Adaptive Natural Language Generation for Task-oriented Dialogue via Reinforcement Learning

Atsumoto Ohashi **Ryuichiro Higashinaka**
Graduate School of Informatics, Nagoya University
ohashi.atsumoto.c0@s.mail.nagoya-u.ac.jp
higashinaka@i.nagoya-u.ac.jp

Abstract

When a natural language generation (NLG) component is implemented in a real-world task-oriented dialogue system, it is necessary to generate not only natural utterances as learned on training data but also utterances adapted to the dialogue environment (e.g., noise from environmental sounds) and the user (e.g., users with low levels of understanding ability). Inspired by recent advances in reinforcement learning (RL) for language generation tasks, we propose ANTOR, a method for **A**daptive **N**atural language generation for **T**ask-**O**riented dialogue via **R**einforcement learning. In ANTOR, a natural language understanding (NLU) module, which corresponds to the user's understanding of system utterances, is incorporated into the objective function of RL. If the NLG's intentions are correctly conveyed to the NLU, which understands a system's utterances, the NLG is given a positive reward. We conducted experiments on the MultiWOZ dataset, and we confirmed that ANTOR could generate adaptive utterances against speech recognition errors and the different vocabulary levels of users.

1 Introduction

In task-oriented dialogue systems, the role of the natural language generation (NLG) component is to convert a system's intentions, called dialogue acts (DAs), into natural language utterances and to convey DAs accurately to users (McTear, 2002; Gao et al., 2019). In recent years, data-driven language generation methods (Wen et al., 2015; Peng et al., 2020) using neural networks have been introduced to NLG for task-oriented dialogue systems, enabling natural utterance generation.

When such NLG is implemented in a realistic environment, however, it is essential to generate not only natural utterances as learned on training data but also utterances adapted to the dialogue environment and the user. For example, when interacting in a noisy environment, such as in a place with loud background noise or through a telephone, the system needs to use sentences and vocabulary that are less likely to be misrecognized. In addition, if the user is a child or a second language learner, it is necessary to generate utterances in plain terms that the user can easily understand. Therefore, it is essential for the NLG module to adaptively generate utterances for the dialogue environment and the user in real-world situations. However, it is challenging to implement optimal NLG using only supervised learning because it is not practical to create training data for every environment or user. Recently, for many generative tasks, such as machine translation, summary generation, and dialogue generation in open domains, many methods using reinforcement learning (RL) have been proposed. In these studies, non-differentiable objective functions, such as generated text quality and subjective user preferences, are used to optimize the language generation model and achieve high performance.

With this background in mind, this study proposes a method for **A**daptive **N**atural language generation for **T**ask-**O**riented dialogue via **R**einforcement learning (ANTOR)[1], which adapts to the dialogue environment and the user. In our method, a reward function using a natural language understanding (NLU) model is set up, and a pre-trained NLG model is fine-tuned by using RL. That is, the NLG generates a system utterance for a given DA, and the NLU provides a positive reward if it can successfully recognize the original DA from the utterance. Experiments using the MultiWOZ dataset (Budzianowski et al., 2018) are conducted with multiple environments and users simulating real-world conditions, such as speech recognition errors and the different vocabulary levels of users.

[1] In this paper, ANTOR refers to both the method of fine-tuning NLG and the fine-tuned NLG model. Our code and data are publicly available at https://github.com/nu-dialogue/antor

Proceedings of the 29th International Conference on Computational Linguistics, pages 242–252
October 12–17, 2022.

Our contribution is threefold:

- We propose ANTOR, a method for fine-tuning NLG for task-oriented dialogue via reinforcement learning. We conducted experiments using MultiWOZ to confirm that ANTOR can generate adaptive utterances for multiple NLUs with different model architectures.
- Experiments were conducted in a noisy environment where speech recognition errors caused by background noise were simulated. The results show that ANTOR could generate utterances with words less likely to cause speech recognition errors.
- Experiments were conducted using NLUs that simulated users with low vocabulary levels. The results confirmed that ANTOR was able to generate utterances using vocabulary appropriate for each vocabulary level.

2 Related Work

2.1 Natural Language Generation for Task-oriented Dialogue

Conventional NLG for task-oriented dialogues had used template-based and rule-based methods (Walker et al., 2002; Stent et al., 2004). There, templates and rules had to be carefully designed manually by experts in each domain. Later, a data-driven method using machine learning was proposed (Oh and Rudnicky, 2002; Angeli et al., 2010; Mairesse and Young, 2014). Kondadadi et al. (2013) proposed a method for statistically generating utterances using k-means clustering and support vector machines. Recently, many generation models based on end-to-end learning have been proposed by using deep learning (Wen et al., 2016; Tran and Nguyen, 2017; Su et al., 2018). Wen et al. (2015) proposed SC-LSTM, which controls utterance generation by using DA feature vectors and reading gates. SC-GPT (Peng et al., 2020) is a state-of-the-art model for MultiWOZ that achieves high performance by fine-tuning the language model GPT-2 (Radford et al., 2019) on a large number of task-oriented dialog datasets.

Some end-to-end models (Budzianowski et al., 2018; Chen et al., 2019) generate system utterances directly from a dialogue history instead of using NLG, which is known as a word-level policy. In particular, Zhao et al. (2019) and Mehri et al. (2019) optimize the word-level policy by RL to improve task completion. Although these methods use RL to generate system utterances, they do not deal with the NLG module itself and ways to make it adaptive to environments and users.

