Paraphrase Augmented Task-Oriented Dialog Generation

Silin Gao[1*], Yichi Zhang[1], Zhijian Ou[1,2], Zhou Yu[3]
[1] Speech Processing and Machine Intelligence Lab, Tsinghua University, Beijing, China
[2] Beijing National Research Center for Infromation Science and Technology, China
[3] University of California, Davis, United States
[1] {gsl16,zhangyic17}@mails.tsinghua.edu.cn
[2] ozj@tsinghua.edu.cn, [3] joyu@ucdavis.edu

Abstract

Neural generative models have achieved promising performance on dialog generation tasks if given a huge data set. However, the lack of high-quality dialog data and the expensive data annotation process greatly limit their application in real-world settings. We propose a paraphrase augmented response generation (PARG) framework that jointly trains a paraphrase model and a response generation model to improve the dialog generation performance. We also design a method to automatically construct paraphrase training data set based on dialog state and dialog act labels. PARG is applicable to various dialog generation models, such as TSCP (Lei et al., 2018) and DAMD (Zhang et al., 2019). Experimental results show that the proposed framework improves these state-of-the-art dialog models further on CamRest676 and MultiWOZ. PARG also significantly outperforms other data augmentation methods in dialog generation tasks, especially under low resource settings. [1] [2]

1 Introduction

Task-oriented dialog systems that are applied to restaurant reservation and ticket booking have attracted extensive attention recently (Young et al., 2013; Wen et al., 2017; Bordes et al., 2016; Eric and Manning, 2017). Specifically, with the progress on sequence-to-sequence (seq2seq) learning (Sutskever et al., 2014), neural generative models have achieved promising performance on dialog response generation (Zhao et al., 2017; Lei et al., 2018; Zhang et al., 2019).

However, training such models requires a large amount of high-quality dialog data. Since each dialog is collected through a human-human or human-machine interaction, it is extremely expensive and time-consuming to create large dialog dataset covering various domains (Budzianowski et al., 2018). After dialogs are collected, we also need to annotate dialog states and dialog acts, which are then used to train language understanding models and learn dialog policy. Hiring crowd-sourcing workers to perform these annotations is very costly. Therefore, we propose automated data augmentation methods to expand existing well-annotated dialog datasets, and thereby train better dialog systems.

We propose to augment existing dialog data sets through paraphrase. Paraphrase-based data-augmentation methods have been proved to be useful in various tasks, such as machine translation (Callison-Burch et al., 2006), text classification (Zhang et al., 2015), question answering (Fader et al., 2013) and semantic parsing (Jia and Liang, 2016). All these approaches first find a set of semantically similar sentences. However, finding isolated similar sentences are not enough to construct a dialog utterances' paraphrase. Because an utterance's paraphrase must fit the dialog history as well. For example, when the system says *"Do you prefer a cheap or expensive restaurant?"*, the user may state his intent of asking for a cheap restaurant by *"Cheap please."* or *"Could you find me a cheap restaurant?"* . However, the latter is obviously an improper response which is not coherent with the system question. In other words, a paraphrased dialog utterance needs to serve the same function as the original utterance under the same dialog context. Therefore, we propose to construct dialog paraphrases that consider dialog context in order to improve dialog generation quality.

We also propose the Paraphrase Augmented Response Generation (PARG), an effective learning

[*] This work was partly done during Silin Gao's summer internship at University of California, Davis.

[1] This work is supported by NSFC (No.61976122), Ministry of Education and China Mobile joint funding (No.MCM20170301).

[2] The code is available at https://github.com/Silin159/PARG

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 639–649
July 5 - 10, 2020. ©2020 Association for Computational Linguistics

framework that jointly optimizes dialog paraphrase and dialog response generation. To obtain dialog paraphrases, we first find all the user utterances that serve the same function in different dialogs, such as different ways of asking for Italian food. Then we select the utterances that have the same semantic content but different surface form, to construct a high-quality dialog paraphrase corpus. The corpus is then used to train a paraphrase generation model to generate additional user utterances. Finally, the augmented dialog data is used to train a response generation model. We leverage the multi-stage seq2seq structure (Lei et al., 2018; Zhang et al., 2019) for both paraphrase and response generation. Moreover, these two models are connected through an additional global attention (Bahdanau et al., 2014) between their decoders, so they can be optimized jointly during training.

In our experiments, we apply our framework on two state-of-the-art models, TSCP (Lei et al., 2018) and DAMD (Zhang et al., 2019) on two datasets CamRest676 (Wen et al., 2017) and MultiWOZ (Budzianowski et al., 2018), respectively. After applying our framework, the response generation models can generate more informative responses that significantly improves the task completion rate. In particular, our framework is extremely useful under low-resource settings. Our paraphrase augmented model only needs 50% of data to obtain similar performance of a model without paraphrase augmentation. Our proposed method also outperforms other data augmentation methods, and its comparative advantage increases in settings where only a small amount of training data is available.

