

# Neural Conversation Model Controllable by Given Dialogue Act Based on Adversarial Learning and Label-aware Objective

Seiya Kawano, Koichiro Yoshino and Satoshi Nakamura

Devision of Information Science, Graduate School of Science and Technology  
Nara Institute of Science and Technology, Nara 630-0192, Japan  
{kawano.seiya.kj0, koichiro, s-nakamura}@is.naist.jp

## Abstract

Building a controllable neural conversation model (NCM) is an important task. In this paper, we focus on controlling the responses of NCMs by using dialogue act labels of responses as conditions. We introduce an adversarial learning framework for the task of generating conditional responses with a new objective to a discriminator, which explicitly distinguishes sentences by using labels. This change strongly encourages the generation of label-conditioned sentences. We compared the proposed method with some existing methods for generating conditional responses. The experimental results show that our proposed method has higher controllability for dialogue acts even though it has higher or comparable naturalness to existing methods.

## 1 Introduction

A dialogue act is defined as the intention or the function of an utterance in dialogues. Dialogue act labels are defined as unique classes to distinguish between kinds of dialogue acts (Boyer et al., 2010; Bunt et al., 2012). Some existing studies have exploited the dialogue act as a component in modeling the dialogue strategy of dialogue systems (Meguro et al., 2010; Yoshino and Kawahara, 2015; Shibata et al., 2016; Keizer and Rieser, 2017).

Neural conversation models (NCMs), which learn a direct mapping between a dialogue history and a response utterance, are widely researched as a scalable approach to building non-task oriented dialogue systems (Vinyals and Le, 2015; Serban et al., 2016). However, it is difficult to control their responses on the basis of actual constraints such as dialogue act classes. Some existing studies have tackled this problem to control responses from

NCMs by using actual labels; however, these models still had some limitations (Wen et al., 2015; Li et al., 2016; Sun et al., 2017; Zhao et al., 2017; Huang et al., 2018; Zhou et al., 2018). One crucial issue was that they do not have any explicit training objectives to guarantee that a generation has a discriminability for a given condition.

We extend a framework of the generative adversarial network for sequential generation (Yu et al., 2017) for improving the controllability of NCMs under the constraint of a given dialogue act condition. We propose an adversarial learning framework that alternatively trains between conditioned generator and a conditioned discriminator. The discriminator has a multi-class objective that explicitly classifies a generated response into an appropriate dialogue act class. This improves the discriminability of generation.

In this paper, we first describe the task of conditional response generation given a dialogue act label and its existing approaches (Section 3). Second, we introduce an adversarial learning framework and extend its architecture and objective to fit the problem of conditional generation (Section 4). In experiments, we use metrics to evaluate the controllability and naturalness of responses (Section 5). The experimental results show that our proposed model achieved the best controllability score in both automatic and human subjective evaluations even if it achieves better or comparable naturalness to existing methods (Section 6).

## 2 Related Work

Dialogue systems that have dialogue management modules determine a dialogue act or dialogue state

of a system response by using statistical methods such as reinforcement learning (Young et al., 2010; Meguro et al., 2010; Yoshino and Kawahara, 2015; Keizer and Rieser, 2017). Response generation modules generate responses according to these dialogue acts or dialogue states on the basis of the rules, templates, agendas or other statistical models (Oh and Rudnicky, 2000; Xu and Rudnicky, 2000). Recently, neural network based generation modules have been widely used.

Wen et al. (2015) proposed a conditional language model (Semantically Conditioned Long Short-Term Memory; SC-LSTM) for task-oriented systems, which generates utterances on the basis of any dialogue acts and frames in the domain of restaurant navigation dialogue by using gating mechanism. However, the training framework of SC-LSTM requires state frames that express the function and the contents of target utterances entirely. Thus, it is not realistic to apply this method to building an open-domain dialogue system. Zhao et al. (2017) proposed an NCM based on a variation of the conditional variational autoencoder (CVAE), which generates responses that have high diversity in discourse level by using latent variables as dialogue acts. However, this model has no mechanism to guarantee for generating discriminable responses for given dialogue acts.

There is another research trend in controlling NCMs with a given condition, such as speaker or emotion labels (Li et al., 2016; Sun et al., 2017; Huang et al., 2018; Zhou et al., 2018). These NCMs are optimized by softmax cross-entropy loss (SCE-loss), which calculates a loss word-by-word. However, such existing training objectives do not necessarily guarantee that a generated response has high discriminability to for a given class label. In other words, SCE-loss is not an appropriate objective that explicitly evaluates whether a generated response reflects the property of the given class label or not. Therefore, the generated response will be biased by majority class labels.

To prevent these problems, we introduce the

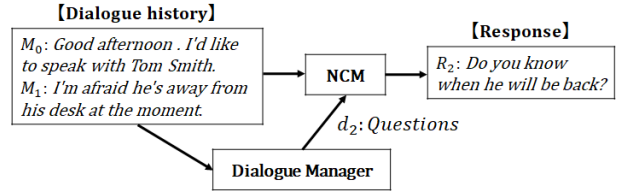


Figure 1: Task of response generation conditioned by dialogue act labels.

framework of the generative adversarial network (Yu et al., 2017; Li et al., 2017a; Tuan and Lee, 2019). This framework makes it possible to consider the total quality of generated sequences unlike SCE-loss, which is optimized for each token. We extend adversarial networks to generate qualified and controlled sentences given a condition, especially dialogue act labels.