2.2 Adaptive Natural Language Generation for Task-oriented Dialogue

Methods have been proposed for generating utterances adapted to the user. Walker et al. (2004) used quantitative user modeling for multimodal dialogue to achieve speech production that takes user preferences into account. Janarthanam and Lemon (2010) used RL to create utterances that suit the user's domain knowledge. Dušek and Jurčíček (2016) proposed an NLG that can generate utterances exhibiting entrainment. Furthermore, Mairesse and Walker (2010) proposed PERSONAGE, an NLG that can generate utterances expressing Big Five personality traits. Our study differs from the above studies in that we optimize an existing NLG for the specific objective function of accurately conveying DAs for specific environments and users.

2.3 Natural Language Generation with Reinforcement Learning

In recent years, many methods have been proposed that use RL for language generation tasks (Luketina et al., 2019). There are machine translation methods (Wu et al., 2016; Bahdanau et al., 2016) using BLEU as the reward function, summary generation methods (Ranzato et al., 2015; Dong et al., 2018) using ROUGE, and story generation (Tambwekar et al., 2019). In addition, human feedback rather than automatic evaluation metrics is also used in many methods including machine translation (Kreutzer et al., 2018), summary generation (Ziegler et al., 2019; Stiennon et al., 2020), and open-domain dialogue (Hancock et al., 2019; Jaques et al., 2019). Our study examines the applicability of these recent advances to NLG in task-oriented dialogues.

3 Method

3.1 Task Overview

Figure 1 shows the overall task performed by NLG in this study. First, NLG takes a reference DA, representing system intentions, converts it into natural language, and outputs a system utterance. The user's NLU then predicts the system's DA (predicted DA) from the system utterance. The goal of ANTOR is to generate utterances such that the pre-

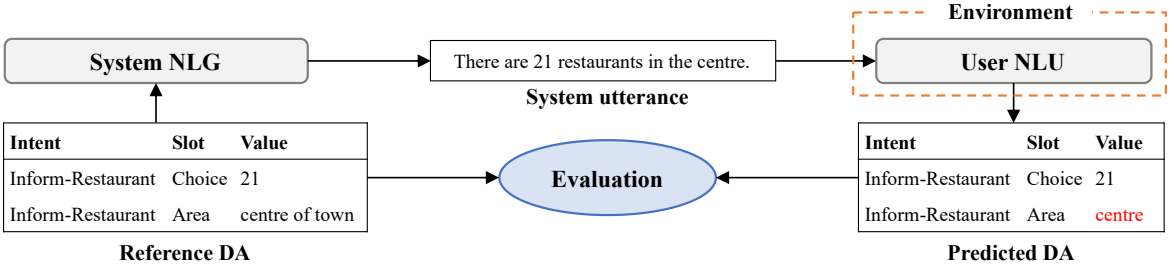

Figure 1: Overview of the task to be performed by NLG. NLG generates system utterance corresponding to reference DA. NLG is evaluated using reference DA and predicted DA, which NLU estimates from system utterance. The ability of User NLU to understand can vary and so can the environment.

dicted DA estimated by the NLU becomes equivalent to the reference DA. Note that this study uses automatic evaluation by comparing the reference DA and predicted DA; more down-to-earth evaluations using user subjective evaluations or user models are left for future work. In the following, the main concepts of the task, namely, DA, system NLG, user NLU, and evaluation, are described.

Dialogue Act The DA is a semantic representation of a system utterance. The reference DA A contains one or more triples consisting of intent I, slot s, and value v:

$$A = \{(I_1, s_1, v_1), ..., (I_{|A|}, s_{|A|}, v_{|A|})\}$$

I represents a system's intention in a domain. For example, in the restaurant domain, there are intentions such as "inform" and "request" (e.g., "Restaurant-Inform," "Restaurant-Request"). s and v indicate the category (e.g., "Choice" and "Area") and specific information belonging to s, respectively. The first line of the reference DA in Figure 1 indicates the semantics that there are 21 possible restaurants.

System NLG NLG generates a system utterance U on the basis of a given A. In this study, we assume a generative model with neural networks. Using the chain rule, a joint probability over $[A; U] = (x_1, ..., x_N)$ is modeled by a neural network ρ_θ with parameters θ:

$$\rho_\theta([A; U]) = \prod_{n=1}^{N} \rho_\theta(x_n | x_{<n}) \quad (1)$$

where N is the length of $[A; U]$. θ is trained by maximizing the log-likelihood (MLE) over dataset $D = \{[A_1; U_1], ..., [A_{|D|}; U_{|D|}]\}$:

$$\mathcal{L}(D) = \sum_{t=1}^{|D|} \sum_{n=1}^{N_t} \log \rho_\theta(x_n^t | x_{<n}^t) \quad (2)$$

where N_t is the length of $[A_t; U_t]$.

User NLU NLU predicts DA A' from U output by NLG. The structure of A' is the same as that of A. In this study, we assume a classification-based prediction model for intent detection and slot tagging. Intent detection performs multi-label classification of an utterance, and slot tagging categorizes each token in an utterance as to which slot it belongs. The training data D for NLG is the same as that for training NLU.

