Abstract

Many conversation datasets have been constructed in the recent years using crowd-sourcing. However, the data collection process can be time consuming and presents many challenges to ensure data quality. Since language generation has improved immensely in recent years with the advancement of pre-trained language models, we investigate how such models can be utilized to generate entire conversations, given only a summary of a conversation as the input. We explore three approaches to generate summary grounded conversations, and evaluate the generated conversations using automatic measures and human judgements. We also show that the accuracy of conversation summarization can be improved by augmenting a conversation summarization dataset with generated conversations.

1 Introduction

Automatic conversation systems require large quantities of data to learn task specific language patterns and underlying conversation policies. Such data either come from human-to-human conversation logs (Lowe et al., 2015; Hardalov et al., 2018) or is collected in crowd-sourced environments, where two or more crowd-workers play specific roles under some guidelines (Zhang et al., 2018; Budzianowski et al., 2018). Since real human-to-human conversation logs are scarce, many datasets have been created using the latter approach. However, crowd-sourced conversation data collection is time consuming, costly and presents multiple challenges to ensure data quality (Kang et al., 2018).

Conversation summarization is an emerging research area that has been ill-studied due to the lack of large-scale datasets. Most existing public datasets in this domain are small, for example, AMI meeting corpus (McCowan et al., 2005) contains 137 summary transcripts. CRD3 (Rameshkumar and Bailey, 2020) is a spoken conversation dataset that consists of 159 conversations and summaries. Samsum (Gliwa et al., 2019), the only large scale dataset for conversation summarization, contains over 16,000 open-domain conversations and summaries created artificially by humans.

Large scale pre-trained language models (PLMs) (Lewis et al., 2020; Brown et al., 2020; Raffel et al., 2020) have been used in various text generation tasks (Budzianowski and Vulić, 2019; Min et al., 2020; Cachola et al., 2020). In recent studies, PLMs are used to generate training data for natural language processing (NLP) applications. For example, Anaby-Tavor et al. (2020); Yang et al. (2020) use PLMs to create paraphrases for intent classifiers in conversation systems, and show that, when the original datasets are augmented with the generated data, performance improves. More recently Mohapatra et al. (2020) generated entire conversations grounded on instructions that are provided to crowd-workers using a modular approach, where different PLMs are trained for different roles.

Our Contributions: We investigate how PLMs can be utilized to generate entire conversations that are grounded on a given summary. We explore three approaches: (1) Supervised Learning (SL) based conversation generation (SL-Gen): where, a PLM is trained to generate an entire conversation, taking the summary of a conversation as input, (2) Reinforced Learning (RL) based conversation generation (RL-Gen): where, we further improve the SL-Gen method using the quality of the generated conversations as a reward, and (3) Controlled turn-by-turn conversation generation (CN-Gen): which allows us to generate conversations turn-by-turn, constrained on the summary and a set of pre-defined control parameters. We evaluate the quality of the generated conversations by conducting automatic and human evaluation. We
also show that once a conversation summarization dataset is augmented with the generated conversations, the performance of the downstream summarization task is improved.

2 Summary grounded conversation generation

In the conversation summarization task, a model takes a conversation as input, and learns to generate a summary. We study the inverse of that problem, where the input to our model is a summary, and the model generates a conversation. In this section, we propose three models for this task and the hyper-parameters used in training the models are available in Section A of the appendix.

2.1 SL based generation (SL-Gen)

A seq2seq model can be trained for this task by providing a summary as the input and generating a conversation token-by-token. As PLMs have shown significant improvement over the traditional seq2seq architecture for text generation, we use a GPT-2 model and fine-tune it to generate a conversation given a summary as the input. Our input to the model follows the following format: 

\[ <\text{bos}> \text{summary text} <\text{dialog}> \text{conversation text} <\text{eos}>. \]

We also use different token-type-ids to indicate the summary and the conversation text. The model is trained to optimize Cross Entropy loss.

2.2 RL based generation (RL-Gen)

Many studies train text generation models with RL (Paulus et al., 2018; Li et al., 2016), where the generator network is optimized with a task specific reward. We investigate how the quality of the generated conversation can be used as a reward to improve the generation network. To this end, we train a summary generator network, which generates a summary, given a conversation. We measure the quality of the generated conversation by identifying the similarity between the summary of the generated conversation (generated, in turn, by the summary generator network) and the ground truth summary. The similarity score is used as a reward to train the conversation generation model. Our RL based generation framework is shown in Figure 1, and the critical components are described below.

**Conversation Generator:** A trained SL-Gen model is used as the conversation generator, which, given an summary can generate a conversation.

