Semantic-aware Conditional Variational Autoencoder for one-to-many dialogue generation

Ye Wang, Jingbo Liao, Hong Yu* and Jiaxu Leng

Chongqing Key Laboratory of Computational Intelligence, Chongqing University of Posts and Telecommunications, Chongqing, 400065, China.

*Corresponding author(s). E-mail(s): yuhong@cqupt.edu.cn;

Abstract

Due to the miscellaneous ambiguity of semantics in open-domain conversation, current deep dialogue models disregard to detect potential emotional and action response features in the latent space, which leads to the general tendency to produce inaccurate and irrelevant sentences. To address this problem, we propose a semantic-aware conditional variational autoencoder (S-CVAE) that discriminates the sentiment and action responses features in the latent space for one-to-many open-domain dialogue generation. Specifically, explicit controllable variables are leveraged from the proposed module to create diverse conversational texts. This controllable variable can constrain the distribution of the latent space, disentangling the latent space features during training. Furthermore, the feature disentanglement improves the dialogue generation in terms of deep learning interpretability and text quality, which also reveals the latent features of different emotions on the logic of text generation.

Keywords: Open-domain dialogue generation, Generation diversity, Feature disentanglement, Explainable artificial intelligence

1 Introduction

Since the dialogue system is initially created, it has been of interest to a wide range of industries. In 1951, Turin [1] proposed the idea of human-machine dialogue to test the level of machine intelligence. Subsequently, question-and-answer models for the navigation system and the keyword search model continued the prototype of the dialogue system concept. In recent years, various deep neural networks and text generative models have been proposed to accelerate the development of dialogue applications, such as seq2seq [2, 3], transformer [4, 5], VAE and those variations [6, 7]. Currently, dialogue systems have been advanced from traditional statistical language models to neural network language models. Early dialogue systems are task-oriented and largely based on structured grammar rules. For example, movie ticket booking, email message searching and online customer service are all related applications. Traditional dialogue system ELIZA [8] is almost hand-crafted rules. Although the rules of ELIZA are clear and quick to implement, the approach contains several drawbacks in terms of expensive manually annotated data, rigid rules and insufficient flexibility. Compared to Young et al. [9] who treat the dialogue problem as a Markovian decision process and require a large number of task-specific semantic databases, the cost based on generative models has been significantly reduced. And the popularity of the deep sequence model seq2seq has dramatically improved the performance of open dialogues. Following this, generative models have been used extensively in dialogue systems. On short textual dialogues, Vinyals et al. [10] proposed an end-to-end generative model based on phrasal dialogues, which is a data-driven approach to generating relatively simple dialogues. Iulian et al. [11] using RNN and n-gram based models to reach the point of mimicking or even replicating human behavior in the corpus. Wang et al. [12] proposed an Attention-aware Bidirectional Multi-residual Recurrent Neural Network (ABMRNN) to overcome the deficiency of long-time dependency in RNN. To make dialogue responses more relevant, Li et al. [13] used the concept of mutual information and introduced the IMMM model to make the dialogue more consistent with semantic rules. In addition, Galetzka et al. [14] proposed a model based on transformer to generate a response with fixed contextual contexts. The model can reduce the spatial requirements without compromising knowledge accuracy and perceptual consistency, thus improving knowledge retrieval.

The end-to-end black-box model also has many drawbacks, as it cannot understand the decision logic and generation process inside its model. To address these shortcomings, a growing number of researchers are investigating the interpretability of deep learning. To solve the problem of low confidence in the prediction results of black-box models, Ribeiro et al. [15] proposed an interpretation method LIME for pre-trained models to implement the interpretation of the results of black-box models. Although LIME explains the results of the black-box model to some extent, the more complex and accurate the model is, the less interpretable it is. To address the problem of conflicting accuracy and interpretability, Lundberg et al. [16] combined with the Shapley Value in game

theory to proposed a SHAP Value to reflect the influence of the features in each sample and thus better explain the model. Zhang et al. [17] build on these works to successfully interpret the convolutional neural network, using decision trees. Meanwhile, feature disentanglement is a commonly used interpretable means in the field of generation. To disentangle the noise features used to generate samples, Chen et al. [18] proposed InfoGan. The model accomplishes the feature disentanglement of noise through mutual information, enhancing the interpretability of these features. Following this, Karras et al. [19] proposed StyleGan. And it maps noise to a vector in another space via a multilayer perceptron. StyleGan's generators gain the ability to disentangle features with this idea.

Therefore, we wish to perform interpretability studies on deep sequence models by mainly addressing the diversity of decoder responses. To summarise our contributions:

- We propose a semantic-aware conditional variational autoencoder that distinguishes the sentiment and action responses in open-domain dialogues. The proposed model is used to control explicitly controllable variables for creating diverse conversational texts.
- We disentangle the latent space features of the deep latent variable model. Different categories of dialogues are allowed to projected in separate locations of the latent space, enhancing the interpretability of text generation.
- We have comprehensively compared the current deep latent variable model with baselines on the Daily Dialog dataset, and the experimental results have fully validated our model.

2 Related Works

Open-domain dialogue systems are also called non-task-based dialogue systems. Two main approaches exist [20], one based on retrieval, learning to select a response from a database or repository that matches the current conversation. The second is a deep generative approach, which generates responses in dialogue that match the reality of the situation. The prevailing approach is still an end-to-end deep generative model. Sutskever et al. [2] proposed the seq2seq model and now it has become one of the most commonly used methods in the field of text generation. However, the seq2seq model tends to generate high-frequency generic responses, such as "I don't know". So Monroe et al. [21] proposed the RL model to improve the seq2seq method. They use developer-defined rewards instead of maximum likelihood estimation (MLE) as an optimization criterion. Thus the dialogue system is able to produce a more realistic response. Meanwhile, Alessandro et al. [22] discovered implicit logical dependencies between conversations. Then they proposed a neural network-based VHRED model to obtain more diverse results in dialogue generation. In addition, Shang et al. [23] proposed a new Neural Responding Machine (NRM)

framework. NRM improves on the encoder-decoder [24] deep learning framework. They use a probabilistic model to measure the correlation between the conversations, resulting in a set of phrased conversations that present a high degree of correlation.