Evaluation The goal of NLG is to generate U such that $A = A'$. Therefore, the rate of concordance between A and A' is used to evaluate the NLG. Specifically, we use the F1 score calculated from true positive triples $A^{TP} = A \cap A'$, false negative triples $A^{FN} = A \cap \overline{A'}$, and false positive triples $A^{FP} = \overline{A} \cap A'$. In addition, following (Wang et al., 2020), Accuracy $= \frac{|A^{TP}|}{|A^{TP}|+|A^{FP}|+|A^{FN}|}$ is also used.

3.2 Fine-tuning via Reinforcement Learning

ANTOR is optimized by fine-tuning NLG pre-trained by MLE in Eq. (2) via RL. We use proximal policy optimization (PPO) (Schulman et al., 2017) for the RL algorithm. We initialize policy π_ϕ by using ρ_θ and add a randomly initialized linear layer that outputs a scalar value for a value network. Parameters ϕ are updated on the basis of the clipped surrogate objective $\mathcal{L}^{CLIP}(\phi)$. We incorporate an understanding of NLU into the reward for ANTOR. When computing the reward r, each utterance U generated by π_ϕ from $A \sim D$ is evaluated using A' predicted by NLU from U as follows:

$$r(A, A') = \mathrm{F1}(A, A') \frac{1}{|A^{TP}|} \sum_{(I,s,v) \in A^{TP}} \mathrm{idf}_D(I, s) \quad (3)$$

where $\mathrm{idf}_D(I, s)$ is the IDF value of (I, s) computed over all intent-slot pairs in D. This weighting

of F1 scores compensates for DAs that frequently occur in D (e.g., greetings) and DAs that occur infrequently.

Following Ziegler et al. (2019), to prevent π_ϕ from moving too far from ρ_θ, a penalty by Kullback–Leibler (KL) divergence is added to $r(A, A')$ as the final reward R:

$$R(A, A', U) = r(A, A') - \beta \log \frac{\pi_\phi(U|A)}{\rho_\theta(U|A)} \quad (4)$$

where β is the coefficient for the penalty. Algorithm 1 summarizes the fine-tuning process of ANTOR.

Algorithm 1 ANTOR with PPO

Require: Dataset D; NLU; Policy ρ_θ pre-trained via MLE by Eq. (2)

Initialize policy $\pi_{\phi_{old}} = \rho_\theta$
Randomly initialize value network in $\pi_{\phi_{old}}$
for i = 1, 2, ..., max iteration **do**
  for j = 1, 2, ..., batch size **do**
    Sample a reference DA A from D
    Sample an utterance U from A by $\pi_{\phi_{old}}$
    Get a predicted DA A' from U by NLU
    Compute reward $R(A, A', U)$ by Eq. (4)
    Compute advantage estimates
  end for
  Optimize $\mathcal{L}^{CLIP}(\phi)$, with pre-determined number of epochs and minibatch size
  $\phi_{old} \leftarrow \phi$
end for

4 Environment

We aim to confirm the feasibility of NLG that can robustly respond to the dialogue environment and the user, which does not exist in typical NLG training data. Therefore, we simulate two conditions in the following subsections. Note that D in this section is the same as the data for training NLG in Section 3.1; $D = \{[U_1; A_1], ..., [U_{|D|}; A_{|D|}]\}$.

4.1 Speech Recognition Error

When a dialogue system interacts with a user via voice, it is assumed that background noise makes it difficult for system utterances to be accurately conveyed to the user. Automatic speech recognition (ASR) error simulation is often used to construct a noisy channel between a user and system (Schatzmann et al., 2007; Fazel-Zarandi et al., 2019; Wang et al., 2020). Therefore, we apply perturbations that take the background noise into account to an utterance from NLG by using ASR error simulation. The noisy utterance is then used as input to NLU. Word substitution with a confusion matrix is used in the simulation (Figure 2). The TTS-ASR pipeline (Park et al., 2019) is used to construct the confusion matrix with the following procedure:

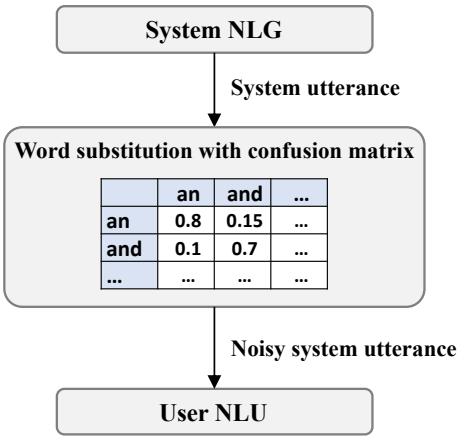

Figure 2: ASR error simulation to add noise to system utterance

1. Convert each U over D to audio data U^{Audio} using a text-to-speech (TTS) system.

2. Create $U^{NoisyAudio}$ by adding background noise to each U^{Audio}.

3. Recover each $U^{NoisyAudio}$ into text U^{Noisy} using an ASR system. This creates $D^{Noisy} = \{U_1^{Noisy}, ..., U_{|D|}^{Noisy}\}$.