### 3 Response Generation Conditioned by Dialogue Act Label

#### 3.1 Task Settings

The task we focus on is building a controllable NCM with a given condition, typically dialogue act labels. The problem is defined as generating the  $i$ th response word sequence  $R_i = \{w_1, w_2, \dots, w_T\}$  given a dialogue history  $M = \{M_{i-1}, M_{i-2}, \dots, M_{i-n}\}$  and dialogue act label  $d_i$ . Here,  $n$  is the length of a dialogue, and  $T$  is the number of words in an utterance. As shown in Figure 1, a response  $R_i$  is required to satisfy not only the behavioral characteristic of a given dialogue act but also the appropriateness in the dialogue context (=history).

One of the simplest approaches to building such a conditional generation system given a class label in NCM is adding the class label to the input of a decoder (Li et al., 2016; Zhao et al., 2017; Sun et al., 2017). We describe this baseline in the following section.

#### 3.2 Conditional NCM with Dialogue Acts

We introduce a general conditional NCM that is conditioned by dialogue act labels as a baseline. We built an NCM on the basis of a hierarchical encoder-decoder model that explicitly gives labels to the decoder at any of the steps of decoding as

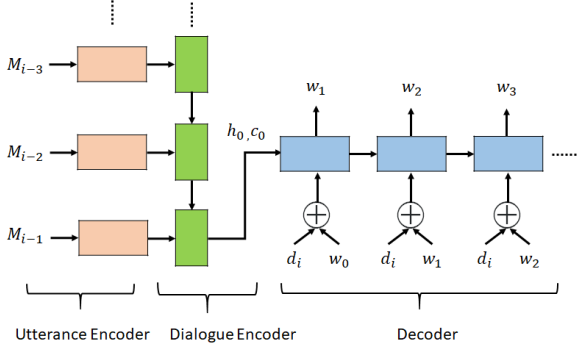


Figure 2: Conditional-NCM with dialogue acts.

shown in Figure 2.

Recurrent neural networks (RNNs) such as long short-term memory (LSTM) are generally used to model a sequential generation of responses in NCMs (Hochreiter and Schmidhuber, 1997; Vinyals and Le, 2015; Serban et al., 2016). The encoder receives a word at each time step by using forward RNNs to encode an utterance into a fixed length vector (utterance encoder). Utterance vectors of a dialogue are input to another fixed length vector according to their time sequence to encode the dialogue context (dialogue encoder). The resultant vector is fed into the decoder to generate a response sentence (word sequence). We used the same encoder architecture as Tian et al. (2017).

In the decoding steps of the NCMs, the decoder receives a previous hidden vector  $h_{t-1}$ , memory cell  $c_{t-1}$ , and generated word  $w_{t-1}$  to generate a word  $w_t$ . Here,  $t$  is an actual time-step of generation. The model has been changed to receive not only the previous word  $w_{t-1}$  but also the dialogue act label  $d$  in conditional generation. The vector representations of  $d$  and  $w_t$  are concatenated and used as the input of the decoder at time-step  $t$ .<sup>1</sup> The decoder itself also predicts the vector representation of words. This architecture is also the same as those of the models proposed by Zhao et al. (2017).

Softmax cross-entropy loss (SCE-loss) is widely used to train the model.

$$loss_{sce} = -\log \frac{\exp(x_c)}{\sum_k^{|\mathcal{V}|} \exp(x_k)}. \quad (1)$$

<sup>1</sup> $w_t$  and  $d$  are converted into vector representation and then concatenated.

Here,  $|\mathcal{V}|$  indicates the vocabulary size,  $x \in \mathbb{R}^{|\mathcal{V}|}$  indicates the output of the projection layer in the decoding steps, and  $x_k \in \mathbb{R}^{|\mathcal{V}|}$  indicates the  $k$ th element of  $x$ .  $x_c$  is the target word. SCE-loss optimizes the prediction of words at each decoding step. However, it does not use the information of a given dialogue act label in the loss calculation during training. Thus, the resultant model often generates a response that does not consider a given dialogue act label or a biased response by using the majority dialogue act labels in the training data. We tackle this problem by introducing an explicit training objective to generate a conditioned word sequence in adversarial learning.

## 4 Conditional Response Generation Based on Adversarial Learning

We introduce sequential generative adversarial networks (SeqGANs) (Yu et al., 2017; Li et al., 2017a; Tuan and Lee, 2019) to improve the controllability and quality of conditional response generation. SeqGAN is a prospective approach to preventing the problems caused by SCE-loss based training because it can evaluate not only the word prediction of each decoding step but also the whole quality of a generated sequence. In this section, we first describe the architecture of SeqGAN (Section 4.1) and then propose our extension of SeqGAN to realize conditional response generation by using given dialogue act labels (Section 4.2).

