Information Engineering Express
International Institute of Applied Informatics
2024, Vol. 10, No. 1, IEE821

Improving the Consistency of Dialog Models Through Speaker Separation Learning

Sakuei Onishi [*] , Takamune Onishi [†] , Hiromitsu Shiina [‡]

Abstract

In recent years, dialog systems, a type of application in the field of natural language processing, have become more prevalent in our daily lives, such as through help desk services. In dialog response generation, responses generated for a specific context may differ from those for other contexts not only grammatically but also semantically in some cases. Thus, simply applying translation technologies would cause issues with the diversity of the generated responses. Previous studies, such as VHRED and GVT, used sampled latent variables for response generation to achieve response diversity. In this study, we propose a method (extended GVTSC) for classifying dialogs before reflecting them in internal dialog processing, in addition to the characteristics of each speaker, to improve diversity while maintaining consistency.

Keywords: Dialogue System, User-RNN, Conditional Variational Autoencoder, Global Variational Transformer, Extended GVT.

1 Introduction

The field of natural language processing is dramatically changing with the development of deep learning models. The Transformer[1] model has been proposed as an alternative to RNN and LSTM[2][3][4], which are suited to sequence-type data processing and can generate sentences. Language processing analysis that takes context into account is now possible, as seen in BERT[5], one of Transformer ' s models. Meanwhile, GTP-2[6] is capable of generating word definitions. The current dialog systems that are used for practical purposes are not generation-based dialog systems[7] that may generate inaccurate or risk-averse responses[8][9], but rather rule-based dialog systems where accuracy is guaranteed by manually generated rules.

RNN-based HRED[10] and VHRED[11] models have been announced. However, because these models do not distinguish between utterances made by two parties in a dialog, the characteristics of each party may become confused, resulting in a loss of consistency in response generation. The GVT model[12], in which the Transformer model is applied to

[*] Graduate School of Informatics, Okayama University of Science, Okayama, Japan
[†] Systems Nakashima, Okayama, Japan
[‡] A Faculty of Informatics, Okayama University of Science, Okayama, Japan

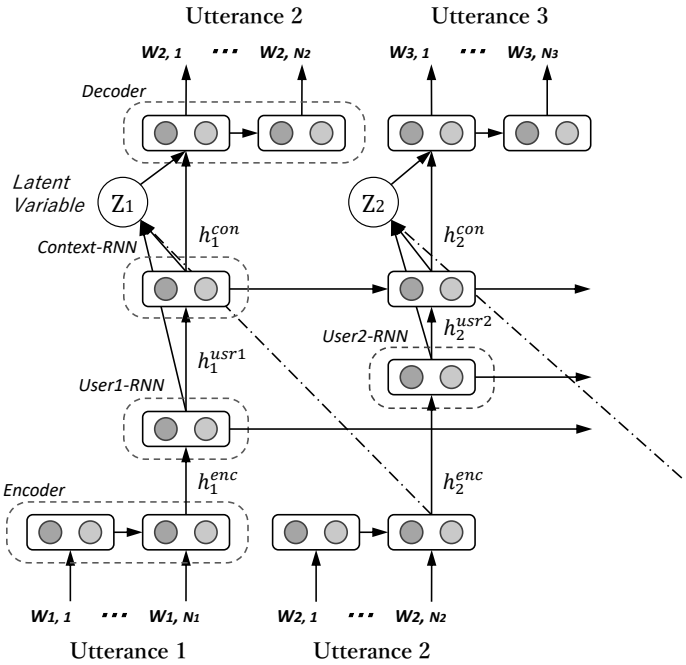

Figure 1: User-RNN added to VHRED model

dialog, also loses the characteristics of each speaker in latent variables generated from the entire context, causing issues with consistent response generation.

Therefore, in this study, we propose a VHRED model with added User-RNN (VHRED + User-RNN) that retains utterance information of each person in a two-person dialog, and an extended GVT model where latent variables of each speaker are added to the GVT model to improve response consistency.