For interpretability studies of deep generative models, the main approach nowadays is to train a disentangled latent space in a deep latent variable model. Higgins et al. [25] proposed the $\beta - VAE$ model, which adds a coefficient β to the KL term in the ELBO objective of the standard VAE, and the approach enhances the ability of the VAE model to disentangle. Hu et al. [26] proposed a new model for text generation. They combined VAE with GAN and introduced new variables in a similar way to InfoGAN, to disentangle the representation of text features after training. Wiseman et al. [27] proposed a neural template generative model based on a Hidden Semi-Markov Model (HSMM) decoder. HSMM allows controlling the diversity of generation and generating interpretable states during generation. Starting from interpretable representation learning for generative dialogue models, Zhao et al. [28] proposed two unsupervised VAE models, DI-VAE and DI-VST. These models can be combined with existing encoder-decoder frameworks to generate interpretable dialogue. See et al. [29] investigated two controlled methods of neural text generation, conditional training, and weighted decoding. At the same time, they controlled for four important attributes of small talk conversations: repetition, specificity, response relevance and questioning. Ficler et al. [30] investigated style in controlling text generation and defined multiple styles as parameters for training and learning. Li et al. [31] proposed a new model of paraphrase generation, DNPG. It decomposes the mechanism to make the generation of paraphrases more interpretable and controllable. Sato et al. [32] discussed the interpretability of adversarial training based on adversarial perturbations, limiting the direction of perturbations to the existing words in the word embedding space position. Then successfully generated reasonable adversarial text and interpretable visualizations of the perturbations in the input embedding space. Pang et al. [33] proposed a priori model based on latent spatial energy. The model combines a generated dense vector with an interpretable and classifiable symbolic vector. And its generator can generate text with high quality, diversity, and interpretability. Compared to the original chaotic latent space, Shi et al. [34] completed a distinguishable latent space with higher interpretability by introducing the exponential mixture distribution for replacing the Gaussian prior. Meanwhile, further latent semantic relationships between mixture components and the data are captured by leveraging such an exponential mixture distribution. Besides, to avoid the model collapse, the variational evidence lower bound is decomposed for acquiring the loss term.

To solve the one-to-many problem, Chen et al. [35] proposed a multi mapping mechanism to better capture the one-to-many relationship, where multiple mapping modules are employed as latent mechanisms to model the semantic mappings from an input post to its diverse responses. Bao et al. [36] proposed the first application of discrete latent variables combined with Transformer

structure to a generic dialogue domain. By introducing discrete latent variables, they can effectively model the one-to-many relationship between input and reply, thus achieving a significant improvement in conversation richness and fluency. Cui et al. [37] pointed out that the current CVAE-based approach relies only on discourse-level latent variables to model diversity, which is coarse-grained. Therefore, it is proposed to introduce a more fine-grained focus signal to measure the semantic focus of the conversation above and responses to produce more diverse and controllable responses. In contrast to existing research, related work on disentangling the latent space typically disentangles the latent space by decomposing penalty terms from the evidence lower bound. When they use the penalty terms to force the latent space to disentangle, we disentangle the latent space by introducing external labels through a classification task. Traditional generative models usually contain only one decoder, we construct multiple decoders with different semantics. Then we select the decoder by controllable variables, and the experiments show that the results are improved regarding the quality and diversity of the generated dialogues.

3 Proposed Scheme

3.1 Conditional Variational Autoencoder Motivation

In practical dialogue generation tasks, data is often presented in pairs, such as questions and responses in dialogue systems. So the data distribution we need to fit is the conditional probability distribution $P(X \mid C)$, where X is the output sequence and C is the input condition. The output sequence X is generally the output response in the dialogue system. The input conditions can be the emotion of the dialogue, the grammar, the text in question, or the context of the dialogue. Traditional end-to-end generative models are trained based on maximum likelihood estimation, resulting in a huge bias between training and testing. Therefore, the Conditional Variational Autoencoder [38] trained by optimising the evidence lower bound is well suited to the dialogue generation task. Its input encoder can encode the question text as condition C, the output encoder can encode the ground-truth response text as the output sequence X, and the decoder gets the output sequence X by decoding the latent variable. CVAE is similar to VAE in that the conditional probability distribution $P(X \mid C)$ is decomposed into Eq. 1 after the introduction of the latent variable z.

$$P(X \mid C) = \int_z p(z \mid C) P(X \mid z, C) dz \tag{1}$$

On the basis of Eq.1, we can obtain the evidence lower bound(ELBO), as shown in Eq.2.

$$\text{ELBO} = E_{q_\phi(z|X,C)} \left[\log P_\theta(X \mid z, C)\right] - KL\left(q_\phi(z \mid X, C) \| p_\varphi(z \mid C)\right) \tag{2}$$

The first expectation $E_{q_\phi(z|X,C)}[\log P_\theta(X \mid z, C)]$ is reconstruction loss. And $q_\phi(z \mid X, C)$ is approximate posterior distribution. During training, the latent variables are sampled from the approximate posterior distribution. Since the process of sampling is not derivatable, a re-parameterisation technique such as Eq.3 is required to solve the derivative problem. $P_\theta(X \mid z, C)$ is the conditional probability distribution for generating responses when the latent variable z and the input condition C are known.

$$z = \mu + \sigma * \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, I) \tag{3}$$

The second loss $KL(q_\phi(z \mid X, C) \| p_\varphi(z \mid C))$ is the KL dispersion of the prior distributions $p_\varphi(z \mid C)$ and approximate posterior distributions $q_\phi(z \mid X, C)$, used to approximate both distributions. In training we have both questions and responses, so we sample the latent variables in the approximate posterior distribution. For the test we only have the question data, so we need to sample the prior distribution. And we need to make the two distributions as close as possible. The prior distribution follows a multivariate Gaussian distribution $\mathcal{N}\left(\mu', \quad \sigma'^2 I\right)$ and the approximate posterior distribution follows $\mathcal{N}\left(\mu, \quad \sigma^2 I\right)$. μ, σ^2, μ', σ'^2 are computed from the outputs of two encoders by two multilayer perceptrons, a priori and recognition networks, respectively. ϕ, φ, θ are the parameters of a probability distribution.
The training process of the CVAE model is the process of maximising the evidence lower bound, so the loss function is the evidence lower bound taken as negative, as shown in Eq. 4.

$$\mathcal{L}_{\text{CVAE}} = -E_{q_\phi(z|X,C)}[\log P_\theta(X \mid z, C)] + KL(q_\phi(z \mid X, C) \| p_\varphi(z \mid C)) \tag{4}$$