4. Align words in U and U^{Noisy} with the Levenshtein distance and calculate the frequency of each word substituted and deleted from U to U^{Noisy}, resulting in an N-dimensional confusion matrix $M \in N^2$. N is the size of vocabulary $V = \{v_1, ..., v_N\}$ appearing in D or D^{Noisy}. Note that a special token denoting deletion is included in V.

Here, $M(i, j)$ indicates how often a word v_i is substituted into v_j. When simulating ASR errors, each word w in a system utterance is replaced by a word v_j according to the following probability:

$$p_w(v_j) = \begin{cases} \frac{M(i,j)}{\sum_{n=1}^{N} M(i,n)} & \text{if } \exists v_i \in V : w = v_i, \\ 0 & \text{otherwise} \end{cases}$$

4.2 Different Vocabulary Levels

In a real environment in which a dialogue system interacts, the users may not have a sufficient vocabulary, such as when they are children or second language learners. Therefore, NLG should use vocabulary and sentences appropriate to the user's vocabulary level. We can simulate the user's vocabulary level by adjusting the training data D for NLU as follows:

1. Prepare a word list $L = \{v_1, ..., v_{|L|}\}$ of the desired vocabulary level.
2. For each $[A; U] \in D$, if the lemma of a non-stop word[2] in U is not in L, the $[A; U]$ is excluded from D.

Using the adjusted training data, it is expected that the NLU can understand only the words in L.

5 Experiments

We wanted to confirm that ANTOR is capable of generating utterances adapted to the dialogue environment and the user. To verify the effectiveness of ANTOR, we conducted experiments using simulations.

5.1 Dataset

We used the MultiWOZ dataset (Budzianowski et al., 2018), which is a task-oriented dialogue dataset between a clerk and a tourist at a tourist information center. The dataset contains 10,438 dialogues in seven domains. We used only system utterances annotated with the clerk's DAs. We used a total of 56,750 utterances and DA pairs in the training data of MultiWOZ to train NLG and NLU. In addition, we also used the utterances to construct the confusion matrix used in the ASR error simulation.

5.2 Training Setup

ANTOR The 117M parameter version of the GPT-2 language model (Radford et al., 2019) was used as a base model. DAs were input to the model as a sequence of intent, slot, and value triples connected by the symbols "+" and "*"; if there were multiple triples, they were connected by commas ",". In addition, to control generation, special tokens "[ACT]" and "[RSP]" were added at the beginning of the DAs and the system utterance sequences, respectively, in order to indicate the start of each sequence. The following is an example input to the model:

```
[ACT] Inform-Restaurant + Choice
* 21, Inform-Restaurant + Area *
centre of town [RSP] There are
21 restaurants in the centre of
town.
```

In MLE, GPT-2 was trained on MultiWOZ for five epochs with a batch size of 8, following Peng et al. (2020). We used the Adam optimizer (Kingma and Ba, 2014) with a learning rate of 5e-5, and the learning rate decreased linearly with the number of steps.

For RL, 60 iterations were trained with a batch size of 1,024 (i.e., 1,024 utterances), and each batch was trained in 4 epochs with a minibatch size of 1. The coefficient β of the penalty for the KL divergence was set to 0.1. We use generalized advantage estimation (Schulman et al., 2015) (GAE) with a γ of 1.0 and λ of 0.95. The Adam optimizer was used with a learning rate of 5e-6, and the learning rate decreased linearly with the number of steps.

For fair evaluation, we trained ANTOR with five different random seeds. 7,372 pairs of DAs and system utterances from MultiWOZ test data were used for testing. The average of the five trials was used as the final score. A greedy search was used for utterance generation in a test.

User NLU As NLUs in our experiments, following the work of Liu et al. (2021), who evaluated NLUs with task-oriented dialogues, we used two models, MILU (Hakkani-Tür et al., 2016) and BERT (Devlin et al., 2019). Each model was trained by using pairs of DAs and system utterances from MultiWOZ training data. The learning rates were 1e-3 for MILU and 1e-4 for BERT as in (Liu et al., 2021).

5.3 Baselines

To evaluate the performance of ANTOR, we used three comparison models.

SC-LSTM (Wen et al., 2015) An LSTM-based method for controlling utterance generation with feature vectors related to DAs. We used a model pre-trained with MultiWOZ, which is available from ConvLab-2 (Zhu et al., 2020), a platform for task-oriented dialogue systems.

SC-GPT (Peng et al., 2020) A GPT-2 based model that has been trained on a large number

[2]We used the Python library Spacy for word tokenization, stop word determination, and word lemmatization.

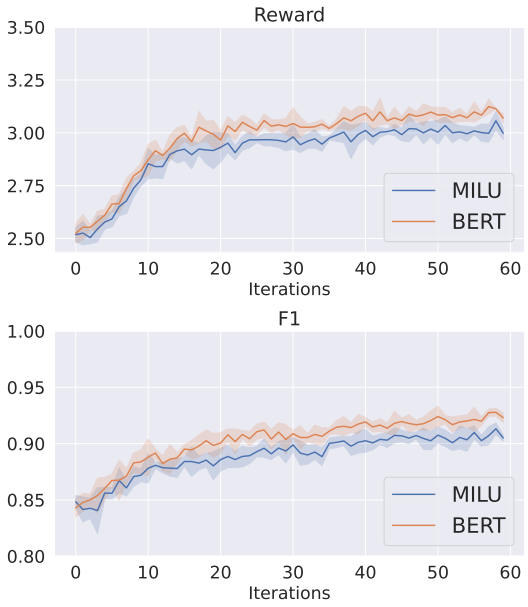

Figure 3: Increase in reward and F1 when ANTOR was trained in a clean environment using MILU and BERT, respectively.

of task-oriented dialogue datasets and further fine-tuned on MultiWOZ. In fine-tuning, training was done for five epochs with a batch size of 8, as reported in the official repository[3].