2 Related Work

Data Augmentation has been used in various machine learning tasks, such as object detection (Redmon et al., 2016) and machine translation (Fadaee et al., 2017). It aims to expand training data to improve model performance. In computer vision, many classical data augmentation methods such as random copy (Krizhevsky et al., 2012) and image pair interpolation (Zhang et al., 2017) have been widely used.

However, those approaches are not applicable for natural language processing since language is not spatially invariant like images. The word order in a sentence impacts its semantic meaning (Zhang et al., 2015). Therefore, human language augmentation methods aim to generate samples with the same semantic meaning but in different surface forms. Such an idea led to recent augmentation work on the language understanding task (Hou et al., 2018; Kim et al., 2019; Yoo et al., 2019; Zhao et al., 2019). However, there is no data augmentation work on task-oriented dialog generation.

Paraphrase is the technique that generates alternative expressions. Most of the existing work on paraphrase aims to improve the quality of generated sentences. For example, phrase dictionary (Cao et al., 2017) and semantic annotations (Wang et al., 2019) are used to assist the paraphrase model to improve the language quality. To make a controllable paraphrase model, syntactic information (Iyyer et al., 2018) is also adopted. And, recently, different levels of granularity (Li et al., 2019b) are considered to make paraphrase decomposable and interpretable. In this paper, we utilize a language environment to assist paraphrase, and use paraphrase as a tool to augment the training data of dialog systems.

3 Proposed Framework

In this section, we first introduce how to construct a paraphrase dataset to train paraphrase generation models. Then we describe the work flow of the proposed PARG model.

3.1 Paraphrase Data Construction

We propose a three-step procedure to find dialog utterances that are a paraphrase of each other. First, we perform delexicalization to pre-process dialog utterances to reduce the surface form language variability. Then for each user utterance, we match the utterances in other dialogs that play the same function to find its paraphrase candidates. Finally, we filter out unqualified paraphrases which have a low semantic similarity or a low surface form diversity comparing to the original utterance.

Similar to the delexicalization process introduced in Henderson et al. (2014), we replace the slot values in each utterance by their slot name indicator. For example, the user utterance *"I want a cheap restaurant."* is delexicalized as *"I want a [pricerange] restaurant."*. The slot values can be dropped since their varieties only influence the database search results but have no impact on how the dialog progresses. In other words, no matter whether the user is asking for a cheap or an expansive restaurant, he represents the same intent of requesting a restaurant with a specific price range

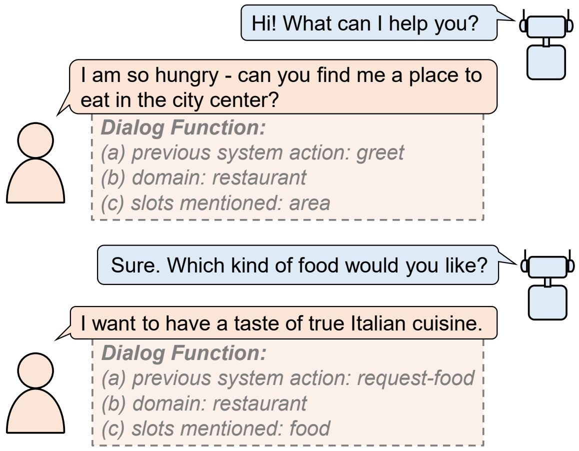
Hi! What can I help you?
I am so hungry - can you find me a place to eat in the city center?
Dialog Function:
(a) previous system action: greet
(b) domain: restaurant
(c) slots mentioned: area
Sure. Which kind of food would you like?
I want to have a taste of true Italian cuisine.
Dialog Function:
(a) previous system action: request-food
(b) domain: restaurant
(c) slots mentioned: food

Figure 1: Illustration of the dialog function of each turn's user utterance.

in the dialog. Therefore through delexicalization, the language variations brought by numerous slot values can be reduced, thus it is easier to find paraphrases.

After delexicalization, we find utterances that play the same role or serve the same *dialog function* in different dialogs. We denote the dialog function of turn t as DF_t. It consists of three types of information: 1) current dialog domain D_t, 2) slots mentioned S_t in the current turn, and 3) system's dialog act A_{t-1} in the previous turn, which is formulated as:

$$DF_t = (D_t, S_t, A_{t-1}) \tag{1}$$

The slots mentioned represent the key information towards task completion, which is the most important information to determine the function of the utterance. The dialog domain is included in the function to avoid ambiguities brought by slots that shared across different domains. For example, asking for the location of a hotel is different from asking for a restaurant. The previous system act is considered to ensure a coherent dialog context, since each turn's user utterance is a reply to the previous system response. Fig.1 gives out an example of dialog function. For each user utterance in the dialog dataset, we go through all the available training data and find all utterances with the same dialog function as paraphrase candidates of it.