3.2 Semantic-aware Conditional Variational Autoencoder

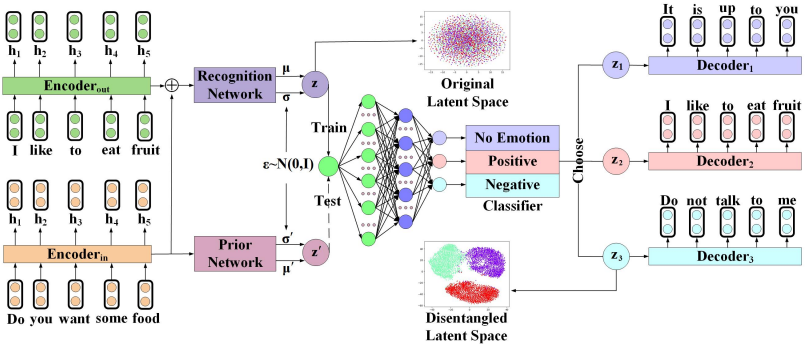


Fig. 1: Schematic diagram of S-CVAE model framework.

Our model (S-CVAE) (Figure 1) is optimised on the basis of Conditional Variational Autoencoders (CVAE). The goal of the model is to select different decoders to generate personalised, high-quality, interpretable and diverse text, guided by controllable variables. We use a controllable variables to guide the latent space to do feature disentangled, allowing the model to map different classes of conversations to different locations in the latent space. And the distance between conversations of the same category encoded into the latent space will be more similar than the distance between conversations of different categories. We choose different decoders to generate different dialogues by controlling the controllable variables, while enhancing the diversity and category controllability of dialogue generation. S-CVAE is divided into three processes: pre-train, train and test.

3.2.1 Pretrain

The pre-training process is to optimize the input encoder $\mathrm{Enc}_{in}(\cdot)$, the output encoder $\mathrm{Enc}_{out}(\cdot)$, the recognition network $\mathrm{RecogNetq}_{\phi}(z \mid X, C)$, a prior network $\mathrm{PriorNetp}_{\varphi}(z \mid C)$, a classifier $\mathrm{Clf}(\cdot)$ and a pre-trained decoder $\mathrm{Dec}_{\mathrm{pre}}(\cdot)$. The training data are dialogue data pairs (C, X, E) with emotion labels. C is the text sequences of questions $\{\mathbf{C}_1, \mathbf{C}_2, \cdots, \mathbf{C}_n\}$ in the dialogue. X is the text sequence of Ground-truth responses $\{\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_n\}$ in the dialogue. E is the emotion tag of the question text (No emotion, Positive, Negative). The label of the question text is chosen here because we are considering generating responses that correspond to the sentiment of the question text.
We use word2vec for word embedding of text, and represent the question sequence C as $\{e(C_1), e(C_2), \cdots, e(C_n)\}$, and the ground-truth response sequence X as $\{e(X_1), e(X_2), \cdots, e(X_n)\}$. $e(\cdot)$ denotes word embedding of text words, $e(X_i), e(C_i) \in \mathcal{R}^D$, D equals 300. The input to the input encoder is the word embedding $e(C)$ of the question text sequence, and the input to the output encoder is the word embedding $e(X)$ of the ground-truth response text sequence. Both encoders are implemented using a bidirectional optimized LSTM [39]. The input encoder $\mathrm{Enc}_{in}(\cdot)$ encodes the question text into a dense vector representation $r_C \in \mathcal{R}^D$ and the output encoder output encoder $\mathrm{Enc}_{out}(\cdot)$ encodes the ground-truth response text into a dense vector representation $r_x \in \mathcal{R}^D$: $r_C = \mathrm{Enc}_{in}(e(C))$ and $r_C = \mathrm{Enc}_{in}(e(C))$. After obtaining the feature representation vector of the text, it is fitted to the Gaussian parameters of the latent space Gaussian distribution. We use two multilayer perceptrons to achieve the fitting of the parameters, an recognition network $\mathrm{RecogNetq}_{\phi}(z \mid X, C)$ and a prior network $\mathrm{PriorNetp}_{\varphi}(z \mid C)$. The reconition network $\mathrm{RecogNetq}_{\phi}(z \mid X, C)$ is used to fit the parameters μ, σ of the approximate posterior distribution, $\mu, \sigma = \mathrm{RecogNetq}_{\phi}(z \mid X, C)$. The prior network $\mathrm{PriorNetp}_{\varphi}(z \mid C)$ is use to fit the parameters μ', σ' of priori distribution, $\sigma', \mu' = \mathrm{PriorNetp}_{\varphi}(z \mid C)$. After calculating the Gaussian parameters $\mu, \sigma, \mu', \sigma'$ of the approximate posterior and prior distributions, the KL scatter of the approximate posterior and prior distributions is calculated

according to Eq.5. dz is the dimensionality of the latent variable. Minimizing the KL distance we can approximate the approximate posterior and prior distributions.

$$-KL\left(q_{\phi} \| p_{\varphi}\right)=\frac{1}{2} \sum_{j=1}^{dz}\left[1+\log\left(\sigma_{j}^{2}\right)-\log\left(\sigma_{j}^{2}\right)-\frac{\sigma_{j}^{2}+\left(\mu_{j}-\mu_{j}^{\prime}\right)^{2}}{\sigma_{j}^{\prime 2}}\right] \tag{5}$$

We mentioned earlier that sampling directly from the latent space would cause the optimisation to fail, as the sampling process cannot be derivative. So reparameterize the latent variable z according to Eq.3, $z=\mu+\sigma * \varepsilon$, ε is obtained by sampling from the standard Gaussian distribution. During pre-training, the latent variable z is used as input to the classifier $\mathrm{Clf}(\cdot)$ and the initial hidden state of the pre-trained decoder $\mathrm{Dec}_{\mathrm{pre}}(\cdot)$.