The GVT model is a method that uses sampled latent variables for Decoder input, where speaker characteristics are represented by latent variables and sampled to achieve response diversity. However, a previous study indicated that latent variables tend to reduce the consistency of generated responses[13]. Thus, this study proposes a method for preparing speaker characteristic vectors via clustering and using them in context encoding (GVTSC) to improve consistency and diversity through abstraction of speaker characteristics and to add these speaker characteristics to Encoder so that the characteristics of each speaker are considered. The study also proposes a method that combines the method of adding latent variables for each speaker with the method of clustering speaker characteristic vectors (extended GVTSC).

To evaluate this model, the diversity of the generated response was evaluated using Dist-n[14], an automated metric, and the similarity to the reference response was evaluated using the BERT score[15]. “The Open 2 Channel Dialog Corpus”[16] was used as the dataset, and the experimental result showed improvements in the diversity of the generated responses.

2 VHRED Model with Added User-RNN

The VHRED model ensures the dialog flow with Context-RNN and uses it to generate responses. However, because utterances from two speakers are handled together, their princi-

ples and assertions become mixed, and consistency may be lost during response generation. Thus, we added User-RNN, which secures the utterance information of each speaker in a two-person dialog to improve response consistency. User1-RNN and User2-RNN were the two User-RNN. In a dialog, utterances with odd numbers were made by the User 1, while those with even numbers were made by User 2. To keep track of each speaker's information, odd-number utterances were processed using User1-RNN, and even-number utterances were processed using User2-RNN.

The model is depicted in Figure 1. User-RNN has the same structure as Context-RNN, and the Encoder output was entered into User-RNN, and the output was entered into Context-RNN. Further, for an utterance by User 1, the Encoder output was entered into User1-RNN. For an utterance by User 2, Encoder output was entered into User2-RNN. In addition to the Encoder and Context-RNN outputs, the output from each User-RNN was used to generate a prior probability distribution and posterior probability distribution to sample latent variables.

When utterance $\mathrm{x}_1, \ldots, \mathrm{x}_{n-1}$ was given as the context in response generation, utterance x_n was predicted and generated as follows. Here, n is the number of turns for the utterance, and f_θ^{enc}, f_θ^{con}, f_θ^{usr1}, f_θ^{usr2} and f_θ^{dec} are the calculations of the Encoder, Context-RNN, User1-RNN, User2-RNN, and Decoder, respectively.

The utterance immediately before, x_{n-1}, is entered into Encoder, it produces a hidden layer vector, h_{n-1}^{enc}.

$$h_{n-1}^{enc} = f_\theta^{enc}(\mathrm{x}_{n-1}) \tag{1}$$

If $n-1$ was an odd number, h_{n-1}^{enc} was used to update the hidden layer vector of User1-RNN and produce h_{n-1}^{usr1}. Similarly, if $n-1$ was an even number, h_{n-1}^{enc} was used to update the hidden layer vector of User2-RNN and produce h_{n-1}^{usr2}.

$$h_{n-1}^{usr1} = f_\theta^{usr1}(h_{n-3}^{usr1}, h_{n-1}^{enc}) \tag{2}$$

$$h_{n-1}^{usr2} = f_\theta^{usr2}(h_{n-3}^{usr2}, h_{n-1}^{enc}) \tag{3}$$

Furthermore, Context-RNN used the hidden layer vector of User-RNN to update the hidden layer vector, resulting in h_{n-1}^{con}.

$$h_{n-1}^{con} = \begin{cases} f_\theta^{con}(h_{n-2}^{con}, h_{n-1}^{usr1}) & \text{if } n-1 \text{ is odd} \\ f_\theta^{con}(h_{n-2}^{con}, h_{n-1}^{usr2}) & \text{if } n-1 \text{ is even} \end{cases} \tag{4}$$

Further, the latent variable $\mathbf{z_{n-1}}$ was sampled. It follows the normal distribution where mean and variance are determined by functions, μ_{prior} and σ_{prior},

$$p_\theta(\mathbf{z}_{n-1}|\mathrm{x}_{<n}) = \mathcal{N}(\mathbf{z}|\mu_{prior}, \sigma_{prior}, \mathbf{I}) \tag{5}$$

where μ_{prior} and σ_{prior} are defined as follows. $\mathbf{MLP}_\theta$ indicates calculation by multilayer perceptrons.