GPT-2 (Radford et al., 2019) A GPT-2 model fine-tuned on MultiWOZ using only MLE. The hyperparameters and input format were the same as those of ANTOR.

5.4 Experimental Procedure

The experiment was conducted in three stages using both MILU and BERT. First, we checked the effectiveness of ANTOR in clean environments with basic task-oriented dialogue. Next, we conducted two experiments: (1) in an ASR error simulation environment and (2) using NLUs trained only with low vocabulary levels.

5.5 Results in Clean Environment

Figure 3 shows the reward and F1 transition of ANTOR, indicating that the scores increased steadily. Table 1 shows the test scores for each model, indicating that ANTOR's accuracy and F1 outperformed the other models. These results show that ANTOR can learn utterance generation that fits both models of MILU and BERT. Note that the BLEU score of ANTOR was lower than those of SC-GPT and GPT-2. This BLEU score was calculated by comparing the utterances in MultiWOZ as references and the utterances generated by NLG as hypotheses. This means that ANTOR no longer generated utterances that appear in MultiWOZ in order to generate utterances tailored to NLU.

Model	MILU		BERT		
	Acc.	F1	Acc	F1	BLEU
SC-LSTM	74.0	78.6	73.6	77.7	25.3
SC-GPT	77.3	81.1	78.3	82.2	**29.9**
GPT-2	79.5	84.0	79.7	83.6	**29.9**
ANTOR (ours)	**86.7**	**89.8**	**87.8**	**90.7**	27.5

Table 1: Scores for each NLG model evaluated using MILU and BERT, respectively.

SNR	WER	Sub.	Ins.	Del.
0	30.4%	15.4%	8.5%	6.6%
5	23.9%	12.6%	9.1%	2.2%
10	21.5%	11.0%	9.2%	1.3%
20	19.9%	9.8%	9.1%	1.0%

Table 2: WER and percentage of error types for each SNR using TTS-ASR pipeline.

5.6 Conditions for Speech Recognition Error

We trained and evaluated ANTOR in an environment with ASR simulation. Google Cloud Text-to-Speech[4] and Speech-to-Text[5] were used for the TTS and ASR in the construction of the confusion matrix (see Section 4.1). The ESC-50 dataset (Piczak, 2015) was used as the background sound source, and it contains a total of 2,000 different sounds in five categories (e.g., natural soundscapes and urban noises). Randomly selected background noise was assigned to each utterance with a signal-to-noise ratio (SNR) of 0, 5, 10, and 20 dB. The range was selected so that the word error rate (WER) between original and noisy utterance text would be evenly distributed. Table 2 shows the WER of all of the data generated at each SNR and the percentages of substitution (Sub.), insertion (Ins.), and deletion (Del.) errors.

Table 3 shows the evaluation results for each model. Overall, ANTOR showed a higher accuracy and F1 than all three comparison models. These indicate that ANTOR can preferentially generate

[3] https://github.com/pengbaolin/SC-GPT

[4] https://cloud.google.com/text-to-speech

[5] https://cloud.google.com/speech-to-text

Model	SNR											
	0			5			10			20		
	Acc.	F1	WER	Acc.	F1	WER	Acc.	F1	WER	Acc.	F1	WER
SC-LSTM	46.6	53.8	26.9	50.6	57.9	17.8	52.2	59.6	14.8	53.5	60.8	13.1
SC-GPT	47.9	55.4	27.2	52.1	59.9	18.0	54.3	61.9	14.9	55.1	62.8	13.3
GPT-2	48.2	56.0	28.2	52.8	60.6	19.2	54.9	62.7	16.0	56.0	63.7	14.3
ANTOR (ours)	**51.9**	**59.4**	**26.8**	**56.9**	**64.2**	**18.5**	**59.9**	**66.9**	**14.8**	**60.9**	**67.9**	**13.7**

(a) MILU

Model	SNR											
	0			5			10			20		
	Acc.	F1	WER	Acc.	F1	WER	Acc.	F1	WER	Acc.	F1	WER
SC-LSTM	46.6	53.2	27.1	50.3	57.4	18.0	52.4	59.5	14.9	53.4	60.4	13.0
SC-GPT	48.9	56.4	27.2	53.5	60.8	18.2	55.1	62.4	15.0	56.2	63.4	13.2
GPT-2	48.7	56.2	28.3	53.5	61.0	19.3	55.6	62.9	16.1	56.5	63.9	14.1
ANTOR (ours)	**54.2**	**61.5**	**28.0**	**58.8**	**66.0**	**18.7**	**60.8**	**67.9**	**15.9**	**61.9**	**69.0**	**13.7**

(b) BERT

Table 3: Scores for methods evaluated in ASR error simulation environment with background noise at each SNR, using MILU and BERT, respectively. WER indicates how much error was imposed on NLG's output utterances.

