DialogDiffAE: Dialogue Generation with Diffusion-Equipped Auto-Encoder

Stanford CS224N {Custom} Project

Fangzhao Zhang
Department of Electrical Engineering
Stanford University
zfzhao@stanford.edu

Xiaohan Song
Department of Geophysics
Stanford University
xhsong@stanford.edu

Abstract

We studied the application of diffusion model in helping auto-encoder better capture the latent space distribution in dialogue generation tasks. Specifically, a model including both an auto-encoder and a diffuser is constructed, and a 3-step training procedure is proposed to adapt the diffusion model in tackling conditional dialogue generation tasks. The responses generated by DialogDiffAE are shown to be more stable and concise responses compared to baseline methods. Such results demonstrate the capability of diffusion model in capturing potentially complex textual latent space distribution.

- Mentor: Yuan Gao

1 Introduction

Variational autoencoder (VAE) [1] is a generative model which has long been widely applied in broader areas, including the natural language processing [2, 3, 4]. VAE [1] encodes data into a latent space and then decodes for the output generation. By training the encoder-decoder networks and approximating the latent variables by a normal distribution meanwhile, it enables the model for generative sampling. However, no matter how one modifies the structures of encoder and decoder, a significant limitation of VAE lies in its simple assumption that word embedding latent variable distribution is close to standard Gaussian. While the true latent space might be much more complex and diverse, VAE model suffers from posterior collapse problem [5]. Prior work such as DialogWAE[6] tries to solve this problem by employing a generative adversarial network (GAN) to model the potentially complex textual latent space, and has achieved superior performance in conditional dialogue generation task. Though the latent space distribution is no longer required to be close to Gaussian for DialogWAE to take effect, the introduction of GAN brings new drawbacks such as the instability introduced by GAN's inherent adversarial training procedure. The problem remains to find a better method to tackle the potential complexity of the autoencoder latent variable distribution, and in this case, diffusion model [7], with its outstanding performance in learning unknown distributions, comes as a remedy.

In this work, we studied the application of diffusion model in aiding auto-encoder in dialogue generation tasks. Specifically, we exploit a diffusion model to learn the potentially complex textual latent space distribution, which is trained together with the classic auto-encoder models. Our simulation results shows that diffusion-equipped auto-encoder can generate more stable and concise responses compared to baseline methods such as DialogWAE. This demonstrates the capability of diffusion model in capturing potentially complex textual latent space distribution.

2 Related Work

Diffusion model [7], as a newly emerged generative model, has gone through its dramatic growth during the past two years and is widely applied in conditional and unconditional generative tasks, especially in computer vision realm. Unlike traditional generative models, diffusion model is inspired by the Markov process in thermodynamics and uses an autoencoder network to learn the latent distributions in every backward Markov step to generate new outputs [7]. Diffusion model used to be incapable of generating high quality images, but after introducing certain pasteurization techniques [7] and using the non-markovian forward processes[8], the training process of diffusion models are largely expedited, which makes diffusion models performing increasingly excellent in generating high quality images. However, diffusion models' advantage is in learning continuous data distribution, which cannot be used directly to process discrete nature language data. Many attempts have been conducted to make diffusion models useful for nature language processing tasks, either by transferring the discrete word/sentence embeddings to a continuous domain [9, 10], or developing new diffusion models for discrete word embeddings [11]. Diffusion-LM and Diffuseq models [9, 10] use a map function to project discrete words into continuous embeddings and has achieved equal or even higher performance than traditional language models in many sequence to sequence tasks, especially in text diversity scores. However, its superior achievements tends to restrict in more intuitive tasks like paraphrasing and simplification [10]. DiffusionBERT [11], instead, relies on discrete diffusion models and gets a better performance than existing continuous diffusion models on uncontrolled generative tasks. Such result shows a better adaptation of discrete diffusion method in processing discrete text embeddings. Thereby, a promising way to introduce diffusion models into language processing must be finding a task within the process that the diffusion model may perform better than other algorithms.