$$\mu_{prior} = \begin{cases} \mathbf{MLP}_\theta(h_{n-1}^{con}, h_{n-1}^{usr1}) & \text{if } n-1 \text{ is odd} \\ \mathbf{MLP}_\theta(h_{n-1}^{con}, h_{n-1}^{usr2}) & \text{if } n-1 \text{ is even} \end{cases} \tag{6}$$

$$\sigma_{prior} = \begin{cases} Softplus_\theta(\mathbf{MLP}_\theta(h_{n-1}^{con}, h_{n-1}^{usr1})) & \text{if } n-1 \text{ is odd} \\ Softplus_\theta(\mathbf{MLP}_\theta(h_{n-1}^{con}, h_{n-1}^{usr2})) & \text{if } n-1 \text{ is even} \end{cases} \tag{7}$$

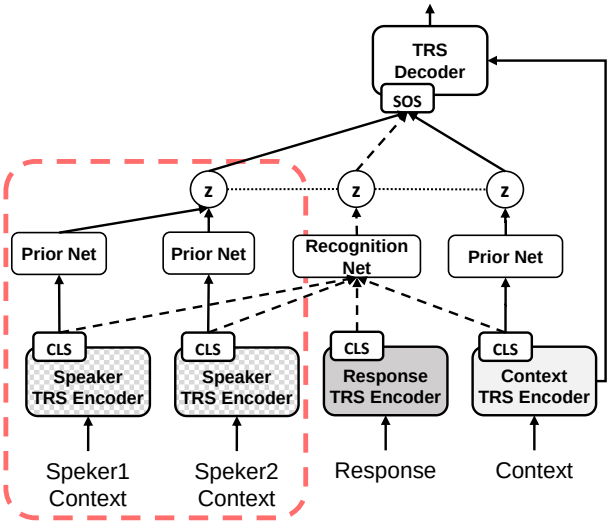

Figure 2: Structure of the extended GVT model

Finally, Decoder generates a response, $\mathrm{x_n}$, using the Context-RNN output, h^{con}_{n-1}, and a latent variable, $\mathbf{z_{n-1}}$.

$$p_\theta(\mathrm{x}|\mathrm{x}_{<n}) = f^{dec}_\theta(x|h^{con}_{n-1}, \mathbf{z}_{n-1}) \tag{8}$$

While learning, $\mathbf{z_{n-1}}$ follows a normal distribution where mean and variance are determined by the functions $\mu_{posterior}$ and $\sigma_{posterior}$, where $\mu_{posterior}$ and $\sigma_{posterior}$ are defined as follows.

$$\mu_{posterior} = \begin{cases} \mathbf{MLP}_\theta(h^{enc}_n, h^{con}_{n-1}, h^{usr1}_{n-1}) & \text{if } n-1 \text{ is odd} \\ \mathbf{MLP}_\theta(h^{enc}_n, h^{con}_{n-1}, h^{usr2}_{n-1}) & \text{if } n-1 \text{ is even} \end{cases} \tag{9}$$

$$\sigma_{posterior} = \begin{cases} Softplus_\theta(\mathbf{MLP}_\theta(h^{enc}_n, h^{con}_{n-1}, h^{usr1}_{n-1})) & \text{if } n-1 \text{ is odd} \\ Softplus_\theta(\mathbf{MLP}_\theta(h^{enc}_n, h^{con}_{n-1}, h^{usr2}_{n-1})) & \text{if } n-1 \text{ is even} \end{cases} \tag{10}$$

$$h^{enc}_n = f^{enc}_\theta(\mathrm{x}_n) \tag{11}$$

3 GVT Model with Information of Each Speaker Extended

The GVT model is one in which the CVAE model[17] and an RNN model used for dialog, has been rewritten as Transformer. In case of GVT, utterances are entered as a context without differentiating speakers. Thus, a latent variable is generated from the entire context, and the characteristics of each speaker are diluted. Therefore, in this study, we propose an extended GVT model in which utterances are separated and input for each speaker and the latent variable of each speaker is used to consider the characteristics of each speaker. In case of the extended GVT, the structure considers changes in speakers as well as past utterances. Figure 2 shows a diagram of the extended GVT model that considers speakers. In contrast to the regular GVT model, there is a part that differentiates utterances by each speaker and enters the context (dotted line in Figure 2), where latent variable z is sampled from Prior Net for each speaker.