### 4.1 SeqGAN for Response Generation

The generation process in SeqGAN is formalized as a Markov decision process (MDP) and optimized with reinforcement learning (RL) (Li et al., 2017a; Tuan and Lee, 2019). The problem of response generation in NCMs is generating response word sequence  $R = \{w_1, \dots, w_T\}$  given a dialogue context  $M$ . Such a word selection process in the generation is defined as an action sequence, which is generated by an actual policy in MDP. In SeqGAN, the generator generates a sentence according to the current policy. The discriminator gives an evaluation score to the generated sentence after the generation. The evaluation score is fed as

a reward to update the policy of the generator in RL. We use a policy gradient (Williams, 1992) to train the policy. The objective function and its gradient of the policy gradient are defined as follows<sup>2</sup>.

$$J(\theta) = \sum_{w_{1:T}} G_\theta(w_t|w_{1:t-1}, M) \cdot Q_{D_\phi}^{G_\theta}((w_{1:t-1}, M), w_t) \quad (2)$$

$$\nabla J(\theta) \simeq \frac{1}{T} \sum_{t=1}^T \sum_{w_t \in \mathcal{V}} Q_{D_\phi}^{G_\theta}((w_{1:t-1}, M), w_t) \cdot \nabla_\theta G_\theta(w_t|w_{1:t-1}, M) \quad (3)$$

$$= \frac{1}{T} \sum_{t=1}^T \mathbb{E}_{w_t \sim G_\theta} [Q_{D_\phi}^{G_\theta}((w_{1:t-1}, M), w_t) \cdot \nabla_\theta \log p(w_t|w_{1:t-1}, M)] \quad (4)$$

Here,  $\theta$  is a parameter of the policy.  $w_{1:t-1}$  indicates a word sequence,  $\mathcal{V}$  is a vocabulary, and  $p$  is the generative probability of word  $w_t \in \mathcal{V}$ .  $Q_{D_\phi}^{G_\theta}((w_{1:t-1}, M), w_t)$  is an action-value function that gives an expected future reward when the system takes the action of generating word  $w_t$  given the state: already generated word sequence  $w_{1:t-1}$  and dialogue context  $M$ .  $\phi$  is a parameter of the discriminator. The discriminator only outputs the reward after the whole generation of the sequence. Thus, the value of the action-value function  $Q_{D_\phi}^{G_\theta}((w_{1:t-1}, M), w_t)$  for each step is calculated by using a Monte Carlo tree search (MCTS) under the current policy and its parameter  $\theta$  (Browne et al., 2012).

The discriminator is trained to classify a generated sentence (fake) and sentence in training data (real). Its training objective is defined as,

$$\min_{\phi} -\mathbb{E}_{R \sim p_{data}(\cdot|M)} [\log D_\phi(R, M)] - \mathbb{E}_{R \sim G_\theta(\cdot|M)} [\log(1 - D_\phi(R, M))]. \quad (5)$$

The generator and the discriminator are trained alternatively to train their network adversarially.

## 4.2 SeqGAN for Conditional Response Generation with Dialogue Acts

The generator and the discriminator in SeqGAN are extended to produce responses according to given dialogue acts. The adversarial framework is extended for jointly optimizing both networks: a generator network to produce response utterances

<sup>2</sup>Detailed derivation is shown in (Yu et al., 2017).

under specified dialogue acts, and a discriminator network to distinguish between generation (fake) and training data (real) that reflect their conditions.

As the generator network, we applied the conditional-NCM described in Section 3.2. Equations (2) - (4) are changed as follows.

$$J(\theta) = \sum_{w_{1:T}} G_\theta(w_t|w_{1:t-1}, M, d) \cdot Q_{D_\phi}^{G_\theta}((w_{1:t-1}, M, d), w_t) \quad (6)$$

$$\nabla J(\theta) \simeq \frac{1}{T} \sum_{t=1}^T \sum_{w_t \in \mathcal{V}} Q_{D_\phi}^{G_\theta}((w_{1:t-1}, M, d), w_t) \cdot \nabla_\theta G_\theta(w_t|w_{1:t-1}, M, d) \quad (7)$$

$$= \frac{1}{T} \sum_{t=1}^T \mathbb{E}_{w_t \sim G_\theta} [Q_{D_\phi}^{G_\theta}((w_{1:t-1}, M, d), w_t) \cdot \nabla_\theta \log p(w_t|w_{1:t-1}, M, d)] \quad (8)$$

As the discriminator network, we incorporated dialogue act labels in the classification model (Figure 3). In the discriminator model, the utterance encoder converts dialogue contexts into fixed length vectors and uses them as features of discrimination. We propose to use two discriminators for incorporating dialogue act label information in the discriminator: implicit and explicit. Each method is described below in respective sections.