The classifier $\mathrm{Clf}(\cdot)$ consists of a multilayer perceptron and a Softmax layer. The latent variable z is used as input to the multilayer perceptron, and the output of the MLP is used as input to the Sofmax. P(E) is the normalised probability that the current conversation belongs to each category. Finally the category with the highest probability is taken as the classification result. The training of the classifier is a supervised training process and the loss function we use is Negative Log Likelihood Loss. Label is the dialogue type label of the training set.

$$P(E)=Clf(z) \tag{6}$$

$$Clf(z)=\mathrm{Softmax}(o) \tag{7}$$

$$o=MLP(z) \tag{8}$$

$$\mathcal{L}_{Clf}=NLLLoss(P(E),\ \text{label}) \tag{9}$$

Since the dimension of the latent variable is not the same as the dimension of the decoder hidden state, we need to fit the latent variable to the dimension of the decoder hidden state before decoding. Then use it as the initialized hidden state h_0 and initialized cell state c_0 of the pre-trained decoder $\mathrm{Dec}_{\mathrm{pre}}(\cdot)$. The input to the decoder is a text sequence $\{<SOS>, \mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_n\}$ consisting of the ground-truth response from the conversation concatenate the start token $<SOS>$. The output of the decoder is mapped to a vector of word list sizes, taking the maximum value as the word $\bar{X}_i$ generated at step i. Finally we calculate the reconfiguration expectation loss $E_{q_{\phi}(z|X,C)}\left[\log P_{\theta}(X \mid z, C)\right]$.

$$\text{output }, (h_i, c_i)=\mathrm{Dec}_{pre}\left(e\left(X_i\right), (h_{i-1}, c_{i-1})\right) \tag{10}$$

$$\bar{X}_i=\arg\max\left(\mathrm{soft}\max\left(\text{ output }\cdot W_v+b_v\right)\right) s \tag{11}$$

In the pre-training phase, the loss function is the evidence lower bound and the reconstruction expectation loss plus the negative log-likelihood loss.

$$\begin{aligned}\mathcal{L}_{\mathrm{pre}}=&-E_{q_{\phi}(z|X,C)}\left[\log P_{\theta}(X \mid z, C)\right]\\&+KL\left(q_{\phi}(z \mid X, C) \| p_{\varphi}(z \mid C)\right)\\&+NLLLoss(P(E), label)\end{aligned} \tag{12}$$

Algorithm 1 The overall framework of the algorithm

Require: Data samples (C, X, E) = $(c_i, x_i, e_j)_{i=1}^{N}$
Ensure: Parameter(ϕ, φ, θ, π) convergence.

Initialize the parameters (ϕ, φ, θ,π).
if Pretrain **then**
  for iter from 1 to max_iter **do**
    Sampling (c, x, e) from the pretrain set.
    Calculate μ, $\sigma \leftarrow RecogNetq_{\phi}(z \mid X, C)$.
    Calculate μ', $\sigma' \leftarrow PriorNetp_{\varphi}(z \mid C)$.
    Reparameterize $z = \mu + \sigma * \varepsilon$.
    Calculate the category of dialogue by Eq.7.
    Calculate variational evidence lower bound by Eq.2.
    Calculate Negative Log Likelihood Loss by Eq.9.
    Update ϕ, φ, θ and π by gradient ascent by Eq.12.
  end for
end if
if Train **then**
  Fixed parameter(ϕ, φ,π)
  for iter from 1 to max_iter **do**
    Sampling (c, x, e) from the train set.
    Calculate μ, $\sigma \leftarrow RecogNetq_{\phi}(z \mid X, C)$.
    Calculate μ', $\sigma' \leftarrow PriorNetp_{\varphi}(z \mid C)$.
    Reparameterize $z = \mu + \sigma * \varepsilon$.
    Calculate Expected loss of reconfiguration by Eq.13
    Update θ by gradient ascent.
  end for
end if
if Test **then**
  Sampling (c, x, e) from the train set.
  Calculate μ', $\sigma' \leftarrow PriorNetp_{\varphi}(z \mid C)$.
  Reparameterize $z = \mu' + \sigma' * \varepsilon$.
  Calculate the category of dialogue by Eq.7.
  Select the corresponding decoder to generate the text by Eq.14
end if

3.2.2 Train

In the training phase, our goal is to optimise the no emotion decoder $Dec_{no}(\cdot)$, the positive decoder $Dec_{poi}(\cdot)$ and negative decoder $Dec_{neg}(\cdot)$. We split the original dialogue dataset into three sub-datasets by category, which was used for training the corresponding decoders. We must freeze the parameters of the encoders. This is because when we use different classes of data to train the corresponding decoders, the different decoders will interact with each other accordingly. For example, when training the no emotion decoder, the encoder can be optimized as well. However, when training the positive decoder by the

previous encoder, the encoder will also be advanced and the parameters of the encoder will be changed. Until the encoder completed the positive decoder training the no emotion decoder cannot decode and generate the dialogue correctly due to the change of the encoder. We found that the encoder, recognition network and a prior network were already optimised to a relatively good extent during pre-training. Therefore, in order to avoid the interaction between different decoders during training, we choose to fix the parameters of the encoder, recognition network, prior network and classifier during the training phase. Finally, only the decoder is trained for the purpose of generating different classes of responses. When training, different sub-datasets are used to train different classes of decoders. After training the different decoders are assembled into a complete model. The training process is the same as the pre-training process, but since only the decoder is optimised, the loss function is only the reconstruction loss function.

$$\mathcal{L}_{\text{train}} = -E_{q_\phi(z|X,C)}\left[\log P_\theta(X \mid z, C)\right] \tag{13}$$

3.2.3 Test

Since the test set cannot use the Ground-truth response X during the testing phase, only the question sequence C. Therefore, we only use the input encoder $Enc_{in}(\cdot)$ and fit the prior distribution $p_\varphi(z \mid C)$ and no longer fit the approximate posterior distribution $q_\phi(z \mid X, C)$. And then we obtain the latent variable z from the prior distribution $p_\varphi(z \mid C)$ using Eq. 3 sampling. The current conversation is classified using $argmax(Clf(z))$ and the corresponding decoder is selected to decode the generated conversation based on the classification result. The decoding process in the test phase is different from the training phase. Since the ground-truth response cannot be used, we change the input strategy of the decoder. The input to the decoder in the first step is the token $\langle SOS \rangle$, and the input to the next step is the output word from the previous step of the decoder. The decoder finishes decoding when it has generated the end token $\langle EOS \rangle$ or when the number of words generated has reached its maximum.

$$\text{output }, (h_i, c_i) = \text{Dec}_{Clf(z)}\left(e\left(\bar{X}_{i-1}\right), (h_{i-1}, c_{i-1})\right) \tag{14}$$