As each utterance may have many paraphrase candidates, we only keep the high-quality paraphrase pairs that are similar in semantic but different in surface form. We use the BLEU (Papineni et al., 2002) score and the diversity score proposed in Hou et al. (2018) to evaluate the paraphrase quality. Specifically, if the BLEU score is too low (below 0.2 in our experiments), we consider the paraphrase pair as semantically irrelevant and filter it out. If the diversity score is too low (below 3.4 in our experiments), we also filter out the paraphrase pair since it is too alike in terms of surface form language. For those utterances that do not have any paraphrases after filtering, we gradually reduce the filter threshold of diversity score until each of them matches a paraphrase.

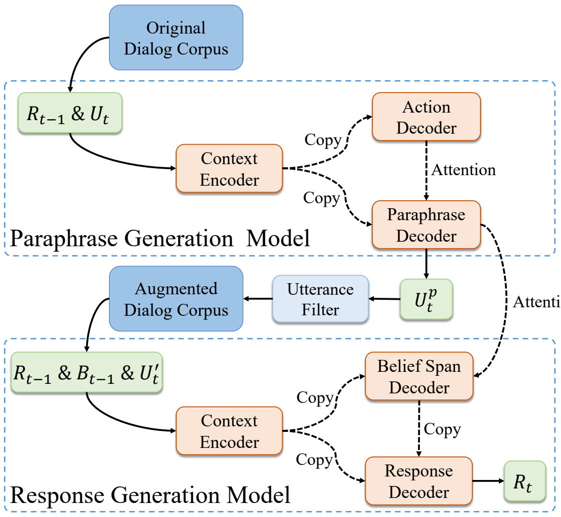

Figure 2: Overview of our Paraphrase Augmented Response Generation (PARG) framework. Solid arrows denote the input or output word sequence. Dash arrows denote hidden states shared between modules. U_t, B_t and R_t represent turn t's user utterance, dialog state and system response respectively. U_t^p represents the paraphrase utterance generated by the paraphrase model. The input of the generation model can be either the generated U_t^p or U_t, denoted as $U_t^{'}$, together with the corresponding dialog state and previous system response.

3.2 Paraphrase Augmented Response Generation

Figure 2 shows an overview of our paraphrase based data augmentation framework. It consists of a paraphrase generation model, a low-quality paraphrase filter and a response generation model. We describe each module in detail below.

Paraphrase Generation Model. Our paraphrase generation model has a seq2seq architecture with a context encoder and two decoders for action decoding and paraphrase decoding. The context encoder takes the concatenation of previous system response R_{t-1} and current user utterance U_t as input and encodes them into hidden states. Then the hidden states are used to decode the previous system action A_{t-1}, where the system action is also a sequence of tokens that first introduced in Zhang et al. (2019). Finally the paraphrase decoder decodes the paraphrase U_t^p based on the hidden states of both the encoder and the action decoder.

$$h^{A_{t-1}} = \text{Seq2Seq}(R_{t-1}, U_t) \tag{2}$$

$$U_t^p = \text{Seq2Seq}(R_{t-1}, U_t | h^{A_{t-1}}) \tag{3}$$

where $h^{A_{t-1}}$ denotes the hidden states of the action decoder. We leverage copy mechanism (Gu et al., 2016) to copy words from input utterances to previous system action and paraphrase. The action decoding process is used to help paraphrase decoding through an attention connection between the decoders, whose significance lies in improving dialog context awareness.

Paraphrase Filter. We then send the generated paraphrase into a filter module to determine if it qualifies as an additional training instance. We aim to keep paraphrases that can serve the same dialog function with the original utterance. So we filter out paraphrases that did not include all of the slots mentioned in the original utterance. Besides, we also filter out paraphrases that have a different meaning and/or a similar surface form compared to the original utterance by the same way in our paraphrase data construction process. We still use 0.2 and 3.4 as the thresholds for BLEU and diversity score respectively in our experiments.

Response Generation Model. We use two state-of-the-art seq2seq model, TSCP (Lei et al., 2018) and DAMD (Zhang et al., 2019) for single domain and multi-domain response generation respectively. We will describe the workflow of our framework based on the TSCP model, as shown in Fig.2. For DAMD the process is similar since the only difference between these two models is that DAMD has an additional action span decoder between the belief span decoder and the response decoder. The model input is the concatenation of the current user utterance $U_t^{'}$, the previous belief span B_{t-1} (slots mentioned by user) and the system response R_{t-1}, where $U_t^{'}$ is either the original user utterance U_t or its paraphrase U_t^p generated by the paraphrase generation model. The model is a two-stage decoding network, where the belief span and system response are decoded sequentially using the copy mechanism. Specifically, we introduce an attention connection between the paraphrase decoder and the belief span decoder to allow the gradient in the response generation model to back-propagate to the paraphrase generation model. So the response generation model can guide the paraphrase decoder to generate better paraphrases and vice versa. This process can be formulated as:

$$h^{B_t} = \text{Seq2Seq}(R_{t-1}, U_t^{'}, B_{t-1} | h^{U_t^p}) \tag{4}$$

$$R_t = \text{Seq2Seq}(R_{t-1}, U_t^{'} | h^{B_t}) \tag{5}$$

where h^{B_t} and $h^{U_t^p}$ denote the hidden states of the belief span decoder and paraphrase decoder respectively.