### 4.2.1 Binary Objective; Implicit-Discriminator

We built a simple extension for the discriminator that incorporates dialogue acts in the feature vectors of the discriminator. We call this architecture ‘‘implicit.’’ This discriminator is defined as,

$$\min_{\phi} -\mathbb{E}_{R \sim p_{data}(\cdot|M, d)} [\log D_\phi(R, M, d)] - \mathbb{E}_{R \sim G_\theta(\cdot|M, d)} [\log(1 - D_\phi(R, M, d))]. \quad (9)$$

We expect that the implicit discriminator can use the information of dialogue acts as a feature and discriminate generated results as fakes if they do not follow a given dialogue act (Figure 3, lower-right). There are some works that have similar approaches in emotional response generation (Sun et al., 2018; Kong et al., 2019). However, this discriminator is still a simple extension of the standard discriminator, which classifies response in two classes. In other words, the objective is not changed; thus it probably has difficulty in distinguishing the class (dialogue act label) of

responses. We propose another discriminator to solve this problem in the next section.

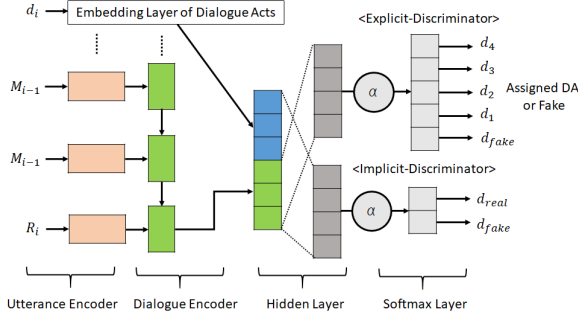


Figure 3: Implicit & Explicit-Discriminator.

#### 4.2.2 Multi-class Objective; Explicit-Discriminator

We propose an approach extending the classification problem of the discriminator from the binary classification of fake/real to multi-class classification to distinguish target dialogue act classes (Figure 3 upper-right). This discriminator has a multi-class objective for  $N + 1$  class classification. Here,  $N$  is the number of unique dialogue act classes; another one is a fake class for categorizing the responses as generated. We call this architecture “explicit.” Its objective function is defined as,

$$\min_{\phi} - \sum_{i=1}^N \mathbb{E}_{R \sim P_{data}(\cdot|M,d)} [\log D_{\phi}(d_i|R, M)] - \mathbb{E}_{R \sim G_{\theta}(\cdot|M,d)} [\log D_{\phi}(d_{fake}|R, M)]. \quad (10)$$

We used the posterior probability  $D_{\phi}(d|R, M)$  estimated by the discriminator as the reward of the generator. We expect that this discriminator will encourage training the generator to generate discriminative sentences with dialogue acts because generations that follow different dialogue act manners are penalized even if they are natural. Odena (2016) proposed a similar idea to use multi-class objective in GAN for the image generation task.

#### 4.3 Speeding Up Adversarial Learning Using Simple Recurrent Unit

Using LSTM or GRU as an encoder and decoder is a general method for building NCMs (Vinyals and Le, 2015; Serban et al., 2016). LSTM is also often used for classification problems to encode a hierarchical structure, such as the discriminators of

GANs (Tran et al., 2017). However, the training speed of LSTM is much slower than other types of networks, although LSTM has dominant performance (Lei et al., 2017). This characteristic is critical for adversarial learning, which requires a large number of iterations.

We used the policy gradient in this research to update the parameters of the generator, which is based on expected reward calculation by MCTS. However, MCTS requires enormous calculation costs because it requires scanning the discriminator and generator  $r \times w$  times per one update of generator, where  $r$  is the number of rollouts and  $w$  is the number of words in a response each time step. Thus, we propose to use a simple recurrent unit (SRU) (Lei et al., 2018) in our generator and discriminator. SRU is known as an extension of RNN, which has comparable performance to LSTM even if it works at significantly higher speed. SRU is defined as follows.

$$\tilde{v}_t = \mathbf{W}v_t \quad (11)$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_f v_t + \mathbf{b}_f) \quad (12)$$

$$\mathbf{r}_t = \sigma(\mathbf{W}_r v_t + \mathbf{b}_r) \quad (13)$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + (1 - \mathbf{f}_t) \odot \tilde{v}_t \quad (14)$$

$$\mathbf{h}_t = \mathbf{r}_t \odot \mathbf{g}(\mathbf{c}_t) + (1 - \mathbf{r}_t) \odot \mathbf{v}_t \quad (15)$$

Here,  $v_t$  is the input vector at time-step  $t$ ,  $f_t$  is the forgetting gate,  $r_t$  is the input gate,  $c_t$  is the memory cell, and  $h_t$  is the hidden vector. The key idea of SRU is minimizing the number of vectors and gates affected by previous states. Under this definition, only  $c_t$  is affected by the previous state  $c_{t-1}$ . Furthermore,  $c_t$  and  $h_t$  are calculated only by the element-wise production and summation of a vector for easy speed up. It was reported that forward and backward propagations in SRU are 10-16 times faster than LSTM (Lei et al., 2018). SRU leads to computational advantage compared with another type of RNNs including GRU as well. We expect to have a significant improvement in speeding up adversarial learning by using SRU instead of LSTM. However, applying SRU to NCM has no track record; thus, we introduce SRU to both the existing methods and our proposed adversarial network; however, we also perform a comparison with another baseline implemented by LSTM.