4 Experiments

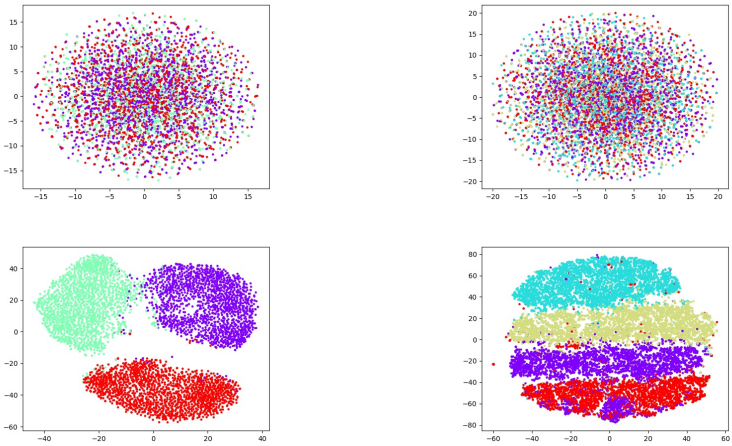
4.1 Dataset

The dataset we use is the Daily Dialog Dataset [40], a multi-round conversation dataset with labels containing high-quality human conversations about everyday life. We split the multi-round conversation into a single round. The dataset has three tags containing 4 categories of action, 7 categories of emotion and 10 categories of themes. We mainly use action labels (inform, question, directive, commissive) and emotion labels (no emotion, anger, disgust, fear, happiness,

sadness, surprice). Due to the uneven amount of data across categories in the dataset, especially the emotion tag where no emotion accounts for more than 80% of the data. We marked all four categories of anger, disgust, fear and sadness as negative, two categories of happiness and surprise as positive, and no emotion as unchanged. The original dataset was then reconstructed into a sub-dataset containing three labels as the dataset for our experiments. As shown in Table 1.

Table 1: Distribution of Daily Dialog dataset after processing.

action	inform	question	directive	commissive
trainset	26754	20313	11108	4720
validset	5733	4353	2380	1011
testset	5733	4353	2380	1011
emotion	no emotion	positive	negative	-
trainset	10000	1000	1800	-
validset	2004	496	200	-
testset	2000	460	303	-



(a) Latent space distribution of emotion

(b) Latent space distribution of action

Fig. 2: Visualization results of the latent space distribution. The top image in Fig. 2 demonstrates the entangled distribution in the latent space for the dataset, and the disentangled distribution after using S-CVAE is shown below.

4.2 Automatic Evaluation

We use three evaluation metrics based on word overlap to evaluate the quality of text generation. One metric evaluates the diversity of the generated text. One metric evaluates the effectiveness of the classifier.

BLEU [41] calculates the frequency of co-occurring words in both sentences, i.e. by counting the number of occurrences of n-gram phrases in the generated and real responses across the training corpus.

ROUGE [42] also assesses text quality by calculating co-occurrence frequency, although unlike BLEU the words in ROUGE can be non-consecutive, whereas BLEU's n-gram requires words to occur consecutively.

METEOR [43] is based on BLEU with some improvements to include an alignment relationship between the generated and real responses. It also uses WordNet to compute sequence-specific matches, synonyms, root and affixes, and paraphrase matches. METEOR improves the effectiveness of BLEU and makes it more relevant to manual discriminations.

Distinct-2 [44] determines whether a large number of generic, repetitive responses occur and is used to evaluate the diversity of text generation.

ACC evaluates how well the classifier classifies results and is calculated by dividing the number of correctly classified samples by the total number of samples.

4.3 Benchmark Methods

We used the same dataset on the dialogue generation task and the following baseline for comparison.

seq2seq [2], using deep neural networks, and converts input sequences into output sequences and optimises the structure by incorporating Global Attention [45], allowing the model to pay attention to more information.

Transformer [4], like seq2seq, uses the same encoder and decoder mechanism. But uses the structure of attention instead of lstm. self-attention is used throughout for encoding and decoding.

Vanilla VAE [6], using continuous latent spaces and a priori normal distributions. The text is encoded as a probability distribution in the latent space, rather than as a deterministic vector. VAE allows better modelling of text diversity and achieves category controllability in text generation.

Vanilla CVAE [38], which introduces conditional variables on top of the VAE. The prior distribution also changes from a fixed standard normal distribution to a normal distribution with parameters fitted by a priori networks. CVAE is more suitable for dialogue generation tasks.

4.4 Implementation Details

The words in the dataset were processed to produce a word list dictionary containing 6918 words. The dictionary contains four special tokens, $\langle PAD \rangle$, $\langle SOS \rangle$, $\langle EOS \rangle$ and $\langle UNK \rangle$. $\langle PAD \rangle$ denotes padding values, $\langle SOS \rangle$ denotes the start of sentence, $\langle EOS \rangle$ denotes the end of sentence and $\langle UNK \rangle$ denotes

Table 2: The examples of dialogue generate.

Example 1	
post	Ohhhh. I hate this part of my job.
response	Why don't you go over the resumes again? They might help you decide who to hire.
emotion	negative
no emotion	But the the condition it was really good. I though I want to buy a house at my school.
positive	I drink when you got through my job, and I couldn't get going to tell you how much to he save at the place!
negative	**But what else are you going to do?**
pretrain	Why are you feeling depressed?
Example 2	
post	Is there any other way I can reach him?
response	I'm afraid not , he has gone out of this town on business. May I take a message?
action	question
inform	overweight to you...
question	**Of course you would call.**
directive	You'll let you know if you don't get off during me.
commissive	No. Thirty minutes.
pretrain	Certainly.

words that do not appear in the word list. The word vector has dimension 300. The encoder uses a bi-directional LSTM with a hidden layer size of 300 and 2 layers. The decoder uses a one-way LSTM with a hidden layer size of 300 and 2 layers. The latent variable dimension is 200, and the KL annealing algorithm is used to avoid the KL posterior collapse problem.

4.5 Results and Analysis

4.5.1 Disentangled result

Fig. 2 shows visualization results of the latent space distribution. From Fig. 2(a) we can see that the original latent space is mixed together, with data from different emotion labels encoded into a single circle. In other words, when we sample from the latent space, the points sampled after encoding the conversations with different emotions into the latent space are not differentiated. In contrast, S-CVAE is trained to disentangle the latent space, with different coloured points representing different emotion conversations. It is clear that conversations of the same type are encoded in one region. Points of the same type are closer to each other, while points of different types are further away from each other. Disentangled latent space enhance the interpretability of our generated dialogues, and we were able to generate more interpretable dialogue

text. Experimentally, our optimised model is able to train a decoupled latent space, making the text generation process interpretable, compared to the deep hidden variable models Vanilla VAE and Vanilla CVAE. The same results can be seen in Fig. 2(b) for the action label data, indicating that S-CVAE is generalisable.