In case of TRS Encoder, the CLS token is added to the beginning of the input series, where Transformer calculates the output vector. The entire context of the dialog is entered

into the Context TRS Encoder to obtain the output vector. The context summarizes the utterances by two people in a dialog and can separate each speaker. To obtain the output vector, each speaker divides the context and inputs it to each Speaker TRS Encoder.

Prior and Recognition Net are used to sample z, where prior and posterior probability distributions were approximated using multilayer perceptron (MLP). Prior Net estimates the mean and variance of the context vector using MLP based on the output vector of the Speaker TRS Encoder or Context TRS Encoder CLS token. z is sampled from the normal distribution that follows the mean and variance. In case of Recognition Net, in addition to the Speaker TRS Encoder and Context TRS Encoder, the output vector of the Response TRS Encoder CLS token is also used to estimate the mean and variance of the vector of the entire dialog using MLP. The latent variable z is sampled from a normal distribution, with mean and variance estimated in the same way as the Prior Net. The output vector of the TRS Encoder CLS token can be considered as a vector that expresses the entire input; thus, prior and posterior probability distributions are generated from the output vector of the CLS token to sample the latent variable z.

In case of TRS Decoder, latent variables are used to generate responses by inserting the latent variable of the response speaker, as well as the normal latent variable of the SOS token at the beginning of the input series. Further, TRS Decoder uses the latent variable sampled by Recognition Net while learning, and uses the latent variable sampled from Prior Net during generation.

The extended GVT model optimizes the model by maximizing the Evidence Lower Bound (ELBO) below, where c denotes the context, c_{s1} denotes the speaker 1 context, c_{s2} denotes the speaker 2 context, x denotes the response, and z denotes the latent variable.

$$\begin{aligned}
&\mathscr{L}_{ELBO}(x,c) \\
&= \log p(x|c) \\
&\geqq \mathbb{E}_q(z|x,c)[\log p(x|z,c)] \\
&\quad - KL(q(z|x,c)\|p(z|c)) \\
&\quad - KL(s(z|x,c_{\mathrm{s1}},c)\|r(z|c_{\mathrm{s1}})) \\
&\quad - KL(s'(z|x,c_{\mathrm{s2}},c)\|r'(z|c_{\mathrm{s2}}))
\end{aligned} \tag{12}$$

where KL is the Kullback–Leibler divergence (KL divergence) between distributions and the prior probability distribution, and p, r, and r' can be defined using the following Equations,

$$p(z|c) \sim \mathscr{N}(\mu_p, \sigma_p^2) \tag{13}$$
$$r(z|c_{\mathrm{s1}}) \sim \mathscr{N}(\mu_r, \sigma_r^2) \tag{14}$$
$$r'(z|c_{\mathrm{s2}}) \sim \mathscr{N}(\mu_{r'}, \sigma_{r'}^2) \tag{15}$$

where

$$\left[\mu_p, \log(\sigma_p^2)\right] = \mathrm{MLP}_p(c) \tag{16}$$
$$\left[\mu_r, \log(\sigma_r^2)\right] = \mathrm{MLP}_r(c_{\mathrm{s1}}) \tag{17}$$
$$\left[\mu_{r'}, \log(\sigma_{r'}^2)\right] = \mathrm{MLP}_{r'}(c_{\mathrm{s2}}) \tag{18}$$

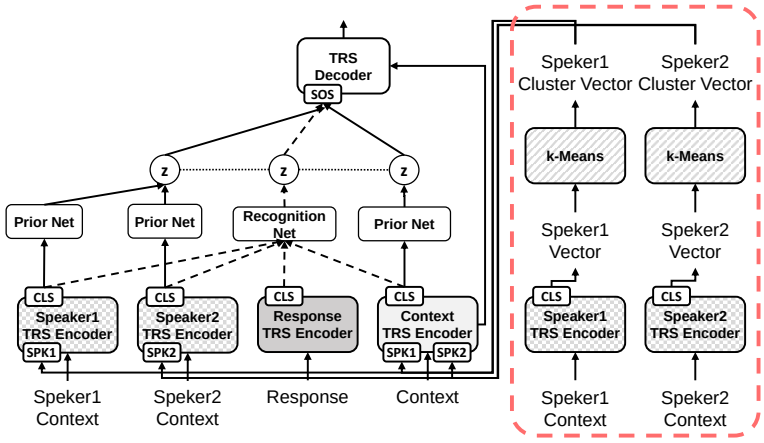

Figure 3: Structure of the extended GVTSC model

and posterior probability distribution, q, s, and s' are defined by the following Equations,