## 5 Experimental Settings

### 5.1 Dataset

We used the DailyDialog corpus that covers ten categories from a wide variety of topics (Li et al., 2017b). The corpus contains 13,118 dialogues, with a total of 102,979 utterances annotated with dialogue act labels: inform (46,532 utterances), questions (29,428 utterances), directives (17,295 utterances) and commissive (9,724 utterances). We divided the corpus into training/validation/test sets with 11,118/1,000/1,000 dialogues according to the work of Li et al. (2017b). In all experiments, the vocabulary size was set to 25,000, and all the OOV words were replaced by “UNK” symbol.

### 5.2 Training Settings

We used the same setting for embedding: words, 256, dialogue acts, 100. The mini-batch size was 32. In the training for conditional-NCM, we set two-layers RNNs in both the encoder and the decoder and used the Adam optimizer with a learning rate of 1e-5.

In the proposed adversarial learning, we followed the training procedure proposed by Li et al. (2017a). The training algorithm that we used is shown as follows.

---

**Algorithm 1** Training procedure

---

```
1: for number of iterations do
2:    $G' \leftarrow G$ 
3:   for number of G-steps do
4:     sample  $(M_G, R_G, d_G)$  from training data
5:     generate response  $\hat{R}_G$  by using  $G'$  on  $(M_G, d_G)$ 
6:     compute reward  $r_{R_G}$  for  $(M_G, \hat{R}_G, d_G)$  by using  $D$ 
7:     update  $G$  on  $(M_G, \hat{R}_G, d_G)$  using  $r_{R_G}$ 
8:   for number of D-steps do
9:     sample  $(M_D, R_D, d_D)$  from training data
10:    generate response  $\hat{R}_D$  by using  $G$  on  $(M_D, d_D)$ 
11:    update  $D$  using  $(M_D, \hat{R}_D, d_D)$  and  $(M_D, R_D, d_D)$ 
```

---

In the training of SeqGAN for conditional response generation based on dialogue acts, we prepare well pre-trained conditional generator and a discriminator in advance. After initializing parameters by pre-trained models, G-steps for the generator  $G$  and D-steps for the discriminator  $D$  are

applied alternatively to train them. In G-steps, a generated response  $\hat{R}_G$  is sampled by using a dialogue history  $M_G$  and dialogue act  $d_G$ , and then the reward  $r_{\hat{R}_G}$  for the generation is calculated by the discriminator  $D$ . By using the calculated reward  $r_{\hat{R}_G}$ , parameters of  $G$  are updated. In D-steps, a real response  $R_D$ , given a dialogue history  $M_D$  and a dialogue act label  $d_D$ , is sampled from the training data. A fake response  $\hat{R}_D$  is generated from the generator  $G$  by using the dialogue history  $M_D$  and the dialogue act label  $d_D$ . Parameters of the discriminator  $D$  is updated by using the real sample and the fake sample.

We set the number of G-steps to 4 and D-steps to 20. In the generator, we used two-layers SRUs in both the encoder and the decoder as 1024 hidden units. We used the Adam optimizer with a learning rate of 1e-5. For the discriminator, we used a one-layer SRU, 1024 hidden units, and the SGD optimizer with a learning rate of 1e-3. We set the number of rollout to 5 in MCTS.

### 5.3 Automatic Evaluation Metrics

We automatically evaluated generation results by comparing with references in the test-set. As the automatic evaluation, we used three different types of metrics: perplexity, relevance scores, and controllability.

Perplexity is a metric for evaluating a language model performance. Likelihoods of models for reference responses are calculated as perplexities. Note that the perplexity score does not directly reflect the quality of generation; dull responses also have good perplexity scores.

Relevance scores are similarities between references and generated results. We used NIST, a variation of BLEU, which focuses on content words more than BLEU (Doddington, 2002). However, using count-based metrics such as BLEU and NIST are not appropriate, because they have small correlations with human judgment score in response generation tasks (Liu et al., 2016). Thus, we also used three different relevance scores proposed by Liu et al. (2016): embedding average (“Average”), greedy matching (“Greedy”) and

vector extrema (“Extrema”)<sup>3</sup>. “Average” calculates a cosine similarity between the reference sentence vector and the generated sentence vector. Each sentence vector is calculated by an average of word embedding vectors in the sentence. “Extrema” also calculates a cosine similarity between sentence vectors; however, the sentence vector is constructed in a different way. Each dimension of the sentence vector is selected from the same dimension of a word embedding vector, which has the highest absolute value in the sentence. “Greedy” calculates cosine similarities of word pairs in the reference and the generated sentence, which is paired by alignment, and then averages these similarities.

The last automatic metric we used is “controllability”, which is given by the classification result of the pre-trained dialogue act classifier by using the training set. We connected our encoder for conditional-NCM (Figure 2 left side) to a multi-class softmax layer to build the classifier.<sup>4</sup> Any generated sentences are labeled by the classifier and then compared with the given condition label to calculate the label accuracy.