Training and Evaluation. The model is joint optimized through supervised learning. Specifically, the system action labels, the paraphrase data (collected through the process introduced in the previous section), the dialog state labels and the reference responses are used to calculate the cross-entropy loss of the four decoders, denoted as $loss_a$, $loss_p$, $loss_b$, and $loss_r$, respectively. Then we calculate the sum of all the losses and perform gradient descent for training. The total loss function for training are formulated as:

$$loss = loss_a + loss_p + loss_b + loss_r \tag{6}$$

Note that we only augment user utterance as additional input utterances during training. We alternatively use the original U_t and generated U_t^p as input to the response generation model, while other elements such as belief spans and responses remain the same. Since both decoders are forced to recognize more user expressions, the language understanding and response generation performance improve simultaneously. If the generated U_t^p is in low quality and filtered out, only the original U_t is used to train the response generation model in that turn. This often happens at the beginning of training when the paraphrase model is under-fitting. During testing, only the ground truth user utterances are used as input. However, we still utilize the paraphrase generation model to compute attention between the paraphrase decoder and the belief span decoder. This is because we believe that the paraphrase decoding process can help the belief span decoding process since it provides additional explanations of the user utterance.

4 Experimental Settings

4.1 Datasets and Evaluation Metrics

We conduct our experiments based on two datasets, CamRest676 (Wen et al., 2017) and MultiWOZ

(Budzianowski et al., 2018). Dialogs in both are collected through crowd-sourcing on the Amazon Mechanical Turk platform. Besides experiments on the full datasets, we also conduct experiments using only 20% or 50% of dialog data for training to evaluate the promotion through data augmentation under low-resource settings.

CamRest676 is a single domain dataset consisting of dialogs about restaurant reservation. The dataset has 676 dialogs which are split into training, development and testing set by the ratio of 3:1:1. The average number of turns is 4.06. There are 3 slot types and 99 allowable values in the task ontology. We use three metrics for evaluation following Lei et al. (2018). **Entity Match Rate** (EMR) is the proportion that the system capture the correct user goal. **Success F1** (Succ.F1) score measures whether the system can provide correct information requested by user. While these two metrics are used for evaluating system's task completion ability, we use **BLEU** (Papineni et al., 2002) to evaluate the language fluency of generated responses.

MultiWOZ is a challenging large-scale multi-domain dataset proposed recently (Budzianowski et al., 2018). It consists of dialogs between tourists and clerks at an information center, across seven domains including hotel, restaurant, train, etc. There are 8433/1000/1000 dialogs in training, development and testing set respectively, and the number of turn is 6.85 on average. Meanwhile, MultiWOZ has a complex ontology with 32 slot types and 2,426 corresponding slot values. We use the evaluation metrics proposed by Budzianowski et al. (2018), which are how often the system provides an correct entity (**inform rate**) and answers all the requested information (**success rate**), and how fluent the response is (**BLEU**). We also report a combined score computed via $(Inform + Success) \times 0.5 + BLEU$ for overall quality measure as suggested in (Mehri et al., 2019).

4.2 Implementation Settings

We use a one-layer, bi-directional GRU as the context encoder and two standard GRU as the action decoder and paraphrase decoder. The embedding size and hidden size are both 50 on CamRest676 and 100 for MultiWOZ. The copy mechanism and attention connection are added as shown in Fig.2. For the response generation model, we leverage the state-of-the-art model on each dataset, which is the Two-stage Copy Net (TSCP) (Lei et al., 2018) for CamRest676 and Domain Aware Multi-Decoder (DAMD) (Zhang et al., 2019) for MultiWOZ. We use the model structures that follow the default settings in the open source implementation of TSCP[3] and DAMD[4]. We use the the Adam optimizer (Kingma and Ba, 2014) with a learning rate of 0.003 and 0.005 for CamRest676 and MultiWOZ, respectively. We halve the learning rate when the total loss of our model on development set does not reduce in three consecutive epochs, and we stop the training when the total loss does not reduce in five consecutive epochs. We set the learning rate to 0.0001 and the decay parameter to 0.8 during reinforcement fine tuning in TSCP.