$$q(z|x,c) \sim \mathcal{N}(\mu_q, \sigma_q^2) \tag{19}$$

$$s(z|x,c_{s1},c) \sim \mathcal{N}(\mu_s, \sigma_s^2) \tag{20}$$

$$s'(z|x,c_{s2},c) \sim \mathcal{N}(\mu_{s'}, \sigma_{s'}^2) \tag{21}$$

where

$$\left[\mu_q, \log(\sigma_q^2)\right] = \mathrm{MLP}_q(x,c) \tag{22}$$

$$\left[\mu_s, \log(\sigma_s^2)\right] = \mathrm{MLP}_s(x,c_{s1},c) \tag{23}$$

$$\left[\mu_{s'}, \log(\sigma_{s'}^2)\right] = \mathrm{MLP}_{s'}(x,c_{s2},c) \tag{24}$$

KL annealing[18] and bag-of-words (BoW) loss[17][19] are incorporated as a result of KL vanishing, which occurs when the Decoder stops considering the information of the latent variable z as the learning progresses. KL annealing is a method in which the KL divergence value in Equation 12 is assigned a weight that increases linearly from 0 to 1 as the learning progresses. BoW loss is a method that includes subtasks that estimate the set of words that are included in the response to strengthen the relationship between the latent variable and words in the response.

4 Extended GVTSC Model with Added Speaker Clustering

We propose a model in which a speaker characteristic vector is prepared via clustering and used to encode the context. To consider the characteristics of each speaker and improve consistency and diversity, we propose a model in which a method that abstracts and adds speaker characteristics via clustering to the Encoder is added to the extended GVT model. Figure 3 shows the schematics of the extended GVTSC model. Compared to the extended GVT model, a part that prepares the speaker characteristic vector through clustering (dotted line in Figure 3) has been added to encode the context.

Next, let us discuss the processing of the extended GVTSC model. First, a speaker characteristic vector is prepared via clustering. The context summarizes the utterances of two people in a dialog and can be divided according to each speaker. Thus, the dialog

context is divided in relation to each speaker for processing. Because the processing for each speaker is the same, let us discuss the case for speaker 1. Speaker1 TRS Encoder encodes the context of the speaker 1. In case of TRS Encoder, the CLS token is added to the beginning of the input series, where Transformer calculates the output vector. The CLS token vector is obtained as the context vector of the speaker 1 (Speaker1 Vector). Clustering is performed for Speaker1 Vector. In this study, we used k-Means for clustering. Cluster number k must be determined in an experiment, as in the hyperparameter. Based on the clustering result, the cluster for Speaker1 Vector is predicted and the center vector of the cluster (Speaker1 Cluster Vector) is acquired. To obtain the Speaker2 Cluster Vector, the same procedure as for Speaker 1 is followed. TRS Encoder trained in response generation is shared by TRS Encoder used for clustering. However, there is no training in the back propagation of error during the clustering process.

The Context TRS Encoder uses the entire context of the dialog to obtain the output vector. Tokens for each speaker (SPK1, SPK2) are added to the input series during context encoding, with Speaker1 Cluster Vector being input to SPK1 and Speaker2 Cluster Vector being input to SPK2. The context is divided in relation to each speaker, which becomes the input for each Speaker TRS Encoder to obtain the output vector. At this point, the token of speaker 1 (SPK1) is added to the input series, and the Speaker1 Cluster Vector is entered. The token of the speaker 2 (SPK2) is added to the input series for the Speaker2 TRS Encoder and Speaker2 Cluster Vector is entered. When encoding the context for each speaker, we used the characteristic vector of each speaker to encode speaker characteristics that were further considered.