Model		CEFR-J level							
		≤A1		≤A2		≤B1		≤B2	
NLG	NLU	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
SC-LSTM	MILU	47.4	53.9	55.7	62.5	62.4	68.8	63.6	69.8
SC-GPT	MILU	47.2	53.9	56.4	63.5	63.4	70.1	66.1	72.5
GPT-2	MILU	48.4	55.0	57.7	64.7	66.0	72.6	68.6	74.8
ANTOR (ours)	MILU	**54.7**	**61.1**	**63.5**	**69.8**	**72.5**	**78.0**	**75.0**	**80.1**
SC-LSTM	BERT	68.6	73.6	72.0	76.3	72.8	76.9	73.1	77.2
SC-GPT	BERT	70.3	75.9	73.3	77.9	77.3	81.4	77.6	81.9
GPT-2	BERT	65.3	70.8	74.9	79.4	79.1	83.5	78.2	82.5
ANTOR (ours)	BERT	**83.0**	**87.0**	**85.7**	**89.2**	**87.8**	**90.6**	**87.7**	**90.6**

Table 4: Scores for each NLG model when evaluated using MILU and BERT. Both MILU and BERT were trained using only vocabulary defined at each CEFR-J level.

Model	% vocab. level in generation			
	≤A1	≤A2	≤B1	≤B2
GPT-2	64.8	77.2	87.2	90.3
ANTOR w/ MILU	**68.1**	**79.8**	**88.0**	**91.0**
ANTOR w/ BERT	67.0	79.6	**88.0**	**91.0**

Table 5: Percentage of vocabulary levels to which words generated by GPT-2 and ANTOR belong. "w/ MILU" and "w/ BERT" are ANTOR models trained with MILU and BERT, respectively. Note that ANTOR is fine-tuned using NLU trained on data at the CEFR-J level indicated by each column.

words that are less likely to be confused. The above result shows that fine-tuning via RL enabled NLG to generate utterances adapted to the noisy environment, regardless of the noise intensity.

5.7 Conditions for Different Vocabulary Levels

We experimented with multiple NLUs trained on data that were gradually filtered along vocabulary levels. For word lists organized by vocabulary level, we used the Common European Framework of Reference (CEFR)'s English Vocabulary Profile[6]. The CEFR defines six levels of language acquisition, from A1 (beginner) to C2 (proficient, comparable to native speakers), with a word list for each level. In our experiment, we used the CEFR-J (Tono and Negishi, 2012) word list for Japanese-English learners[7]. We created four types of training data for NLU by filtering MultiWOZ data with a

[6] https://www.englishprofile.org/

[7] http://www.cefr-j.org/download_eng.html

Intent-slot pair	Num.	% TP change
(Request-Taxi, Depart)	142	14.5 → 90.4
(OfferBook-Train, People)	6	0.0 → 66.7
(Recommend-Hotel, Postcode)	12	14.3 → 80.0
(Recommend-Restaurant, Price)	49	20.8 → 84.0
(NoOffer-Hotel, none)	32	23.5 → 86.7

Table 6: Top five intent-slot pairs that were correctly recognized by BERT at a higher percentage (% TP) by ANTOR compared with GPT-2. "Num." indicates the number of times each intent-slot pair appeared during test.

focus on the four levels ≤A1, ≤A2, ≤B1, and ≤B2 (see Section 4.2). Note that C1 and C2 were not available in CEFR-J and were not used in our experiment. As a result of the filtering, the number of utterances in the datasets at the ≤A1, ≤A2, ≤B1, and ≤B2 levels was 11,190, 15,538, 24,311, and 28,999 utterances, respectively.

ANTOR was trained and evaluated using MILU and BERT trained on each of the four vocabulary levels. Table 4 shows the results. ANTOR outperformed all three comparison models at all vocabulary levels, both with MILU and with BERT. In particular, when using BERT at level ≤A1, the original GPT-2's accuracy and F1 were lower than SC-LSTM and SC-GPT. However, ANTOR had an accuracy and F1 that were significantly improved over the GPT-2 scores by 17.7% and 16.2%, respectively, and it outperformed the scores of SC-LSTM and SC-GPT.

We checked whether the vocabulary in the utterances generated by ANTOR actually changed due to RL. Table 5 shows the percentage of the vocabulary in the utterances that ANTOR had generated during the evaluation when trained with each NLU of each level. Note that stop words and proper nouns were excluded from the calculation. We see that both ANTOR w/ MILU and w/ BERT generated words at each level with a higher frequency than the original GPT-2. From these results, it is considered that the NLG was able to learn utterance generation tailored to the NLU's ability to understand.

6 Case Study

To see how ANTOR improved the performance of NLU, we analyzed the behavior of ANTOR. In this analysis, we used BERT because we thought that a difference from GPT-2 could be clearly seen since the F1 was improved more by using BERT than MILU as in Table 1. The case studies here are done

Ref. DA	[Request-Taxi-Depart-?, Request-Taxi-Leave-?]
GPT-2	I can help you with that. What time would you like to leave and where are you departing from?
BERT	[Request-Train-Depart-?, Request-Train-Leave-?]
ANTOR	Where would you like to **be picked up** and at what time?
BERT	[Request-Taxi-Depart-?, Request-Taxi-Leave-?]