4.3 Baseline Methods

We compare the proposed method with five other data augmentation methods, three of which are based on text replacement and the other two are based on neural paraphrase generation models.

- **WordSub** denotes the rare word substitution method proposed by Fadaee et al. (2017). It generates new sentences by replacing common words with rare ones. A bi-directional LSTM language model is trained to select the proper substitution words. We do not substitute key words associated with slot values to maintain the dialog function of utterances.

- **TextSub** denotes the text span replacement method proposed by Yin et al. (2019). It replaces a sequence of tokens (text span) by their paraphrase candidates from the lexicon database (PPDB (Pavlick et al., 2015)). The selection of text spans is based on a policy network, which is trained jointly with the belief span decoder through reinforcement learning. The slot values are also fixed with the same purpose as in WordSub.

- **UtterSub** denotes the simple utterance replacement augmentation. We use the paraphrases obtained in dialog dataset as new training samples directly instead of training the paraphrase model to generate new samples.

- **NAEPara** denotes a paraphrase model with single encoder-decoder structure. This model, denoted as noising auto-encoder (NAE) in Li

[3] https://github.com/WING-NUS/sequicity

[4] https://gitlab.com/ucdavisnlp/damd-multiwoz

Model	20% Data			50% Data			Full Data		
	BLEU	EMR	Succ.F1	BLEU	EMR	Succ.F1	BLEU	EMR	Succ.F1
TSCP	0.154	0.791	0.806	0.225	0.853	0.817	0.253	0.927	0.854
WordSub	0.140	0.821	0.818	0.212	0.866	0.822	0.239	0.930	0.846
TextSub	0.144	0.834	0.826	0.220	0.895	0.831	0.245	0.942	0.850
UtterSub	0.149	0.826	0.829	0.216	0.876	0.838	0.245	0.938	0.852
NAEPara	**0.155**	0.830	0.831	0.222	0.891	0.843	0.251	0.940	0.855
SRPara	0.154	0.832	0.826	**0.228**	0.886	0.840	**0.254**	0.938	0.852
PARG	**0.155**	**0.852**	**0.849**	0.226	**0.908**	**0.853**	0.252	**0.943**	**0.861**

Table 1: Results on CamRest676. The best scores are in bold.

Model	20% Data				50% Data				Full Data			
	BLEU	Info	Succ	Comb	BLEU	Info	Succ	Comb	BLEU	Info	Succ	Comb
DAMD	0.121	0.779	0.703	0.862	0.169	0.830	0.729	0.948	0.183	0.895	0.758	1.009
WordSub	0.119	0.783	0.712	0.866	0.166	0.821	0.736	0.944	0.176	0.882	0.754	0.994
TextSub	0.123	0.813	0.719	0.889	0.174	0.841	0.741	0.965	0.182	0.890	0.760	1.007
UtterSub	0.112	0.802	0.714	0.870	0.169	0.853	0.737	0.964	0.179	0.893	0.761	1.006
NAEPara	0.126	0.820	0.723	0.898	0.164	0.850	0.750	0.964	0.179	0.893	0.761	1.006
SRPara	**0.130**	0.817	0.725	0.901	**0.175**	0.864	0.753	0.984	0.186	0.903	0.773	1.024
PARG	0.127	**0.825**	**0.739**	**0.909**	0.172	**0.878**	**0.768**	**0.995**	**0.188**	**0.911**	**0.789**	**1.038**

Table 2: Results on MultiWOZ. The best scores are in bold.

et al. (2019a), injects random noise to the encoder's hidden states to improve generation varieties, which has proven to be effective in (Kurata et al., 2016). For model implementation, we use the same GRU nets as in our paraphrase model. And we multiply perturbations, sampled from the uniform distribution between 0.6 and 1.4, to the encoder's hidden states when generating paraphrases.

- **SRPara** denotes a paraphrase model with SR-PB (Wang et al., 2019) structure. In this structure, a semantic parser SLING (Ringgaard et al., 2017) is used to analyze the semantic frame of an utterance and the semantic role of each token in it. Then the sequences of token, semantic frame labels and semantic role labels are fed into three parallel encoders separately. The outputs of the three encoders are projected through a linear layer, and then sent to a decoder to generate the paraphrase. The implementation of encoders and the decoder is the same as NAEPara.

We utilize the same dataset (CamRest676 or MultiWOZ) to train all the models for fair comparison. Specifically, we use all the user utterances in the training corpus of CamRest676 or MultiWOZ to train the LSTM language model of WordSub and the policy network of TextSub. And we use the same paraphrase data constructed in 3.1 to train the paraphrase models in NAEPara and SRPara.