Latent variable z is sampled from Prior Net and Recognition Net, where the prior and posterior probability distributions were approximated with MLP. Prior Net estimates the mean and variance of the context vector with MLP based on the output vector of the CLS token of the Speaker TRS Encoder or Context TRS Encoder. z is sampled from a normal distribution with a mean and variance. Recognition Net estimates the mean and variance of the entire dialog vector with MLP using the output vectors of the Response TRS Encoder CLS token, Speaker TRS Encoder, and Context TRS Encoder. z is sampled from a normal distribution that follows the mean and variance, which are estimated in the same way as in Prior Net. Because the output vector of the TRS Encoder CLS token can be considered as the vector that represents the entire input, prior and posterior probability distributions are generated from the output vector of the CLS token and z is sampled.

The latent variable of the response speaker is entered into the TRS Decoder so that it can be used as the latent variable for response generation, as well as the normal latent variable of the SOS token at the beginning of the input series. At this point, TRS Decoder uses the latent variable sampled from Recognition Net during learning and the latent variable sampled from Prior Net during generation.

The extended GVTSC model optimizes the model in the same manner as the extended GVT by maximizing the ELBO below, with c as the context, c_{s1} as the speaker 1 context, c_{s2} as the speaker 2 context, x as the response, z as the latent variable, v_{s1} as the speaker 1

cluster vector, and v_{s2} as the speaker 2 cluster vector.

$$\begin{aligned}
&\mathscr{L}_{ELBO}(x,c)\\
&=\log p(x|c)\\
&\geqq \mathbb{E}_q(z|x,c,v_{s1},v_{s2})[\log p(x|z,c,v_{s1},v_{s2})]\\
&\quad - KL(q(z|x,c,v_{s1},v_{s2})\|p(z|c,v_{s1},v_{s2}))\\
&\quad - KL(s(z|x,c_{s1},c,v_{s1},v_{s2})\|r(z|c_{s1},v_{s1}))\\
&\quad - KL(s'(z|x,c_{s2},c,v_{s1},v_{s2})\|r'(z|c_{s2},v_{s2}))
\end{aligned} \tag{25}$$

where KL is the KL divergence between distributions and the prior probability distribution, and p, r, and r', are defined with the following Equations,

$$p(z|c,v_{s1},v_{s2}) \sim \mathscr{N}(\mu_p,\sigma_p^2) \tag{26}$$
$$r(z|c_{s1},v_{s1}) \sim \mathscr{N}(\mu_r,\sigma_r^2) \tag{27}$$
$$r'(z|c_{s2},v_{s2}) \sim \mathscr{N}(\mu_{r'},\sigma_{r'}^2) \tag{28}$$

where

$$\left[\mu_p,\log(\sigma_p^2)\right] = \mathrm{MLP}_p(c,v_{s1},v_{s2}) \tag{29}$$
$$\left[\mu_r,\log(\sigma_r^2)\right] = \mathrm{MLP}_r(c_{s1},v_{s1}) \tag{30}$$
$$\left[\mu_{r'},\log(\sigma_{r'}^2)\right] = \mathrm{MLP}_{r'}(c_{s2},v_{s2}) \tag{31}$$

and posterior probability distribution, and q, s, and s', are defined by the following Equations,

$$q(z|x,c,v_{s1},v_{s2}) \sim \mathscr{N}(\mu_q,\sigma_q^2) \tag{32}$$
$$s(z|x,c_{s1},c,v_{s1},v_{s2}) \sim \mathscr{N}(\mu_s,\sigma_s^2) \tag{33}$$
$$s'(z|x,c_{s2},c,v_{s1},v_{s2}) \sim \mathscr{N}(\mu_{s'},\sigma_{s'}^2) \tag{34}$$

where

$$\left[\mu_q,\log(\sigma_q^2)\right] = \mathrm{MLP}_q(x,c,v_{s1},v_{s2}) \tag{35}$$
$$\left[\mu_s,\log(\sigma_s^2)\right] = \mathrm{MLP}_s(x,c_{s1},c,v_{s1},v_{s2}) \tag{36}$$
$$\left[\mu_{s'},\log(\sigma_{s'}^2)\right] = \mathrm{MLP}_{s'}(x,c_{s2},c,v_{s1},v_{s2}) \tag{37}$$

KL annealing and BoW loss are incorporated, as in the extended GVT.