(a) Case 1

Ref. DA	[Recommend-Restaurant-Name-charlie chan, Recommend-Restaurant-Price-cheap, Recommend-Restaurant-Postcode-cb21db]
GPT-2	charlie chan is a cheap restaurant and the postcode is cb21db.
BERT	[Inform-Restaurant-Name-charlie chan, Inform-Restaurant-Price-cheap, Inform-Restaurant-Postcode-cb21db]
ANTOR	I **recommend** charlie chan, it is in the cheap price range and the postcode is cb21db.
BERT	[Recommend-Restaurant-Name-Charlie chan, Recommend-Restaurant-Price-cheap, Recommend-Restaurant-Postcode-cb21db]

(b) Case 2

Table 7: Examples of utterances generated by GPT2 and ANTOR from DAs and DAs predicted by BERT for each utterance. Letters in red indicate DAs misrecognized by BERT. Letters in blue indicate words that may have influenced BERT's prediction.

in clean environments, but similar behaviors were observed for noisy environments and different user vocabulary levels.

First, we listed the intents and slots in the DAs for which the NLU prediction accuracy was considerably improved by the utterances of ANTOR compared with those of GPT-2 (Table 6). Next, we examined the utterances that each model generated from the listed DAs. Table 7 shows examples of utterances generated for (Request-Taxi, Depart) and (Recommend-Restaurant, Price), which have a particularly high occurrence as in Table 6. In case 1, BERT misidentified the train domain instead of the taxi domain from the GPT-2 utterances. In contrast, ANTOR correctly conveyed the DAs by explicitly using the phrase "be picked up." In case 2, the intention of "inform" was conveyed by GPT-2 instead of "recommend." However, ANTOR explicitly used the word "recommend" to correctly convey the DA.

These results suggest that fine-tuning NLG using RL enables NLG to generate utterances adapted to the NLU.

Note that since humans will not have the problems that the NLU had here because humans have a better understanding, we expect ANTOR to adapt differently when interacting with humans.

7 Summary and Future Work

This paper investigated whether NLG can generate utterances adapted to the dialogue environment and the user via RL. We proposed a method, ANTOR, and conducted experiments using MultiWOZ to confirm that ANTOR can generate such utterances for multiple NLUs with different model architectures. In addition, we also consistently confirmed the effectiveness of ANTOR for noisy environments and a user's vocabulary levels.

For future work, we plan to evaluate whether ANTOR optimized for NLU is also effective for humans. We are also interested in extending our method for practical use (e.g., real-time adaptation to users in an online dialogue environment). Furthermore, we would like to utilize methods to optimize an entire system with RL, such as (Mehri et al., 2019) and (Ohashi and Higashinaka, 2022), so that all modules of a system can be adapted to users.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number 19H05692. We used the computational resources of the supercomputer "Flow" at the Information Technology Center, Nagoya University.

References

Gabor Angeli, Percy Liang, and Dan Klein. 2010. A simple domain-independent probabilistic approach to generation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 502–512.

Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2016. An actor-critic algorithm for sequence prediction. *arXiv preprint arXiv:1607.07086*.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 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.

Wenhu Chen, Jianshu Chen, Pengda Qin, Xifeng Yan, and William Yang Wang. 2019. Semantically Conditioned Dialog Response Generation via Hierarchical Disentangled Self-Attention. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3696–3709.

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*, pages 4171–4186.

Yue Dong, Yikang Shen, Eric Crawford, Herke van Hoof, and Jackie Chi Kit Cheung. 2018. BanditSum: Extractive Summarization as a Contextual Bandit. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3739–3748.

Ondřej Dušek and Filip Jurčíček. 2016. A Context-aware Natural Language Generator for Dialogue Systems. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 185–190.

Maryam Fazel-Zarandi, Longshaokan Wang, Aditya Tiwari, and Spyros Matsoukas. 2019. Investigation of error simulation techniques for learning dialog policies for conversational error recovery. *arXiv preprint arXiv:1911.03378*.

Jianfeng Gao, Michel Galley, Lihong Li, et al. 2019. Neural approaches to conversational ai. *Foundations and trends® in information retrieval*, pages 127–298.

Dilek Hakkani-Tür, Gokhan Tur, Asli Celikyilmaz, Yun-Nung Vivian Chen, Jianfeng Gao, Li Deng, and Ye-Yi Wang. 2016. Multi-Domain Joint Semantic Frame Parsing using Bi-directional RNN-LSTM. In *Proceedings of The 17th Annual Meeting of the International Speech Communication Association*.

Braden Hancock, Antoine Bordes, Pierre-Emmanuel Mazare, and Jason Weston. 2019. Learning from Dialogue after Deployment: Feed Yourself, Chatbot! In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3667–3684.

Srinivasan Janarthanam and Oliver Lemon. 2010. Learning to Adapt to Unknown Users: Referring Expression Generation in Spoken Dialogue Systems. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 69–78.

Natasha Jaques, Asma Ghandeharioun, Judy Hanwen Shen, Craig Ferguson, Agata Lapedriza, Noah Jones, Shixiang Gu, and Rosalind Picard. 2019. Way off-policy batch deep reinforcement learning of implicit human preferences in dialog. *arXiv preprint arXiv:1907.00456*.

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

Ravikumar Kondadadi, Blake Howald, and Frank Schilder. 2013. A statistical nlg framework for aggregated planning and realization. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 1406–1415.