**Summary Generator:** We use a lightweight variant of BART (Lewis et al., 2019), named DistilBART, which is fine-tuned on the Extreme summarization task (Narayan et al., 2018). We further fine-tune this instance on the conversation summarization data by providing the conversations as the input and training the model to output summaries.

**Reward Model:** Once the Summary Generator generates an output summary for the generated conversation, the reward model compares it with the ground truth summary, which was used to ground the conversation generation. As Paulus et al. (2018) we use ROUGE-2 F1-score as the reward.

**Policy training:** We use proximal policy optimization (Schulman et al., 2017) as the optimizer for the policy training as it prevents the generator from deviating far away from the pretrained LM (Wu et al., 2020).

2.3 Controlled conversation generation

We propose another approach, (CN-Gen), for conversation generation, which grants more control over the properties of the generated conversations. Here, we generate one utterance of the conversation at a time, as opposed to the RL-Gen, where we generate the whole conversation at once. The properties of the generated conversations is controlled by adding several components to the input sequence to the model. The following three variables were used as the control parameters, (1) Number of remaining turns to generate in the conversation (Num turns): During the generation of a turn, we indicate the remaining number of turns in the conversation. In generating a \( n \) turn conversation, this starts with \( n \) for the first turn and reduces by 1 after the generation of each turn, (2) The speaker of the next
3 Experiments

We experiment on the Samsum (Gliwa et al., 2019) dataset, which, to the best of our knowledge, is the only public large-scale conversation summarization dataset. We pre-process the dataset by replacing the personal names (ex: John) with unique tags (ex: <person_0>). First, we evaluate the quality of generated conversations using automatic measures and human judgments, and then assess the performance of the generated conversations in a downstream summarization task after augmentation.

3.1 Quality of the generated conversations

We evaluate the quality of the conversations generated by the three approaches that were introduced in Section 2. In Table 2 we show the properties of generated conversations and the ground truth conversations in the test set of Samsun dataset.

**Table 2:** Properties of the generated conversations.

| Model         | Ave. Turns | Ave. Tokens/Turn |
|---------------|------------|------------------|
| Ground truth  | 11.55 ± 6.48 | 7.10 ± 6.29     |
| SL-Conv-Gen   | 10.54 ± 6.80 | 5.69 ± 4.40     |
| RL-Conv-Gen   | 8.40 ± 4.78  | 5.14 ± 3.64     |
| CN-Conv-Gen   | 9.70 ± 5.67  | 5.62 ± 4.05     |

**Table 1:** Multiple conversations generated by the CN-Gen approach grounded on the same summary turn (Speaker): This indicates to the model the speaker of the next turn, and (3) The length of the next turn (Turn length): We define, 3 categories of lengths: Short (≤ 3 tokens), Long (> 10 tokens) and Medium (otherwise).

We use the following input representation to fine-tune a GPT-2 model: `<box> summary text <context> dialog context <turns_to_go> Num turns <speaker> speaker <turn_length> turn length <utterance> <eos>`.

| 2 turn conversation | 3 turn conversation |
|---------------------|---------------------|
| `<person>`: I'll be late | `<person>`: I'll be late |
| `<person>`: I'll order some pasta with salmon and basil for you. | `<person>`: I'll order some pasta with salmon and basil for you. |
| `<person>`: Thanks a lot! | `<person>`: Thanks a lot! |

| 6 turn conversation | 10 turn conversation |
|---------------------|----------------------|
| `<person>`: Hello, I am going to be late. | `<person>`: I'll be late |
| `<person>`: Ok | `<person>`: ok |
| `<person>`: I'll order some pasta with salmon and basil | `<person>`: do you want me to order something for you? |
| `<person>`: Ok, sounds good! | `<person>`: pasta? |
| `<person>`: Yes | `<person>`: Yes |
| `<person>`: with salmon? | `<person>`: Yes |
| `<person>`: Yes | `<person>`: Yes |
| `<person>`: how about basil? | `<person>`: Ok |
| `<person>`: I'll order some pasta | `<person>`: Thanks a lot! |
| `<person>`: I'll be late. | `<person>`: Late. |
| `<person>`: Yes please! | `<person>`: Ok |

A summary and a conversation from the Samsum dataset (Gliwa et al., 2019), along with the conversations generated by the three aforementioned algorithms are shown in Figure 2. More examples are provided in the Section B of the Appendix.
Figure 2: Examples of a conversations grounded on the same summary. The key terms are highlighted in colors.