Table 3: The automatic evaluation results of emotion. No emotion, positive, negative means that all dialogues use a decoder of a certain category. pretrain means that the decoder trained during pre-training is used to generate dialogues. our mode means that different decoders are chosen to generate dialogues based on the results of the classifier. The rest is the baseline model we chose.

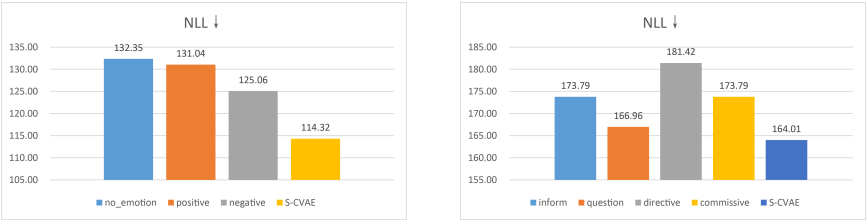
Method	BLEU	METEOR	ROUGE	dist-2	ACC
S-CVAE	**12.63±0.33**	**13.08±0.35**	30.47±0.30	0.32	0.69
no emotion	10.64±0.24	10.64±0.25	28.64±0.24	0.29	-
positive	10.77±0.25	11.17±0.27	29.46±0.25	0.32	
negative	8.66±0.19	9.33±0.19	27.68±0.23	0.26	
pretrain	10.52±0.20	11.07±0.22	28.61±0.22	0.22	
Seq2seq	11.15±0.22	11.89±0.26	30.99±0.25	0.30	-
Transformer	10.44±0.19	11.03±0.21	**31.59±0.25**	**0.42**	
Vanilla VAE	10.01±0.17	10.48±0.19	28.09±0.21	0.18	
Vanilla CVAE	11.17±0.22	11.50±0.25	29.46±0.23	0.25	

Table 4: The automatic evaluation results of action. Unlike Table 3, we use a DD dataset with action labels to train the model.

Method	BLEU	METEOR	ROUGE	dist-2	ACC
S-CVAE	**13.22±0.15**	**14.03±0.17**	**33.12±0.14**	0.18	0.76
inform	11.42±0.12	12.06±0.13	31.93±0.12	0.18	-
question	10.82±0.11	11.29±0.11	31.35±0.11	0.18	
directive	9.91±0.08	10.18±0.09	30.09±0.10	0.15	
commissive	10.05±0.08	10.51±0.08	30.94±0.10	0.13	
pretrain	11.07±0.10	11.56±0.11	30.66±0.10	0.14	
Seq2seq	10.14±0.08	10.52±0.09	30.15±0.11	0.17	-
Transformer	10.11±0.08	11.16±0.09	33.09±0.11	**0.25**	
Vanilla VAE	10.85±0.10	11.52±0.11	30.73±0.11	0.14	
Vanilla CVAE	11.50±0.11	11.95±0.12	31.12±0.11	0.14	

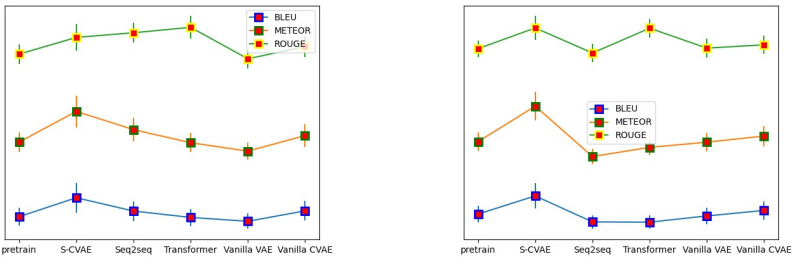
4.5.2 Automatic evaluation result

Table 2 shows some examples generated using S-CVAE. Post is the input question to the model, and response is the ground-truth response. The rest are the



(a) NLL result of emotion (b) NLL result of action

Fig. 3: Negative log likelihood (NLL) results for the test set. The result allows the effectiveness of the model in reconstructing the dialogue to be assessed. And the smaller the value of NLL the better the reconstruction.



(a) Error bar result of emotion (b) Error bar of action

Fig. 4: Error bar results for metrics with different generation models.

responses generated using different semantic decoders. We can see that different decoders can generate responses with different sentiments. Decoders that use controlled variables to select generate dialogue are more logical. However, other decoders would generate dialogue that is out of context. Table 3 and table 4 show the results of training on the S-CVAE and the four baseline models using the DD dataset. We can see that the quality of the text generated by the decoder selected by the results of the classifier is better than that generated by using only a single type of decoder or by using a pre-trained decoder. The result suggests that the emotion and the action of the dialogue have some influence on the text generation. In all three metrics, BLEU, METEOR, and ROUGE, which are based on word overlap, S-CVAE compares favourably with the baseline model. Therefore, S-CVAE can generate higher-quality conversations and S-CVAE also outperforms most of the models on dist-2, indicating that S-CVAE can generate a diversity of dialogues. Experiments show that S-CVAE can improve the performance of dialogue responses when it correctly identifies both the emotions and actions of the dialogue. S-CVAE achieves better results on both the emotion dataset and the action dataset, which indicates

that S-CVAE is capable of generalization.
Fig. 3 shows the negative log likelihood values calculated for the test set on both S-CVAE and the but semantic decoder. A lower NLL value means a better reconstruction. In Fig. 3(a) you can see that S-CVAE has the lowest result, indicating that it has the best reconstruction on the test set and is stronger than using a single semantic decoder. The same result can be seen in Fig. 3(b). Better reconstruction results indicate better generative power.

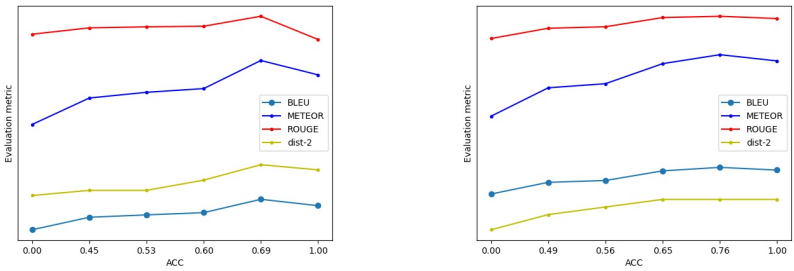
Table 5: The performance of the model under emotion labels in different classifier accuracies.

