM release will include the set of questions that were generated as part of the human evaluation task in the Results section.

[6] https://github.com/nlpyang/PreSumm

[7] https://huggingface.co/sshleifer/distilbart-cnn-12-6

[8] https://github.com/huggingface/transformers/tree/master/examples/seq2seq

An awful smell in a flight

Ground truth (human) abstractive summary	
	Customer complains about smell in flight. Agent updated the customer to seek further assistance by reaching out to one of their in-flight crew members on duty.
Sample QA Questions	
	Does the summary specify the customer is complaining about bad smell in his flight?
	Does the summary specify the agent asked to contact in-flight crew member on duty for assistance?
	Does the summary specify the customer asked to change seat in rebooking?
	Does the summary specify the agent apologized for the discomfort?
Automated abstractive summary	
BART	Customer is complaining about the smell on flight 1287 from Miami to LaGuardia. Agent requests to reach out to a flight attendant to address the odor in the aircraft.
Automated extractive summaries	
NRP	**Customer** Flight1287 from Miami to LaGuardia smells awful. Every person getting on the flight is complaining.
	Agent Unfortunately, our First Class Cabin is full on our DL1287 flight for today, Allie. Please reach out to a flight attendant to address the odor in the aircraft.
LEAD	**Customer** Flight1287 from Miami to LaGuardia smells awful. It's really really bad.
	Agent Allie, I am very sorry about this. Please reach out to a flight attendant to address the odor in the aircraft.
CES	**Customer** Flight1287 from Miami to LaGuardia smells awful. They told us to rebook, then told us the original flight was still departing.
	Agent Unfortunately, our First Class Cabin is full on our DL1287 flight for today, Allie. You may seek further assistance by reaching out to one of our in-flight crew members on duty.

A Red Eye Removal issue

Ground truth (human) abstractive summary	
	Customer is asking help how to remove red eye in Ligthroom CC since he can't find it in tool, and customer wants some new advanced features. Agent is giving details on it, then sends a link where he can get help and also asks customer to report a complaint and his engineer team will get alert and help him over it.
Sample QA Questions	
	Does the summary specify the customer asks to do red eye removal?
	Does the summary specify the customer is using Lightroom CC?
	Does the summary specify the agent sent an article containing the required information?
	Does the summary specify the agent explained the released version contains all the features of the old version?
	Does the summary specify the agent suggested the customer to report a complaint so the engineering team will get an alert and help?
Automated abstractive summary	
BART	Customer is asking how to do red eye removal in Lightroom CC. Agent is looping their expert team to help answer the question.
Automated extractive summaries	
NRP	**Customer** Can you tell me how to do Red Eye Removal in Lightroom CC? I just moved to it and don't see the Red Eye Removal tool.
	Agent Hi Bob, here is a link to show you to use the Red eye removal in Lightroom CC. Hi Bob, I am looping our expert team to help answer your question.
LEAD	**Customer** Can you tell me how to do Red Eye Removal in Lightroom CC? I just moved to it and don't see the Red Eye Removal tool.
	Agent Hi Bob, here is a link to show you to use the Red eye removal in Lightroom CC. Please let us know if you have any questions or need further help.
CES	**Customer** Can you tell me how to do Red Eye Removal in Lightroom CC? I wish a list of features missing in Lightroom CC would have been noted before I migrated my library.
	Agent Hi Bob, this feature is not available in Lightroom CC as of now, however you may suggest it as a feature here: [URL]. We have released Lightroom Classic CC which has all the features the old Lightroom CC 2015.12 had, you can check this article to see the differences betweem LR Classic and the new Lightroom CC: [URL].

Figure 2: Two ground-truth summaries with corresponding automated summaries and QA questions