Variational autoencoder (VAE) is a commonly used tool in text/image generation tasks. It was first intruduced in [1], and is used in text generation tasks including but not limited to [12, 13, 14].An intuitive example of applying VAE into generative language model is by setting the encoder and decoder networks to be both LSTMs, but it turns out to perform not as good as simple LSTMs, until redesigning the decoder with a dilated CNN [2]. As BERT and transformers become the trend in NLP field, new attempts of combining VAE with different types of state-of-the-art methods are subsequently conducted [3, 4]. The generation capability of VAE comes from that it penalizes the embedded latent space distribution and a simple known distribution one can sample from, i.e., standard Gaussian distribution. However, this only works well when the latent space distribution is indeed close to the simple known distribution, which is usually false because latent space distribution can be arbitrarily complex in most tasks. Thus, to alleviate this drawback of VAE, tools for learning arbitrary distribution from a simple known distribution has been introduced to learn the latent space distribution. With this trick, one can model the latent space distribution much better even if the distribution is complex and hard to sample from. Different tools such as GAN ([15]) has been used to model such latent space distribution, see [6].

With the recent advancement in generative networks, it has been shown that diffusion model is capable of modeling complex data distribution by sampling from random white noise and carrying out a denoising procedure. Diffusion model has achieved great success in image generation tasks, see [7]. Given diffusion model's great ability to model complex distribution from white noise, our work focuses on studying the use of diffusion model in capturing latent space distribution in text generation tasks.

3 Approach

3.1 Model Structure

Inspired by DialogWAE model[6], we introduced double encoder lines for context-response pairs and context-only processings respectively. Our DialogDiffAE model consists of RNN language encoders & decoders, AE-like downsizing and up-sizing encoders (latent encoders & decoders), and a diffusion network, shown in Fig. 1.

The response and context sentences will first be processed with the RNN encoders which maps the discrete words into continuous domain. After going through the latent encoders & diffuser & latent decoder, an RNN response decoder is then applied to transform the continuous embeddings back to

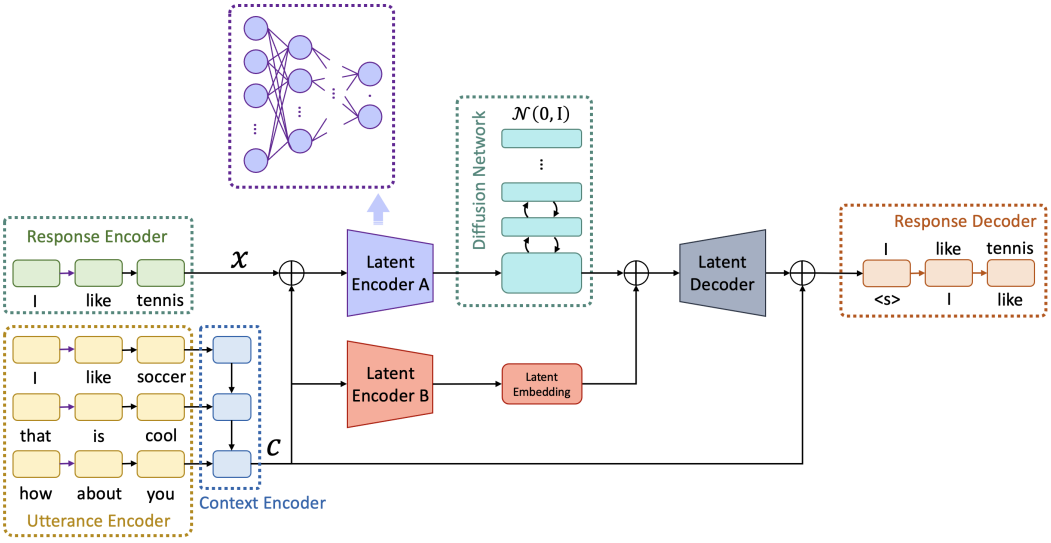

Figure 1: DialogDiffAE model structure

language sentences. Here we use the same encoder and decoder structures applied in DialogWAE [6] model, which uses gated recurrent units (GRU) [16] as the RNN elements and the glove vector model (twitter pre-trained) [17] is used as the word-vector embeddings.

Latent encoders and decoders are applied to downsize the embeddings into a lower-dimensional latent space. They are consist of fully connected linear layers with LeakyReLU activations. Latent encoders A and B process the context-response pairs and context-only information separately, in which A is expected to extract the basic dialog features embedded in the context-response pairs that feed into the diffusion model, while B is expected to transform the context information into a suitable latent space for diffuser output to be concatenated with. The latent decoder processes the concatenated information which includes a combination of diffuser generated "dialog rules" and encoder processed context information. This output will form a concatenation with the pre-downsized context embeddings for the final RNN decoding.

The Diffusion model applied in our network is the DDIM model [8]. This network propagates the input (context-response pairs) towards the uniform normal distribution $\mathcal{N}(0, I)$. This diffusion network is expected to learn and generate the generalized dialogue rules from the processed context-response pairs from encoder A.