| Ground Truth Dialogue | SL-Gen | RL-Gen | CN-Gen |
|-----------------------|--------|--------|--------|
| <person_0> Hi darling, do you want to come for Easter? | Hi person_1, I'm coming this weekend. | Hi person_1, I'm coming this Easter! | Hi person_1, I'm coming this Easter! |
| <person_1> Love to, I'm off on Friday. It's a nice day, I'll invite Louise too! | Ok, I'll bring some chocolate eggs and some cake | Ok, I'll bring some chocolate eggs and some cake | Ok, I'll bring some chocolate eggs and some cake |
| <person_0> Are you still coming this weekend? | Ok, I'll bring some chocolate eggs and some cake | Ok, I'll bring some chocolate eggs and some cake | Ok, I'll bring some chocolate eggs and some cake |
| <person_1> Great! | Ok, I'll bring some chocolate eggs and some cake | Ok, I'll bring some chocolate eggs and some cake | Ok, I'll bring some chocolate eggs and some cake |
| <person_0> Thanks darling. | | | |

Table 3: Evaluation of generated conversations against ground truth conversations

| Model  | BLEU-4 | METEOR | ROUGE-L |
|--------|--------|--------|---------|
| SL-Gen | 2.81   | 12.06  | 21.53   |
| RL-Gen | 3.53   | 12.29  | 25.40   |
| CN-Gen | 4.94   | 15.64  | 26.22   |

| Model  | ROUGE_1 | ROUGE_2 | ROUGE_L |
|--------|---------|---------|---------|
| SL-Gen | 46.85   | 25.29   | 45.97   |
| RL-Gen | 52.51   | 31.23   | 51.68   |
| CN-Gen | 53.46   | 32.52   | 52.93   |

3.2 Evaluation on the summarization task

To further evaluate the quality of the generate conversations, we augmented a conversation summarization dataset with generated conversations and evaluated the summarization model. We followed the following process: (1) We randomly selected x% of the summaries of the dataset and trained our conversation generation models, (2) The trained models were applied on the other (y=100-x%) of the summaries and generated conversations, (3) Those generated conversations along with the original summaries were added to the data. Using this approach, we can add extra y% (summary, conversation) pairs to the training data, (4) The conversation summarization model (discussed in Section 2 under ‘Summary Generator’) was trained on the augmented data. We compare the performance of the conversation summarization model on the original dataset and with augmentation.

Automatic Evaluation: We compare the three conversation generation methods at different augmentation percentages, and the results are shown in Table 7. At all augmentation levels, the summarization models trained with augmented data outperform the summarization model trained on the original dataset (without augmentation). CN-Gen based augmentation produces the best accuracy compared to other two methods. One prevalent pattern is that, when augmentation data increases, the accuracy seems to increase up to a certain point and then starts to decrease. The best accuracies were found around 30% data augmentation. We believe that more augmentation leads performance to drop due to the following reason. For augmenting with more data, we are left with less data to train the model for conversation generation (for 10% augmentation, the conversation generation models are trained on 90% of the data, while for 50% augmentation, the models are trained only on 50% of the data).
## Table 5: Human evaluation of generated conversations

| Model   | Info | Gram | Cohe |
|---------|------|------|------|
| Ground-True | 4.56 | 4.46 | 4.47 |
| SL-Gen  | 2.22 | 2.85 | 2.37 |
| RL-Gen  | 3.20 | 3.50 | 3.14 |
| CN-Gen  | 3.10 | 3.43 | 3.09 |

## Table 6: Average Cohen’s Kappa for human evaluation of generated conversations

| Model   | Info | Gram | Cohe |
|---------|------|------|------|
| Ground-True | 0.04 | 0.22 | 0.25 |
| SL-Gen  | 0.35 | 0.26 | 0.42 |
| RL-Gen  | 0.47 | 0.35 | 0.45 |
| CN-Gen  | 0.60 | 0.40 | 0.60 |

## Table 7: ROUGE F-1 evaluation on Samsum test set.

| Method | Augmentation % | ROUGE_1 | ROUGE_2 | ROUGE_L |
|--------|----------------|---------|---------|---------|
|        | 0% (Original)  | 51.84   | 30.98   | 43.98   |
| SL-Gen | 10%            | 52.82   | 31.99   | 44.89   |
|        | 20%            | 52.90   | 32.01   | 44.97   |
|        | 30%            | 52.88   | 32.02   | **45.01** |
|        | 40%            | 52.61   | 31.98   | 44.96   |
|        | 50%            | 52.55   | 31.98   | 44.80   |
| RL-Gen | 10%            | 52.93   | 32.05   | 44.92   |
|        | 20%            | 53.30   | 32.15   | 45.20   |
|        | 30%            | 53.81   | 32.21   | **45.77** |
|        | 40%            | 52.86   | 32.06   | 44.99   |
|        | 50%            | 52.64   | 32.07   | 44.88   |
| CN-Gen | 10%            | 53.29   | 32.36   | 45.08   |
|        | 20%            | 53.36   | 32.53   | 45.27   |
|        | 30%            | **54.02** | 33.28 | **46.06** |
|        | 40%            | 52.14   | 31.76   | 44.14   |
|        | 50%            | 52.36   | 31.75   | 44.85   |