5 Evaluation Experiment

5.1 Evaluation by Benchmark

“ Open 2 Channel Dialog Corpus ” was used as the dataset. We used SentencePiece to divide it into subwords as a preprocessing step. We used Dist-n to evaluate the diversity of the generated response and the BERT score to evaluate the similarities to the reference response as an automatic evaluation method. Dist-N calculates the proportion of the number of types of N-grams relative to the total number of N-grams. When this ratio was higher, the

Table 1: Automatic evaluation of each model

Open 2 Channel Dialog Corpus				
	Diversity			**Simirality**
Model	**Dist-1**	**Dist-2**	**Dist-3**	**BERT Score**
1-turn				
VHRED	0.013	0.220	0.608	0.650
VHRED + User-RNN	0.013	0.225	0.632	**0.651**
GVT	0.008	0.392	0.935	0.643
Extended GVT	0.018	0.445	0.908	0.650
GVTSC	0.008	0.443	0.952	0.643
Extended GVTSC	**0.020**	**0.567**	**0.967**	0.649
Actual response	0.017	0.552	0.926	-
3-turn				
VHRED	0.036	0.330	0.699	0.653
VHRED + User-RNN	**0.038**	0.347	0.724	0.653
GVT	0.007	0.361	0.911	0.643
Extended GVT	0.017	0.413	0.886	**0.654**
GVTSC	0.008	0.405	0.931	0.644
Extended GVTSC	0.017	**0.515**	**0.949**	0.652
Actual response	0.017	0.552	0.926	-

diversity was higher. The BERT score uses the pre-trained BERT embeddings to evaluate the similarities of the responses generated by the model to the reference response.

Table 1 shows the results of the automatic evaluation of the responses generated by each model. The GVTSC model adds speaker clustering to the GVT model in the same way as the extended GVTSC model. The number of clusters in the clustering (k-Means) was 8 and 3 for the GVTSC model and the extended GVTSC model, respectively, based on the preliminary experiment.

A model that included User-RNN in addition to the VHRED model, as presented in Table 1, improved diversity for both 1-turn and 3-turn; however, there was no difference in the evaluation of similarities.

Compared to GVT, the extended GVT model with an additional Encoder for each speaker improved in Dist-1 and Dist-2 but declined in Dist-3. In the evaluation of similarities, the 1-turn and 3-turn BERT scores improved by approximately 0.007 and 0.011, respectively.

All diversity evaluations for the GVTSC model showed an improvement compared to the GVT model. The GVTSC model outperformed the extended GVT model by approximately 0.045 in Dist-3.

However, there was little difference in the evaluation of the similarities between the extended GVT model and GVTSC model. Further, in terms of diversity, both 1-turn and 3-turn evaluations improved. Dist-2, in particular, improved the evaluation by at least 0.1. Considering the speaker characteristics in the encoding process by Encoder, it had an impact on the sampling of the latent variables and output vector of each token that the Encoder used for the attention of the Decoder.

Table 2: Example of response

Context
Utterance1: 頑張ってくれ。有名どころばかりだ **Utterance2:** 分かるなら秒で答えるけどピンとも来なくて... あんまり読書しないからなんだけども... **Utterance3:** おすすめばかりだから、このスレで出た小説は読んでくれ **Utterance4:** なら暇だし読んでみようかな。一番のオススメは? **Utterance5:** 一番は難しいな
Response
GVT: ああ、下手で小説王室とか聞いて分やってみる (出来そうとは言ってないそうか 別にいいと思うぞ。 **Extended GVT:** なんかありがとうございます読んではみる。とヒゲの H と **GVTSC:** せやな、キンだ... どう全然、風知らなくて日本から探していいな気だろ。そういうこと **Extended GVTSC:** 探してみるよ。でもその時点でめっちゃくちゃ苦手かな。読み y **Actual response:** えー。決めてよー

5.2 Comparison of Dialogue Generation

Examples of responses generated by each method to Open 2 Channel Dialog Corpus are shown in Table 2. In Table 2, the top row represents the context that is the dialogue, and the bottom row represents the response by each method when the context is input. The last row in the bottom row is the actual response.