#### 5.4 Human Subjective Evaluation Metrics

The automatic evaluation scores still have a problem in that they do not have high correlations with human subjective evaluation results (Liu et al., 2016). Thus, we also evaluated systems with a human subjective evaluation to confirm the naturalness and controllability of responses.

In the evaluation of naturalness, we used a 3-point scale score in accordance with the existing work (Li et al., 2019). Thirty generated responses were randomly selected from each dialog act (120 in total) and human annotators selected an evaluation for the sample in following instructions.

- 2: The response can be used as a reply and it is informative and interesting; the response is natural and can make the conversation continue.

<sup>3</sup>We used fastText embeddings trained by wikipedia-dump data. The size of vectors was 300.

<sup>4</sup>The accuracy of the classifier in the test-set was 0.8303.

- 1: The response can be used as a reply, but it is too generic like “ I don’t know. ”
- 0: The response cannot be used as a reply to the given dialogue history. It is either semantically irrelevant or disfluent.

Each sample was evaluated by three annotators, and the final score was decided by majority voting. If the evaluation was completely separated (0, 1 and 2), the example was evaluated as 1.

In the controllability evaluation, we requested one annotator to annotate dialogue acts for generated responses, who had two years experience in dialogue act annotation. The annotator was trained by using training data of DailyDialog corpus before the evaluation.

## 6 Experimental Results

### 6.1 Results of Automatic Evaluation

Table 1 shows the results of the automatic objective evaluation. We compared our proposed SeqGAN based on explicit-discriminator (Adversarial-Explicit) with the following baselines. “Vanilla-NCM” shows performances of vanilla LSTM, which has no mechanism to receive condition labels. These scores indicate a general performance of systems in DailyDialog corpus. “Conditional-NCMs” show performances of NCMs that receives condition labels on its decoder as proposed by Zhao et al. (2017). We compared two variations of “Conditional-NCMs”, LSTM and SRU, to check the performance of SRU compared with LSTM. “Adversarial-Implicit” shows performances of SeqGAN that has implicit discriminator, which is proposed by Sun et al. (2018) and Kong et al. (2019). “Adversarial-Explicit” indicates the proposed model that has a multi-class discriminator on its SeqGAN. The table shows both results of beam search (width=5) and random sampling in the decoding process.

#### 6.1.1 Speeding up Using SRU

The comparison between “Conditional-NCM w/ LSTM” and “Conditional-NCM w/ SRU” indicates that the speeding up using SRU works well; SRU achieves higher relevance scores to LSTM. SRU used 53,539K parameters, whereas LSTM

Decoding	Models	Perplexity	Relevance				Controllability
			NIST	Average	Greedy	Extreme	
Beam search	Vanilla-NCM w/ LSTM	33.9076	0.0386	0.7603	0.5763	0.5079	-
	Conditional-NCM w/ LSTM	33.2739	0.0502	0.7804	0.5980	0.5319	0.8651
	Conditional-NCM w/ SRU	36.6614	0.0610	0.7850	0.5981	0.5380	0.8702
	Adversarial-Implicit	39.2864	0.0596	0.7773	<b>0.6041**</b>	<b>0.5417**</b>	0.8615
	Adversarial-Explicit	39.6993	0.0613	0.7824	<b>0.6034**</b>	<b>0.5401*</b>	<b>0.8946**</b>
Sampling	Vanilla-NCM w/ LSTM	33.9076	0.0265	0.7746	0.5326	0.4662	-
	Conditional-NCM w/ LSTM	33.2739	0.0342	0.7867	0.5540	0.4849	0.8015
	Conditional-NCM w/ SRU	36.6614	0.0365	0.7917	0.5590	0.4905	0.8176
	Adversarial-Implicit	39.2864	<b>0.0401**</b>	0.7853	<b>0.5641**</b>	<b>0.5018**</b>	0.8081
	Adversarial-Explicit	39.6993	<b>0.0411**</b>	0.7907	<b>0.5687**</b>	<b>0.5036**</b>	<b>0.8531**</b>

Table 1: Results in automatic evaluation: note that results of random sampling are an average of five decoding runs, each decoding is initialized with random seed. \*:  $p < 0.05$  and \*\*:  $p < 0.001$  indicate results of significance tests (vs. conditional-NCM w/ SRU).

required 82,924K parameters. These results support our experiments using SRU instead of LSTM. We will mainly focus on comparisons of SRU models in the following sections.

### 6.1.2 Qualities of Generated Responses

The “Vanilla-NCM w/ LSTM” scores show the difficulty of conversation modeling in Daily-Dialog corpus. By comparing these scores with scores of other conditional generation models, “Conditional-NCM” and “Adversarial”, conditional generation models improved relevance scores even if they have controllability. This is probably because the dialogue act condition can be a training constrain to prevent dull responses. Generation methods using adversarial learning improved relevance scores than “Conditional-NCM w/ SRU”.