Therefore as the augmentation increases, the quality of generated conversations go down. This leads to overall smaller gains in the summarization task with increased augmentation after some point. To neutralize the effect of increasing the data points during augmentation, we experimented with a baseline which over-samples the original training data at different percentages to obtain same number of training instances as the augmented datasets. While the ROUGE-2 obtained with the original training data is 30.98, oversampling at 10%, 20%, 30%, 40% and 50%, only changes the ROUGE-2 to 30.55, 30.38, 30.74, 30.99 and 30.27 respectively. Hence, this suggests that oversampling hardly changes ROUGE scores obtained by training with the original dataset, while the augmentation according to our algorithms show significantly improved scores (as shown in Table 7).

### Human Evaluation:
We recruited 3 NLP experts to evaluate 50 instances of summaries generated with data augmentation (RL-Gen, CN-Gen), and respective summaries generated without augmentation (No-Aug). Here we consider two aspects with respect to a ground-truth summary: Coherence (whether the summary is easy to read) and Focus (whether the summary represents the ground-truth summary). Following (Amplayo and Lapata, 2020) we use the Best-Worst Scaling method. The score of each system is computed as the percentage of times it was chosen as the Best system minus times it was chosen as Worst. On the Coherence question, RL-Gen, CN-Gen and No-Aug obtained scores of 12.6, 6.6 and -4.0 respectively. On the Focus question RL-Gen, CN-Gen, and No-Aug obtained scores of 14.6, 6.0 and -2.6 respectively. These results confirm that the use of augmentation improves the quality of the summaries.

## 4 Conclusion

We investigated how the PLMs can be utilized to generate entire conversations that are grounded on a summary. We propose three approaches for conversation generation: SL-Gen, RL-Gen and CN-Gen and conducted multiple automatic and human evaluations to assess the quality of the generated conversations. Both automatic and human evaluations show that when compared to the ground truth conversations, RL-Gen and CN-Gen obtain high scores, suggesting that the proposed models generate high quality conversations. When a conversation summarization dataset is augmented with the generated conversations, the performance of conversation summarization is improved (over to 7% improvement in ROUGE-2 F-1), which also suggests that the proposed methods generate high quality conversations.

## 5 Ethics

We have used the publicly available Samsum dataset ([https://huggingface.co/datasets/samsum](https://huggingface.co/datasets/samsum)). For the human evaluation of both conversations and summaries, we recruited 3 NLP researchers, who have graduate degree in NLP and Machine Learning. The annotation task itself was executed on Appen.com platform. Before the official annotation, we sampled 10 tasks to get an estimate of the duration of the task, and to make sure the instructions are clear enough.
References

Reinald Kim Amplayo and Mirella Lapata. 2020. Unsupervised opinion summarization with noising and denoising. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 1934–1945.

Ateri Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. 2020. Do not have enough data? deep learning to the rescue! In Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pages 7383–7390.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. arXiv preprint arXiv:2005.14165.

Paweł Budzianowski and Ivan Vulić. 2019. Hello, it’s gpt-2-how can i help you? towards the use of pre-trained language models for task-oriented dialogue systems. In Proceedings of the 3rd Workshop on Neural Generation and Translation, pages 15–22.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Ihigo 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.

Isabel Cachola, Kyle Lo, Arman Cohan, and Daniel S Weld. 2020. Tldr: Extreme summarization of scientific documents. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, pages 4766–4777.

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.

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.