The extended GVT and extended GVTSC models are highly relevant to the context. The GVT and GVTSC models produce responses with low fluency. The extended GVTSC model produces a diverse response to the last utterance of the context, which is different in direction from the reference response.

6 Conclusions

In this study, we proposed a method of using speaker information to improve consistency and diversity in dialogue response generation. First, for RNN-based models, we proposed a User-RNN that keeps speaker information separately and confirmed the improvement in diversity. Next, we proposed an extended GVT model that adds an Encoder to the Transformer-based GVT model to encode speaker-specific context and confirmed the improvement in diversity and similarity. Finally, we proposed an extended GVTSC model that uses pre-clustering to abstract speaker features to improve diversity while maintaining

consistency. The abstracted speaker information is added as a vector to the input of the Encoder so that the speaker information is considered in the Encoding process. The similarity evaluation showed almost no difference between the two models. The highest rating was obtained in the diversity evaluation, confirming that the diversity is close to the actual response.

In the future, we intend to develop a dialogue model suitable for domains with limited speaker attributes. In this study, clustering by k-means was used as a method for pre-creating speaker features, but the impact of other clustering and classification methods needs to be examined.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[2] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[3] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm," *Neural computation*, vol. 12, no. 10, pp. 2451–2471, 2000.

[4] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "Lstm: A search space odyssey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222–2232, 2016.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[6] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[7] O. Vinyals and Q. Le, "A neural conversational model," in *ICML Deep Learning Workshop 2015*, 2015.

[8] A. Sordoni, M. Galley, M. Auli, C. Brockett, Y. Ji, M. Mitchell, J.-Y. Nie, J. Gao, and B. Dolan, "A neural network approach to context-sensitive generation of conversational responses," in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Denver, Colorado: Association for Computational Linguistics, May–Jun. 2015, pp. 196–205. [Online]. Available: https://aclanthology.org/N15-1020

[9] R. Csáky, P. Purgai, and G. Recski, "Improving neural conversational models with entropy-based data filtering," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 5650–5669. [Online]. Available: https://aclanthology.org/P19-1567

[10] I. V. Serban, A. Sordoni, Y. Bengio, A. Courville, and J. Pineau, "Building end-to-end dialogue systems using generative hierarchical neural network models," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, ser. AAAI'16. AAAI Press, 2016, p. 3776–3783.

[11] I. Serban, A. Sordoni, R. Lowe, L. Charlin, J. Pineau, A. Courville, and Y. Bengio, "A hierarchical latent variable encoder-decoder model for generating dialogues," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, Feb. 2017. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/10983

[12] Z. Lin, G. I. Winata, P. Xu, Z. Liu, and P. Fung, "Variational transformers for diverse response generation," *arXiv preprint arXiv:2003.12738*, 2020.

[13] B. Sun, S. Feng, Y. Li, J. Liu, and K. Li, "Generating relevant and coherent dialogue responses using self-separated conditional variational AutoEncoders," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 5624–5637. [Online]. Available: https://aclanthology.org/2021.acl-long.437

[14] J. Li, M. Galley, C. Brockett, J. Gao, and B. Dolan, "A diversity-promoting objective function for neural conversation models," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, Jun. 2016, pp. 110–119. [Online]. Available: https://aclanthology.org/N16-1014

[15] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with bert," *In International Conference on Learning Representation*, 2019.

[16] M. Inaba, "A example based dialogue system using the open 2channel dialogue corpus," *Journal of Japanese Society for Artificial Intelligence*, vol. 87, pp. 129—-132, 2019.

[17] T. Zhao, R. Zhao, and M. Eskenazi, "Learning discourse-level diversity for neural dialog models using conditional variational autoencoders," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, Jul. 2017, pp. 654–664. [Online]. Available: https://aclanthology.org/P17-1061

[18] S. R. Bowman, L. Vilnis, O. Vinyals, A. Dai, R. Jozefowicz, and S. Bengio, "Generating sentences from a continuous space," in *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 10–21. [Online]. Available: https://aclanthology.org/K16-1002

[19] X. Zhou and W. Y. Wang, "MojiTalk: Generating emotional responses at scale," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 1128–1137. [Online]. Available: https://aclanthology.org/P18-1104