5 Results and Analysis

The experimental results on CamRest676 and MultiWOZ are shown in Table 1 and Table 2, respectively. In both tables, the first line is the baseline results without data augmentation, the second to sixth lines are results obtained by different data augmentation methods (substitution-based or paraphrase-based), and the last line is the performance of our proposal. The results are grouped into three columns according to the size of training data (20%/50%/full).

We observe some common conclusions supported by the experimental results on both datasets. First, our proposed data augmentation framework significantly improves the system's task completion ability (EMR, Succ.F1, Info and Succ) consistently without harming the language fluency. This indicates that incorporating additional dialog paraphrases is beneficial for learning informative responses, since more user expressions are seen by the model during training.

Secondly, our framework outperforms other data augmentation methods in terms of dialog task relevant metrics under all circumstances. In particular, paraphrase based methods are more likely to produce more fluent and informative responses than local substitution methods (WordSub and TextSub), because neural generative models consider dialog history to generate more coherent utterances. The improvement of PARG over UtterSub suggests that our paraphrase generation model provides a more robust way of utilizing the additional information contained in paraphrases. Our paraphrase generation model outperforms other paraphrase based methods (NAEPara and SRPara) since the decoding process of previous system action and the gradient back-propagation through the belief span decoder provide strong dialog context information for paraphrase generation.

Thirdly, the less data is available, the more improvement can be achieved through our data augmentation. It is worth noting that after applying PARG, the model trained on only 50% data obtain comparable results to the model trained on the full dataset without data augmentation, in terms of task relevant metrics. The similar results are also observed by comparing the models trained on 20% data with augmentation and 50% data without augmentation. This indicates that our method is of great significance under low resource settings.

PARG sometimes gets a slightly lower BLEU score compared to other methods. This is potentially because that although seq2seq models can learn responses which corresponding to a correct action, the surface language can still vary among training and testing utterances due to the natural variety of human languages. Therefore, the BLEU score, which measures the likeness of surface language, may drop despite the system generate good functional responses.

We also observe some diverse results on CamRest676 and MultiWOZ. Under the full data setting, the improvement gained by our data augmentation method on CamRest676 is lower than on MultiWOZ, since the single domain task in CamRest676 is easy and the data is enough for model training without conducting augmentation. While for MultiWOZ, due to large language variations and the complex ontology, the utterance space is not well-explored, thus the response generation process can benefit more through incorporating additional dialog data.

6 Ablation Study

In this section we investigate the function of each component in our paraphrase augmented response generation framework. In particular, we discard 1) the act decoder (PARG w/o Act), 2) the utterance filter (PARG w/o Filt) or 3) joint training (PARG w/o Join) one at a time, then do model training and evaluation on the full MultiWOZ dataset. The results are shown in Table 3.

Model	BLEU	Info	Succ	Comb
DAMD	0.183	0.895	0.758	1.009
PARG	0.188	0.911	0.789	1.038
PARG w/o Filt	0.173	0.887	0.765	0.999 (-0.039)
PARG w/o Act	0.180	0.897	0.763	1.010 (-0.028)
PARG w/o Join	0.185	0.905	0.782	1.028 (-0.010)

Table 3: Ablation results on MultiWOZ. The changes of combine score compared to PARG are shown in parentheses.

We observe that removing the utterance filter brings the biggest drop in response quality in terms of combined score (-0.039). This suggests the importance of using only high-quality paraphrases to train the response generation model, because the ill generation utterances will introduce errors to the downstream model. The model also suffers from a performance drop (-0.028) after removing the previous system action decoder, which indicates that the supervision from previous system action labels is beneficial for generating better paraphrases. Finally, we train the paraphrase generation model and response generation model separately and observe a slight drop of combined score (-0.010). This is because through the attention connection between the paraphrase decoder and belief span decoder, the loss computed for response generation can also guide the paraphrase generation model to generate paraphrases that directly benefit to the response generation process. Although the improvement is relatively marginal, joint training has additional advantages in simplifying the training process. Specifically, we only need to conduct a single run of training and optimize a single set of hyperparameters.

7 Case Study and Error Analysis

We conduct several case studies to illustrate the response generation quality, paraphrase generation quality, as well as errors made by our model.

Table 4 compares the dialog state and system response generated by the original model TSCP to those generated by PARG. We investigate the results from both the 50% and the full scale CamRest676 experiments, to further show our framework's superiority in low resource scenarios. On full training data, TSCP and PARG both generate correct dialog state slots. However, TSCP generates a wrong question "Would you like something different?", as if no restaurant satisfies the user's request. While PARG generates an appropriate

User Utterance: Can you help me find a restaurant in the south that doesn't cost a lot of money. Ground Truth Dialog State: pricerange=cheap, area=south Reference Response: Nandos is a nice place, it serves Portuguese food. Is there anything else?		
Full Data	TSCP	Generated Dialog State: pricerange=cheap, area=south Generated Response: Nandos is a restaurant in the south. Would you like something different?
	PARG	Generated Dialog State: pricerange=cheap, area=south Generated Response: Nandos is a Portuguese restaurant in the south. Anything else you need?
50% Data	TSCP	Generated Dialog State: area=south Generated Response: Taj Tandoori is an Indian restaurant, it is in the expensive price range.
	PARG	Generated Dialog State: pricerange=cheap, area=south Generated Response: Nandos serves Portuguese food. Would you like the address?