Metric / ACC	BLEU	METEOR	ROUGE	dist-2
0.00	9.66±0.18	9.96±0.19	28.72±0.22	0.26
0.45	10.88±0.26	11.25±0.28	29.34±0.33	0.27
0.53	11.11±0.26	11.53±0.28	29.44±0.26	0.27
0.60	11.33±0.27	11.71±0.29	29.50±0.27	0.29
0.69	**12.63±0.33**	**13.08±0.35**	**30.47±0.30**	**0.32**
1.00	12.01±0.30	12.38±0.33	28.81±0.29	0.31

Table 6: The performance of the model under action labels in different classifier accuracies.

Metric / ACC	BLEU	METEOR	ROUGE	dist-2
0.00	9.71±0.08	9.98±0.08	30.19±0.10	0.10
0.49	11.25±0.11	11.85±0.12	31.54±0.11	0.14
0.56	11.49±0.12	12.12±0.13	31.74±0.12	0.16
0.65	12.76±0.13	13.44±0.15	32.95±0.13	0.18
0.76	**13.22±0.15**	**14.03±0.17**	**33.12±0.14**	**0.18**
1.00	12.86±0.15	13.62±0.16	32.82±0.14	0.18

We investigated the performance of the S-CVAE under different classifier accuracies and Table 5 and Table 6 show the generation effect of the model at different classification accuracies. The results of the tables show that when we use the higher accuracy of the classifier, the better performance can be obtained. As the classifier accuracy falls, the model's generation effectiveness decreases. Besides, the worst quality of text generation occurring when the classifier accuracy is 0. This is due to the selection of the wrong class of decoder to generate the responses in the generation phase. However, when the classifier accuracy is 1, the best results are not achieved as expected. This phenomenon is because that the performance of the model is saturated even when the classifier accuracy is 1. As the classifier and encoder are trained at the same time, we discarded some of the classifier capability in order to guarantee the encoder



(a) Evaluation metrics line chart of emotion (b) Evaluation metrics line chart of action

Fig. 5: Evaluation metrics line chart of the model's performance in different classifier accuracies.

capability. If we can improve the accuracy of the classifier with guaranteed encoder power, then there will be some improvement to the text generation results. From Fig. 5(a) we can see that the quality of the text generated by the model improves the fastest when the classifier accuracy is between 0.60 and 0.69. After the accuracy reaches a certain critical value, the model generation effect will decrease again. If we can find this critical value, then we can get a better model. In the future work, we will try to improve the effectiveness of the generative model by increasing the accuracy of the classifier.

4.5.3 Semantic result

(a) ACC result of emotion (b) ACC result of action

Fig. 6: Semantic results of the generated response.

We have trained a classifier in the pre-training phase. Now we use pre-trained classifier to classify the generated response, using accuracy to represent the classification effect. The expectation is that the classification task will verify that the model captures the corresponding semantics and generates dialogues with the corresponding semantics. Fig. 6 shows the semantic result of the generated text. Fig. 6(a) shows the classification results for the dataset using the emotion labels. The radar chart shows that the text generated by

our model S-CVAE performs best in the classification task. *Pretrain* indicates that a generic decoder was used to generate the text, the generated text is inferior to S-CVAE in the classification task. The results of using single-semantic decoders such as *no_emotion* and *positive* fluctuate, but they are also inferior to S-CVAE. The result indicates that our model can generate dialogue with the corresponding semantics. The classification results for the dataset using action labels are shown in Fig. 6(b).

The results of the semantic experiments demonstrate that S-CVAE generates semantically more accurate conversational text compared to a single-semantic decoder. However, the classification results of the generated text were not particularly improved. In the pre-training phase, we use the text generation effect as the main optimization metric and sacrificed the performance of the classifier to ensure that the ability of the encoder for encoding. Therefore, the accuracy of the classifier was decreased, which led to biased classification results of the classifier for the generated text. In future work, we will try to improve the accuracy of the classifier while ensuring the encoder effect.

5 Conclusions

In this paper, we propose a semantic-aware conditional variational autoencoder (S-CVAE) for one-to-many dialogue generation to solve the problem regarding miscellaneous ambiguity of semantics. Specifically, our model leverages explicitly controllable variables to distinguish sentiment and action responses for creating diverse conversational texts. Meanwhile, controllable variables are capable of disentangling the latent features to investigate the impact of different categories of latent space features on the logic of text generation. Experimental results validate S-CVAE for text generation in terms of quality, diversity, and interpretability.

6 Acknowledgement

This work was partly supported by National Key R&D Program of China (2019YFB2103000), the National Natural Science Foundation of China (62136002,62102057 and 61876027), the Science and Technology Research Program of Chongqing Municipal Education Commission (KJQN202100627 and KJQN202100629), and the National Natural Science Foundation of Chongqing (cstc2019jcyj-cxttX0002), respectively.

7 Compliance with ethical standards

Conflict of interest We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

References

[1] Turing, A.M.: Computing machinery and intelligence. In: The Philosophy of Artificial Intelligence, pp. 40–66 (1990)

[2] Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, pp. 3104–3112 (2014)

[3] Yao, K., Zhang, L., Luo, T., Du, D., Wu, Y.: Non-deterministic and emotional chatting machine: learning emotional conversation generation using conditional variational autoencoders. Neural Computing and Applications **33**(11), 5581–5589 (2021)

[4] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, pp. 5998–6008 (2017)

[5] Potamias, R.A., Siolas, G., Stafylopatis, A.-G.: A transformer-based approach to irony and sarcasm detection. Neural Computing and Applications **32**(23), 17309–17320 (2020)

[6] Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: 2nd International Conference on Learning Representations (2014)

[7] Chen, M.-Y., Chiang, H.-S., Sangaiah, A.K., Hsieh, T.-C.: Recurrent neural network with attention mechanism for language model. Neural Computing and Applications **32**(12), 7915–7923 (2020)

[8] Weizenbaum, J.: Eliza—a computer program for the study of natural language communication between man and machine. Communications of the ACM **9**(1), 36–45 (1966)

[9] Young, S., Gašić, M., Thomson, B., Williams, J.D.: Pomdp-based statistical spoken dialog systems: A review. Proceedings of the IEEE **101**(5), 1160–1179 (2013)