### 6.1.3 Controllability of each Dialogue Act

By comparing “Controllability” scores, the proposed “Adversarial-Explicit” achieved best scores in both beam-search and sampling decoding. For a detailed analysis, we show a controllability score of “Adversarial-Explicit” for each dialogue act label in Table 2 and Table 3 (beam-search and sampling). Tables show precision, recall and their harmonic mean (F1) for each dialogue act, and the improvement from the score of “Conditional-NCM w/ SRU”, which achieved the best controllability in baselines (**improv.**). The proposed “Adversarial-Explicit” achieved improvements for any classes, but in particular, it achieved large improvements on “Directives” and “Commissive”

Dialogue acts	Prec.	Recall	F1 (improv.)
Inform (46%)	0.8881	0.9291	0.9081 ( <b>+0.0167</b> )
Questions (30%)	0.9693	0.9982	0.9836 ( <b>+0.0060</b> )
Directives (16%)	0.9372	0.7164	0.8121 ( <b>+0.0971</b> )
Commissive (8%)	0.6285	0.6921	0.6588 ( <b>+0.0240</b> )
Weighted avg	0.8980	0.8946	0.8935 ( <b>+0.0192</b> )

Table 2: Result for beam search decoding, controllability of each dialogue act (adversarial-explicit). The score in round brackets indicates the improvement from conditional-NCM w/ SRU.

Dialogue acts	Prec.	Recall	F1 (improv.)
Inform (46%)	0.8684	0.8973	0.8826 ( <b>+0.0164</b> )
Questions (30%)	0.9243	0.9738	0.9484 ( <b>+0.0281</b> )
Directives (16%)	0.7628	0.6652	0.7107 ( <b>+0.1152</b> )
Commissive (8%)	0.6338	0.5579	0.5934 ( <b>+0.0382</b> )
Weighted avg	0.8477	0.8531	0.8494 ( <b>+0.0257</b> )

Table 3: Result for random sampling, controllability of each dialogue act (adversarial-explicit). The score in round brackets indicates the improvement from conditional-NCM w/ SRU.

tags. Our model generated more discriminative sentences even if these classes have similar attribute.

## 6.2 Results of Human-Subjective Evaluation

Table 4 and Table 5 show human evaluation results for naturalness and controllability, respectively. We used beam search (beam width of 5) for generating examples to be evaluated. Regarding the naturalness of responses (Table 4), models used adversarial learning produced a more acceptable response to the dialogue context. Regarding the controllability of response generation (Table 5), the adversarial-explicit model achieved the best performance among the compared models.

In summary, the proposed model based on



Models	+2	+1	+0
Conditional-NCM w/ SRU	0.06	0.43	0.51
Adversarial-Implicit	0.04	0.53	0.43
Adversarial-Explicit	0.06	0.51	0.42

Table 4: Response quality of each model. This table show the distribution of scores.

Models	Accuracy	Weighted-F1
Conditional-NCM w/ SRU	0.7432	0.7583
Adversarial-Implicit	0.7058	0.6810
Adversarial-Explicit	<b>0.7971</b>	<b>0.7868</b>

Table 5: Controllability of response generation. Table shows filtered results of contradicted responses. Weighted-F1 is a weighted average of F1 score of each dialogue act.

adversarial learning with multi-class objective achieved the best controllability, the main focus of this paper, even if it realized a comparable naturalness to existing methods.

## 7 Conclusion

In this paper, we introduced an extended framework of the generative adversarial network that is optimized by both conditioned generation and discrimination of dialogue act classes. Experimental results showed that our conditional response generation model improved both the response quality and controllability in neural conversation generation. In future works, we will examine the possibility of incorporating our adversarial framework in various generation approaches (Serban et al., 2017; Shen et al., 2017; Zhou and Wang, 2018) to build a more generalized conditional response generation model. We will also focus on different types of labels to be used as conditions.

## Acknowledgements

This research and development work was supported by the JST PRESTO (JPMJPR165B) and JST CREST (JPMJCR1513).

## References

Kristy Elizabeth Boyer, Eun Young Ha, Robert Phillips, Michael D Wallis, Mladen A Vouk, and James C Lester. 2010. Dialogue act modeling in a complex task-oriented domain. In *Proc. of SIG-DIAL*, pages 297–305.

Cameron B Browne, Edward Powley, Daniel Whitehouse, Simon M Lucas, Peter I Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. 2012. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games*, 4(1):1–43.

Harry Bunt, Jan Alexandersson, Jae-Woong Choe, Alex Chengyu Fang, Koiti Hasida, Volha Petukhova, Andrei Popescu-Belis, and David R Traum. 2012. Iso 24617-2: A semantically-based standard for dialogue annotation. In *Proc. LREC*, pages 430–437.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the second international conference on Human Language Technology Research*, pages 138–145. Morgan Kaufmann Publishers Inc.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Chenyang Huang, Osmar Zaiane, Amine Trabelsi, and Nouha Dziri. 2018. Automatic dialogue generation with expressed emotions. In *Proc. of NAACL-HLT*, volume 2, pages 49–54.

Simon Keizer and Verena Rieser. 2017. Towards learning transferable conversational skills using multi-dimensional dialogue modelling. In *Proc. of SEM-DIAL*, pages 158–159.