Table 4: Comparison of response generation results before and after applying our paraphrase augmented method. Models trained on full data and 50% data are compared respectively.

Dialog Function	Utterance Paraphrase
Domain: train Slots Mentioned: leave Previous System Act: request-leave	Previous Response: What time would you like to leave from norwich? Original Utterance: I would like to leave at 14:45. What is the price? Matched Paraphrase: 14:45, please. What is the duration of the train ride?
Domain: hotel Slots Mentioned: parking Previous System Act: inform-name	Previous Response: Acorn Guest House is available if that works for you. Original Utterance: That is good. And I need a free parking, does it have? Matched Paraphrase: This place is fine. Is it near a hotel with free parking?

Table 5: Examples of ill-matched paraphrase pairs obtained by our paraphrase matching method.

Original Utterance: I need an inexpensive restaurant on the north side.	
TextSub	I'm looking for place inexpensive restaurant is located in the north.
SRPara	Please find me an inexpensive restaurant in the north part of the town.
PARG	Can you recommend me a cheap restaurant in the north area.

Table 6: Paraphrased utterances generated by different methods.

question “Anything else you need?” to ask user for further request about the recommended restaurant. When we reduce the training data to half, TSCP generates wrong dialog state slots, and therefore recommends an expensive restaurant. But PARG does not suffer from this problem and generates a correct response. This example suggests that PARG can effectively improve the quality of dialog generation in low resource settings.

Although our paraphrase augmented data augmentation framework shows a notable superiority on the dialog generation quality, it still has some limitations. Table 5 shows some errors that PARG made in our paraphrase data construction process. In the first case, the question “What is the price?” raised by the original utterance doesn't match the question “What is the duration of the train ride?” in the paraphrase. This error is made since we do not have user act labels in the dialog datasets. Defining the dialog function of user utterance more precisely by adding its user act can solve this problem. Another incoherence of paraphrase sources from the switch of dialog domains in multi-domain dialogs. In the example, the word “place” in the paraphrase refers to another site irrelevant to the hotel in the previous system response, which might be an attraction or a restaurant. The domain of the previous turn should also be considered in the dialog function to provide more domain information, which is regarded as a potential solution for this issue.

We also compare the utterances generated by different data augmentation methods to show the superiority of PARG in terms of paraphrase generation quality. We select TextSub and SRPara for comparison, since they are the best replacement-based and paraphrase-based methods achieving the highest combined scores on MultiWOZ respectively. Table 6 shows an example of paraphrases generated by the three methods. We find that the paraphrase generated by TextSub is of bad quality because it is not in accordance with normal grammar, while the paraphrase generated by SRPara is fluent and semantically similar to the original utterance. However, the paraphrase generated by our proposed PARG has higher quality. It flexibly changes the rare word “inexpensive” to the common word “cheap”, which enlarges the surface form diversity. The high-quality paraphrases can give better guidance to the downstream response generation model, which explains the significant improvement in terms of task completion rate obtained by PARG.

8 Human Evaluation

We conduct human evaluation to further illustrate PARG's superiority in terms of paraphrase generation. We use one-to-one comparison to evaluate the relative quality of paraphrases generated by PARG versus strong baselines (NAEPara and SRPara).

In our experiments, we advise the judges to evaluate the quality of a paraphrase according to its similarity of user intent with the original utterance. We sample one hundred dialog turns. And in each turn, the paraphrase generated by PARG is given one-to-one comparisons with each baseline's paraphrase by five judges. Specifically, we ask the judges to choose whether the paraphrase generated by PARG is of better, equal or worse quality than the paraphrase generated by NAEPara or SRPara, given the original utterance.

Comparison	Better%	Equal%	Worse%
PARG vs. NAEPara	59.2%	18.4%	22.4%
PARG vs. SRPara	55.4%	20.8%	23.8%

Table 7: Human evaluation results.

The results are shown in Table 7. We report the percentage of different choices made by the judges in each one-to-one comparison, including the percentage of cases that PARG generates better (Better%), so-so (Equal%), or worse (Worse%) paraphrases. We observe that PARG generates better paraphrases in a large proportion of cases, no matter compared to NAEPara or SRPara. This suggests that PARG outperforms both NAEPara and SRPara in terms of paraphrase generation quality, which further proves that the dialog data augmented by PARG can provide better guidance to the response generation tasks.