Yiping Kang, Yunqi Zhang, Jonathan K Kummerfeld, Lingjia Tang, and Jason Mars. 2018. Data collection for dialogue system: A startup perspective. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers), pages 33–40.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461.
A.1 Supervised Conversation Generation (SL-Conv-Gen)

We fine-tune a GPT-2 language model using the implementation available at HuggingFace (Wolf et al., 2019). The hyper-parameters used during training and inference are shown below. The model takes around 6 hours to train on 2 V100 GPUs (single machine).

```
model_name_or_path: gpt2
per_gpu_train_batch_size: 4
per_gpu_eval_batch_size: 4
gradient_accumulation_steps: 4
learning_rate: 6.25e-5
adam_epsilon: 1e-8
max_grad_norm: 1.0
num_train_epochs: 10
warmup_steps: 500
min_length: 20
max_length: 512
top_k: 0
top_p: 0.95
```

A.2 Summary Generator

We use DistilBART instance\(^1\) fine-tuned on the extreme summarization (XSum) task, and we fine-tune this model further on the Samsum dataset. The model takes around 12 hours to train on 2 V100 GPUs (single machine).

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

```
train_batch_size: 4
eval_batch_size: 4
num_train_epochs: 10
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
```

A.3 Reinforced Learning based conversation generation (RL-Conv-Gen)

To train the RL based conversation generation model, we adapted a publicly available Proximal Policy Optimization (PPO) implementation\(^2\). The model takes around 12 hours to train on 2 V100 GPUs (single machine). Following hyperparameters were used to train the model.

```
steps: 10000
batch_size: 16
forward_batch_size: 4
learning_rate: 1.41e-5
init_kl_coef: 0.2
target: 6
horizon: 10000
gamma: 1
lambda: 0.95
cliprange: 0.2
cliprange_value: 0.2
vf_coef: 0.1
```

\(^1\)https://huggingface.co/sshleifer/distilbart-cnn-12-6
\(^2\)https://github.com/lvwerra/trl
B Sample summaries with corresponding ground-truth

Figure 3 shows some samples of dialogs with their corresponding summaries - ground-truth and automatic generated ones.
### Summary:
Person0 closed some deals today. Person1 didn’t manage to do it.

#### Ground Truth Dialog:

- **Person0**
  - I have a problem with the application.
  - I want to keep some pasta in the microwave.
  - I bought everything.
  - Look what I’ve bought! A table, six chairs, a vase and a pile of clothes.
  - How much did you pay for everything?
  - I paid 70 euros for everything.

- **Person1**
  - I closed some deals today.
  - I'm busy at work.
  - I closed some deals today.
  - Yes, I do. What’s inside?
  - I closed some deals today.

#### SL-Gen:
- **Person0**
  - HI, are you still alive?
  - HI, yes, still alive.
  - HI, I closed some deals today.
  - Thanks.

- **Person1**
  - Hey, how are you?
  - Good, busy as usual.
  - I closed some deals today.
  - Sorry.

#### RL-Gen:
- **Person0**
  - Do you remember the second hand shop downtown?
  - Yes, I do. What’s inside?
  - A table, six chairs, a vase and a pile of clothes.
  - It is expensive?
  - 70 euros.

- **Person1**
  - Are you at home?
  - No, I’m busy at work.
  - Good, I should be home in about 15 minutes.
  - Can you put my pasta in the microwave?
  - Sure, I’ll check it.

#### CN-Gen:
- **Person0**
  - Did you see this? I bought a table, six chairs, a vase and a pile of clothes.
  - Cool. What’s up?
  - I closed some deals today and I feel good about it.
  - Great. I didn’t manage to do it today. I’m sorry. I had no choice.

- **Person1**
  - I’m sorry.
  - I didn’t have the tools.

### Summary:
Person0 bought a table, six chairs, a vase and a pile of clothes and the second hand shop downtown. She paid 70 euros for everything.

#### Ground Truth Dialog:

- **Person0**
  - I went to the second hand shop downtown.
  - I bought everything.
  - I paid 70 euro for everything.
  - I bought a table, six chairs, a vase and a pile of clothes.
  - What's up, mate?

- **Person1**
  - How are you doing?
  - Yes, I'm at work.
  - No, I'm busy at work.
  - How much?
  - 6 chairs, six chairs, a vase and a pile of clothes.

#### SL-Gen:
- **Person0**
  - I bought the second hand shop downtown.
  - So what happened?
  - I paid 70 euro for everything.
  - I love the black dress.

- **Person1**
  - I'm not at home.
  - Sounds like a bargain.
  - Should be in business.

#### RL-Gen:
- **Person0**
  - Do you remember the second hand shop downtown?
  - Yes, I do. What’s inside?
  - A table, six chairs, a vase and a pile of clothes.
  - Is it expensive?
  - 70 euros.

#### CN-Gen:
- **Person0**
  - Look what I’ve bought! A table, six chairs, a vase.
  - I bought a table, six chairs, a vase and a pile of clothes.
  - How much did you pay for everything?
  - I paid 70 euro for everything.

- **Person1**
  - Me too. 70 euro for everything, including a vase!!!