Xiang Kong, Bohan Li, Graham Neubig, Eduard Hovy, and Yiming Yang. 2019. An adversarial approach to high-quality, sentiment-controlled neural dialogue generation. In *AAAI 2019 Workshop on Reasoning and Learning for Human-Machine Dialogues (DEEP-DIAL 2019)*, Honolulu, Hawaii.

Tao Lei, Yu Zhang, and Yoav Artzi. 2017. Training rnns as fast as cnns. *arXiv preprint arXiv:1709.02755*.

Tao Lei, Yu Zhang, Sida I Wang, Hui Dai, and Yoav Artzi. 2018. Simple recurrent units for highly parallelizable recurrence. In *Proc. of EMNLP*, pages 4470–4481.

Jiwei Li, Michel Galley, Chris Brockett, Georgios P. Spithourakis, Jianfeng Gao, and William B. Dolan. 2016. A persona-based neural conversation model. *CoRR*, abs/1603.06155.

Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. 2017a. Adversarial learning for neural dialogue generation. In *Proc. of EMNLP*, pages 2157–2169.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017b. Dailydialog: A manually labelled multi-turn dialogue dataset. In *Proc. of IJCNLP*, volume 1, pages 986–995.

- Ziming Li, Julia Kiseleva, and Maarten de Rijke. 2019. Dialogue generation: From imitation learning to inverse reinforcement learning. In *Proc. of AACL*.
- Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proc. of EMNLP*, pages 2122–2132.
- Toyomi Meguro, Ryuichiro Higashinaka, Yasuhiro Minami, and Kohji Dohsaka. 2010. Controlling listening-oriented dialogue using partially observable markov decision processes. In *Proc. of ACL*, pages 761–769.
- Augustus Odena. 2016. Semi-supervised learning with generative adversarial networks. *arXiv preprint arXiv:1606.01583*.
- Alice H Oh and Alexander I Rudnicky. 2000. Stochastic language generation for spoken dialogue systems. In *ANLP-NAACL 2000 Workshop: Conversational Systems*.
- Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proc. of AACL*, pages 3776–3784.
- Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Thirty-First AAAI Conference on Artificial Intelligence*.
- Xiaoyu Shen, Hui Su, Yanran Li, Wenjie Li, Shuzi Niu, Yang Zhao, Akiko Aizawa, and Guoping Long. 2017. A conditional variational framework for dialog generation. In *Proc. of ACL*, volume 2, pages 504–509.
- Tomohide Shibata, Yusuke Egashira, and Sadao Kurohashi. 2016. Chat-like conversational system based on selection of reply generating module with reinforcement learning. In *Situated Dialog in Speech-Based Human-Computer Interaction*, pages 63–69. Springer.
- Xiao Sun, Xinmiao Chen, Zhengmeng Pei, and Fujii Ren. 2018. Emotional human machine conversation generation based on seqgan. In *Proc. of ACII Asia*, pages 1–6. IEEE.
- Xiao Sun, Xiaoqi Peng, and Shuai Ding. 2017. Emotional human-machine conversation generation based on long short-term memory. *Cognitive Computation*, 10:389–397.
- Zhiliang Tian, Rui Yan, Lili Mou, Yiping Song, Yansong Feng, and Dongyan Zhao. 2017. How to make context more useful? an empirical study on context-aware neural conversational models. In *Proc. of ACL*, volume 2, pages 231–236.
- Quan Hung Tran, Ingrid Zukerman, and Gholamreza Haffari. 2017. A hierarchical neural model for learning sequences of dialogue acts. In *Proc. of ACL*, volume 1, pages 428–437.
- Yi-Lin Tuan and Hung-Yi Lee. 2019. Improving conditional sequence generative adversarial networks by stepwise evaluation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.
- Oriol Vinyals and Quoc V. Le. 2015. A neural conversational model. *CoRR*, abs/1506.05869.
- Tsung-Hsien Wen, Milica Gasic, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. In *Proc. of EMNLP*, pages 1711–1721.
- Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256.
- Wei Xu and Alexander I Rudnicky. 2000. Task-based dialog management using an agenda. In *ANLP-NAACL 2000 Workshop: Conversational Systems*.
- Koichiro Yoshino and Tatsuya Kawahara. 2015. Conversational system for information navigation based on pomdp with user focus tracking. *Computer Speech & Language*, 34(1):275–291.
- Steve Young, Milica Gašić, Simon Keizer, François Mairesse, Jost Schatzmann, Blaise Thomson, and Kai Yu. 2010. The hidden information state model: A practical framework for pomdp-based spoken dialogue management. *Computer Speech & Language*, 24(2):150–174.
- Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proc. of AAAI*, pages 2852–2858.
- Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. 2017. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *Proc. of ACL*, volume 1, pages 654–664.
- Hao Zhou, Minlie Huang, Tianyang Zhang, Xiaoyan Zhu, and Bing Liu. 2018. Emotional chatting machine: Emotional conversation generation with internal and external memory. In *Proc. of AAAI*.
- Xianda Zhou and William Yang Wang. 2018. Mojtalk: Generating emotional responses at scale. In *Proc. ACL*, volume 1, pages 1128–1137.