9 Conclusion

In this paper, we propose to use dialog paraphrase as data augmentation to improve the response generation quality of task-oriented dialog systems. We give out the definition of the paraphrase for a dialog utterance and design an approach to construct paraphrase dataset from a dialog corpus. We propose a Paraphrase Augmented Response Generation (PARG) framework which consists of a paraphrase generation model, an utterance filter and a response generation model, where the models are trained jointly to take fully advantage of the paraphrase data for better response generation performance. Our framework achieves significant improvements when it is applied to state-of-the-art response generation models on two datasets. It also beats other data augmentation methods, especially under the low-resource settings.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

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, Inigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. *arXiv preprint arXiv:1810.00278*.

Chris Callison-Burch, Philipp Koehn, and Miles Osborne. 2006. Improved statistical machine translation using paraphrases. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 17–24. Association for Computational Linguistics.

Ziqiang Cao, Chuwei Luo, Wenjie Li, and Sujian Li. 2017. Joint copying and restricted generation for paraphrase. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Mihail Eric and Christopher D Manning. 2017. Key-value retrieval networks for task-oriented dialogue. *arXiv preprint arXiv:1705.05414*.

Marzieh Fadaee, Arianna Bisazza, and Christof Monz. 2017. Data augmentation for low-resource neural machine translation. *arXiv preprint arXiv:1705.00440*.

Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. 2013. Paraphrase-driven learning for open question answering. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1608–1618.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. *arXiv preprint arXiv:1603.06393*.

Matthew Henderson, Blaise Thomson, and J. Steve Young. 2014. Robust dialog state tracking using delexicalised recurrent neural networks and unsupervised adaptation. *Spoken Language Technology Workshop*, pages 360–365.

Yutai Hou, Yijia Liu, Wanxiang Che, and Ting Liu. 2018. Sequence-to-sequence data augmentation for dialogue language understanding. *arXiv preprint arXiv:1807.01554*.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. *arXiv preprint arXiv:1804.06059*.

Robin Jia and Percy Liang. 2016. Data recombination for neural semantic parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12–22.

Hwa-Yeon Kim, Yoon-Hyung Roh, and Young-Gil Kim. 2019. Data augmentation by data noising for open-vocabulary slots in spoken language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 97–102.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Gakuto Kurata, Bing Xiang, and Bowen Zhou. 2016. Labeled data generation with encoder-decoder lstm for semantic slot filling. In *INTERSPEECH*, pages 725–729.

Wenqiang Lei, Xisen Jin, Min-Yen Kan, Zhaochun Ren, Xiangnan He, and Dawei Yin. 2018. Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1437–1447.

Juntao Li, Lisong Qiu, Bo Tang, Dongmin Chen, Dongyan Zhao, and Rui Yan. 2019a. Insufficient data can also rock! learning to converse using smaller data with augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6698–6705.

Zichao Li, Xin Jiang, Lifeng Shang, and Qun Liu. 2019b. Decomposable neural paraphrase generation. *arXiv preprint arXiv:1906.09741*.

Shikib Mehri, Tejas Srinivasan, and Maxine Eskenazi. 2019. Structured fusion networks for dialog. *arXiv preprint arXiv:1907.10016*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2015. Ppdb 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 425–430.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788.

Michael Ringgaard, Rahul Gupta, and Fernando CN Pereira. 2017. Sling: A framework for frame semantic parsing. *arXiv preprint arXiv:1710.07032*.

I Sutskever, O Vinyals, and QV Le. 2014. Sequence to sequence learning with neural networks. *Advances in NIPS*.

Su Wang, Rahul Gupta, Nancy Chang, and Jason Baldridge. 2019. A task in a suit and a tie: paraphrase generation with semantic augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7176–7183.

TH Wen, D Vandyke, N Mrkšíc, M Gašíc, LM Rojas-Barahona, PH Su, S Ultes, and S Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. In *15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017-Proceedings of Conference*, volume 1, pages 438–449.

Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, and Qun Liu. 2019. Dialog state tracking with reinforced data augmentation. *arXiv preprint arXiv:1908.07795*.

Kang Min Yoo, Youhyun Shin, and Sang-goo Lee. 2019. Data augmentation for spoken language understanding via joint variational generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7402–7409.

Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. 2013. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.

Yichi Zhang, Zhijian Ou, and Zhou Yu. 2019. Task-oriented dialog systems that consider multiple appropriate responses under the same context. *arXiv preprint arXiv:1911.10484*.

Tiancheng Zhao, Allen Lu, Kyusong Lee, and Maxine Eskenazi. 2017. Generative encoder-decoder models for task-oriented spoken dialog systems with chatting capability. *arXiv preprint arXiv:1706.08476*.

Zijian Zhao, Su Zhu, and Kai Yu. 2019. Data augmentation with atomic templates for spoken language understanding. *arXiv preprint arXiv:1908.10770*.