Julia Kreutzer, Shahram Khadivi, Evgeny Matusov, and Stefan Riezler. 2018. Can Neural Machine Translation be Improved with User Feedback? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 92–105.

Jiexi Liu, Ryuichi Takanobu, Jiaxin Wen, Dazhen Wan, Hongguang Li, Weiran Nie, Cheng Li, Wei Peng, and Minlie Huang. 2021. Robustness Testing of Language Understanding in Task-Oriented Dialog. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 2467–2480.

Jelena Luketina, Nantas Nardelli, Gregory Farquhar, Jakob Foerster, Jacob Andreas, Edward Grefenstette, Shimon Whiteson, and Tim Rocktäschel. 2019. A Survey of Reinforcement Learning Informed by Natural Language. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pages 6309–6317.

François Mairesse and Marilyn A Walker. 2010. Towards personality-based user adaptation: psychologically informed stylistic language generation. *User Modeling and User-Adapted Interaction*, pages 227–278.

François Mairesse and Steve Young. 2014. Stochastic Language Generation in Dialogue using Factored Language Models. *Computational Linguistics*, pages 763–799.

Michael F. McTear. 2002. Spoken Dialogue Technology: Enabling the Conversational User Interface. *ACM Computing Surveys*, page 90–169.

Shikib Mehri, Tejas Srinivasan, and Maxine Eskenazi. 2019. Structured Fusion Networks for Dialog. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 165–177.

Alice H Oh and Alexander I Rudnicky. 2002. Stochastic natural language generation for spoken dialog systems. *Computer Speech & Language*, pages 387–407.

Atsumoto Ohashi and Ryuichiro Higashinaka. 2022. Post-processing Networks: Method for Optimizing Pipeline Task-oriented Dialogue Systems using Reinforcement Learning. In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 1–13.

Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le. 2019. SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition. In *Proc. Interspeech 2019*, pages 2613–2617.

Baolin Peng, Chenguang Zhu, Chunyuan Li, Xiujun Li, Jinchao Li, Michael Zeng, and Jianfeng Gao. 2020. Few-shot natural language generation for task-oriented dialog. *arXiv preprint arXiv:2002.12328*.

Karol J. Piczak. 2015. ESC: Dataset for Environmental Sound Classification. In *Proceedings of the 23rd Annual ACM Conference on Multimedia*, pages 1015–1018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, page 9.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*.

Jost Schatzmann, Blaise Thomson, and Steve Young. 2007. Error simulation for training statistical dialogue systems. In *Proceedings of 2007 IEEE Workshop on Automatic Speech Recognition & Understanding*, pages 526–531.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. 2015. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Amanda Stent, Rashmi Prasad, and Marilyn Walker. 2004. Trainable Sentence Planning for Complex Information Presentations in Spoken Dialog Systems. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 79–86.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems*, pages 3008–3021.

Shang-Yu Su, Kai-Ling Lo, Yi-Ting Yeh, and Yun-Nung Chen. 2018. Natural Language Generation by Hierarchical Decoding with Linguistic Patterns. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 61–66.

Pradyumna Tambwekar, Murtaza Dhuliawala, Lara J. Martin, Animesh Mehta, Brent Harrison, and Mark O. Riedl. 2019. Controllable Neural Story Plot Generation via Reward Shaping. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pages 5982–5988.

Yukio Tono and Masashi Negishi. 2012. The CEFR-J: Adapting the CEFR for English language teaching in Japan. *Framework & Language Portfolio SIG Newsletter*, pages 5–12.

Van-Khanh Tran and Le-Minh Nguyen. 2017. Natural Language Generation for Spoken Dialogue System using RNN Encoder-Decoder Networks. In *Proceedings of the 21st Conference on Computational Natural Language Learning*, pages 442–451.

M.A. Walker, S.J. Whittaker, A. Stent, P. Maloor, J. Moore, M. Johnston, and G. Vasireddy. 2004. Generation and evaluation of user tailored responses in multimodal dialogue. *Cognitive Science*, pages 811–840.

Marilyn A. Walker, Owen C. Rambow, and Monica Rogati. 2002. Training a sentence planner for spoken dialogue using boosting. *Computer Speech & Language*, pages 409–433.

Longshaokan Wang, Maryam Fazel-Zarandi, Aditya Tiwari, Spyros Matsoukas, and Lazaros Polymenakos. 2020. Data Augmentation for Training Dialog Models Robust to Speech Recognition Errors. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 63–70.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, and Steve Young. 2016. Multi-domain Neural Network Language Generation for Spoken Dialogue Systems. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 120–129.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically Conditioned LSTM-based Natural Language Generation for Spoken Dialogue Systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1711–1721.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Tiancheng Zhao, Kaige Xie, and Maxine Eskenazi. 2019. Rethinking Action Spaces for Reinforcement Learning in End-to-end Dialog Agents with Latent Variable Models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1208–1218.

Qi Zhu, Zheng Zhang, Yan Fang, Xiang Li, Ryuichi Takanobu, Jinchao Li, Baolin Peng, Jianfeng Gao, Xiaoyan Zhu, and Minlie Huang. 2020. ConvLab-2: An Open-Source Toolkit for Building, Evaluating, and Diagnosing Dialogue Systems. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 142–149.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.