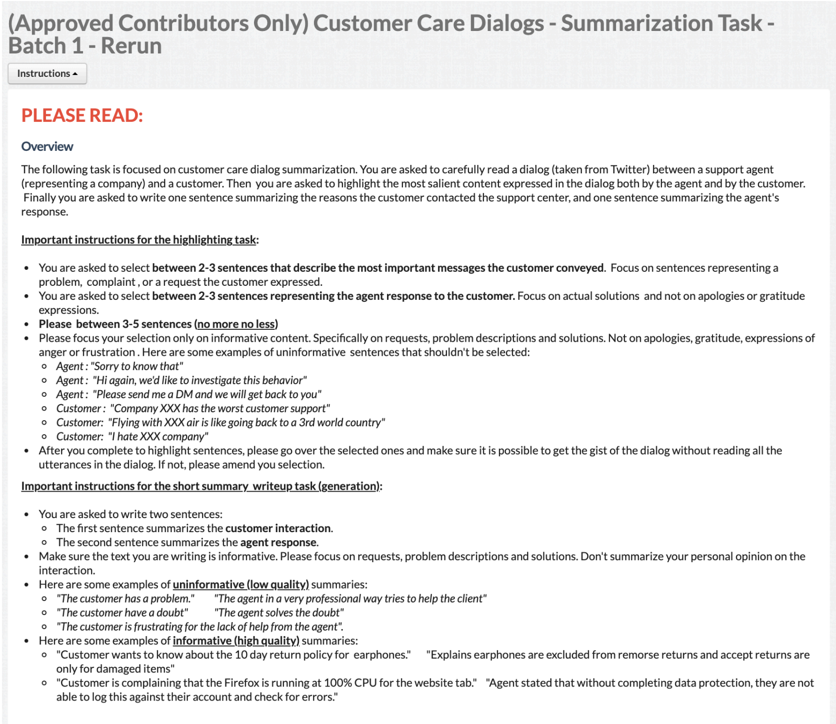

Figure 3: Annotation interface - Instructions for the summary generation task

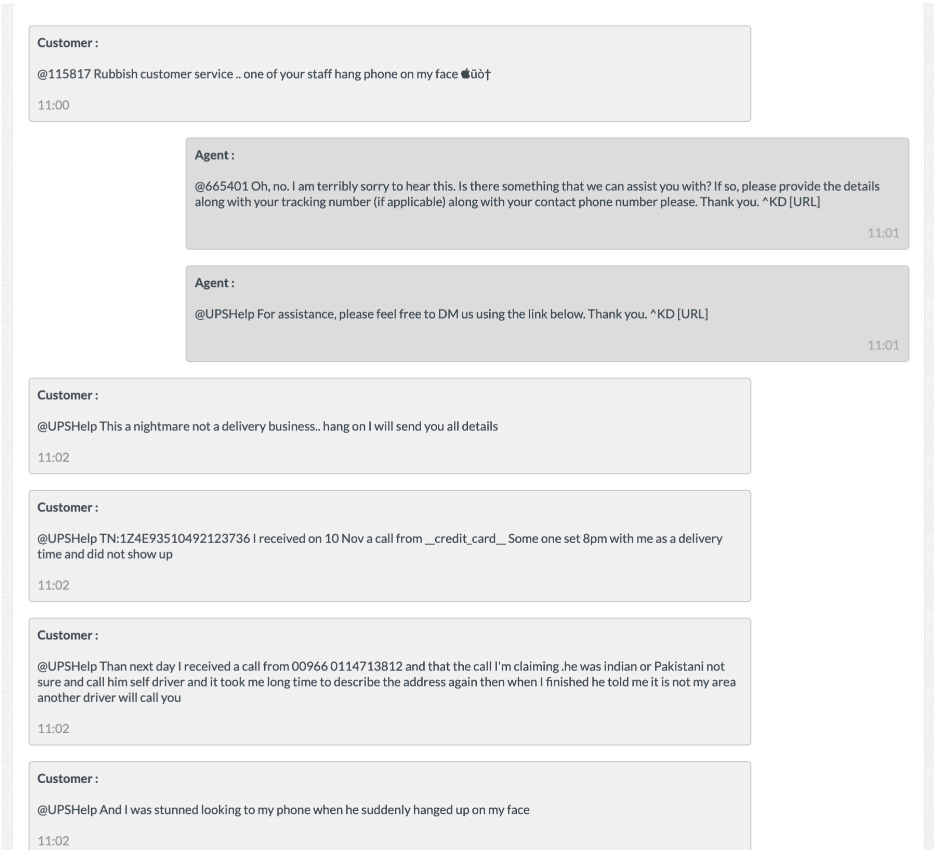

Figure 4: Annotation interface - A dialog presented to annotators

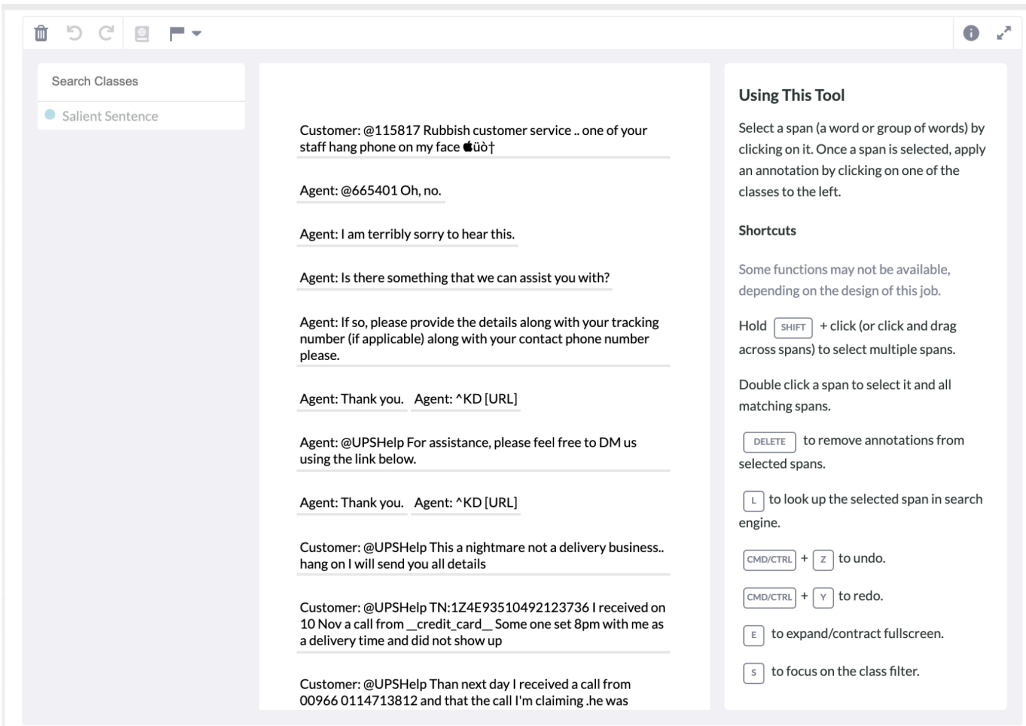

Figure 5: Annotation interface - Annotators are asked to highlight salient sentences (for the extractive summary)