[10] Vinyals, O., Le, Q.: A neural conversational model. Computer Science (2015)

[11] Serban, I., Sordoni, A., Bengio, Y., Courville, A., Pineau, J.: Building end-to-end dialogue systems using generative hierarchical neural network models. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 30 (2016)

[12] Wang, Y., Wang, H., Zhang, X., Chaspari, T., Choe, Y., Lu, M.: An attention-aware bidirectional multi-residual recurrent neural network (abmrnn): A study about better short-term text classification. In: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3582–3586 (2019). IEEE

[13] Li, J., Galley, M., Brockett, C., Gao, J., Dolan, B.: A diversity-promoting objective function for neural conversation models. Computer Science (2015)

[14] Galetzka, F., Rose, J., Schlangen, D., Lehmann, J.: Space efficient context encoding for non-task-oriented dialogue generation with graph attention transformer. 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), pp. 7028–7041 (2021)

[15] Ribeiro M T, G.C. Singh S: "why should i trust you?" explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1135–1144 (2016)

[16] Lundberg S M, L.S.I.: A unified approach to interpreting model predictions. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, pp. 4768–4777 (2017)

[17] Zhang, Q., Yang, Y., Ma, H., Wu, Y.N.: Interpreting cnns via decision trees. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6261–6270 (2019)

[18] Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., Abbeel, P.: Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In: Proceedings of the 30th International Conference on Neural Information Processing Systems, pp. 2180–2188 (2016)

[19] Karras T, A.T. Laine S: A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4401–4410 (2019)

[20] Chen, H., Liu, X., Yin, D., Tang, J.: A survey on dialogue systems: Recent advances and new frontiers. Acm Sigkdd Explorations Newsletter **19**(2), 25–35 (2017)

[21] Li, J., Monroe, W., Ritter, A., Galley, M., Gao, J., Jurafsky, D.: Deep reinforcement learning for dialogue generation. In: Proc. of EMNLP (2016)

[22] Serban, I., Sordoni, A., Lowe, R., Charlin, L., Pineau, J., Courville, A., Bengio, Y.: A hierarchical latent variable encoder-decoder model for generating dialogues. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31 (2017)

[23] Shang, L., Lu, Z., Hang, L.: Neural responding machine for short-text conversation. IEEE (2015)

[24] Cho, K., Merrienboer, B.V., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. Computer Science (2014)

[25] Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., Lerchner, A.: beta-vae: Learning basic visual concepts with a constrained variational framework. In: 5th International Conference on Learning Representations, pp. 4401–4410 (2019)

[26] Hu, Z., Yang, Z., Liang, X., Salakhutdinov, R., Xing, E.P.: Toward controlled generation of text. In: 34th International Conference on Machine Learning, pp. 1587–1596 (2017)

[27] Wiseman, S., Shieber, S., Rush, A.: Learning neural templates for text generation. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 3174–3187. Association for Computational Linguistics, Brussels, Belgium (2018)

[28] Zhao T, E.M. Lee K: Unsupervised discrete sentence representation learning for interpretable neural dialog generation. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, pp. 1098–1107 (2018)

[29] See, A., Roller, S., Kiela, D., Weston, J.: What makes a good conversation? how controllable attributes affect human judgments. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 1702–1723. Association for Computational Linguistics, Minneapolis, Minnesota (2019)

[30] Ficler, J., Goldberg, Y.: Controlling linguistic style aspects in neural language generation. In: Proceedings of the Workshop on Stylistic Variation, pp. 94–104. Association for Computational Linguistics, Copenhagen, Denmark (2017)

[31] Li, Z., Jiang, X., Shang, L., Liu, Q.: Decomposable neural paraphrase generation. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 3403–3414. Association for Computational

Linguistics, Florence, Italy (2019)

[32] Sato, M., Suzuki, J., Shindo, H., Matsumoto, Y.: Interpretable adversarial perturbation in input embedding space for text. In: the 27th International Joint Conference on Artificial Intelligence and the 23rd European Conference on Artificial Intelligence (2018)

[33] Pang, B., Wu, Y.N.: Latent space energy-based model of symbol-vector coupling for text generation and classification. In: Proceedings of the 38th International Conference on Machine Learning, vol. 139, pp. 8359–8370 (2021)

[34] Shi, W., Zhou, H., Miao, N., Li, L.: Dispersed exponential family mixture vaes for interpretable text generation. In: Proceedings of the 37th International Conference on Machine Learning, vol. 119, pp. 8840–8851 (2020)

[35] Chen, C., Peng, J., Wang, F., Xu, J., Wu, H.: Generating multiple diverse responses with multi-mapping and posterior mapping selection. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019, pp. 4918–4924 (2019)

[36] Bao, S., He, H., Wang, F., Wu, H., Wang, H.: PLATO: pre-trained dialogue generation model with discrete latent variable. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020, pp. 85–96 (2020)

[37] Cui, Z., Li, Y., Zhang, J., Cui, J., Wei, C., Wang, B.: Focus-constrained attention mechanism for cvae-based response generation. In: Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020, pp. 2021–2030 (2020)

[38] Sohn, K., Lee, H., Yan, X.: Learning structured output representation using deep conditional generative models. In: Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, pp. 3483–3491 (2015)

[39] Wang, Y., Zhang, X., Lu, M., Wang, H., Choe, Y.: Attention augmentation with multi-residual in bidirectional lstm. Neurocomputing **385**, 340–347 (2020)

[40] Li, Y., Su, H., Shen, X., Li, W., Cao, Z., Niu, S.: Dailydialog: A manually labelled multi-turn dialogue dataset. In: Proceedings of the Eighth International Joint Conference on Natural Language Processing, pp. 986–995 (2017)

[41] Papineni, K., Roukos, S., Ward, T., Zhu, W.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pp. 311–318 (2002)

[42] Lin, C.-Y.: Rouge: A package for automatic evaluation of summaries. In: Text Summarization Branches Out, pp. 74–81 (2004)

[43] Banerjee, S., Lavie, A.: METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In: Proceedings of the Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation And/or Summarization@ACL 2005, pp. 65–72 (2005)

[44] Li, J., Galley, M., Brockett, C., Gao, J., Dolan, B.: A diversity-promoting objective function for neural conversation models. In: The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 110–119 (2016)

[45] Luong, T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1412–1421 (2015)