3.2 Diffusion Model

Follow the classic diffusion model setup, assume data is distributed according to $x_0 \sim q(x_0)$. Consider the procedure of gradually diffusing the data to obtain latent variables $x_1, \ldots, x_T$, where $x_t \sim \mathcal{N}(\sqrt{1-\beta_t}x_{t-1}, \beta_t I)$ ($\{\beta_t\}$ is a sequence of hyperparameters increasing from 0 to 1, and is referred to as noise scheduling.) The stochastic process from x_0 to x_T forms a Markov process that is referred as the forward procedure.

To generate data from pure noise, we need to reverse the above forward procedure and form the backward procedure that allows us to get to x_0 from x_T. In [7], the authors models the backward process again as Markovnian, and employs the transition probability

$$p_\theta(x_{t-1}|x_t) \sim \mathcal{N}(\mu_\theta(x_t, t), \sigma_t^2 I)$$

where $\sigma_t^2 = \beta_t$ and μ_θ is chosen with reparametrization trick as

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\varepsilon_\theta(x_t, t)\right)$$

where $\alpha_t = 1 - \beta_t$ and $\overline{\alpha_t} = \Pi_{s=1}^{t} \alpha_s$. With this specific modeling of backward process, the training procedure is simplified to penalize square loss

$$\mathbb{E}_{t\sim U(1,2,...,T),\varepsilon\sim\mathcal{N}(0,I),x_0\sim q}\|\varepsilon - \varepsilon_\theta(\sqrt{\overline{\alpha}_t}x_0 + \sqrt{1-\overline{\alpha}_t}\varepsilon, t)\|^2 \tag{1}$$

The sampling process first samples $x_T \sim \mathcal{N}(0, I)$, and then from $t = T$ to $t = 1$ iteratively.

$$x_{t-1} = \frac{1}{\alpha_t}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\overline{\alpha}_t}}\varepsilon_\theta(x_t, t)\right) + \sigma_t z$$

where $z \sim \mathcal{N}(0, I)$. Note the model is connected to score matching since the backward process resembles Langevin dynamics with ε_θ as a learned score function. See [18] for a deeper discussion of connection between DDPM model and score-based model.

In our model, the diffusion model is applied to learn the distribution of output of latent encoder A, and is in aid for latent encoder B and latent decoder to learn the conditional probability $\mathbb{P}(x|c)$.

3.3 Training and Generation

To achieve efficient training which allows different networks to learn different tasks, we separate a full training epoch into three steps as listed below, where (c, x) represents a pair of context-response data in the training set:

1. AE Training Phase 1: Train the latent encoder A for the context-response pairs only, while fixing all the other networks. This step aims to let the encoder learn extracting general dialog rules for diffusion model to train on. This time we are using the same loss function as in step 1, with one modification of replacing diffuser output with encoder A's output:
$$\mathcal{L}_1 = E_{(x,c)\in\text{training set}} \log \mathbb{P}_\theta(x|c, A(x, c))$$
2. Diffusion Training: Train the diffusion network only, and fixing the other networks. This step let the diffuser learn generating sample "dialog rules" produced by the latent encoder A. Here we use the same loss function in equation (1).
3. AE Training Phase 2: With randomly sampled diffusion output, fixing the diffusion network and update all the networks remained (Latent encoder B, Latent decoder, and RNN encoder & decoders). This step helps training the encoder in processing context-only embeddings and at the same time training the decoder in getting outputs without the knowledge of response. This time, the loss function used for the gradient descent is (note that $\mathbb{P}_\theta$ is the RNN decoder output):
$$\mathcal{L}_1 = E_{(x,c)\in\text{training set}} E_{\varepsilon\sim\mathcal{N}(0,\text{I})} \log \mathbb{P}_\theta(x|c, \text{Diffuser}(\varepsilon))$$

In the generation procedure, we generate random sample from distribution $\mathcal{N}(0, \boldsymbol{I})$ and run the backward denoising procedure with learned diffusion model. The concatenation of diffuser output and the encoder-B-processed context embeddings will then be passed to the latent decoder and the latent decoder output will then be combined with the raw context to form the input of the RNN response decoder, the output of which will be our final output sentence.

4 Experiments

4.1 Data

We use DailyDial [19], a dataset containing over 10 thousands examples of english daily dialogue sequences, and split it to 11,118 examples & 200 examples for the training and test of our dialogue generation model. We use GLOVE as our word to vec embeddings (glove.twitter.27B.200d.txt [20]). We borrowed code from DialogWAE [6] and used the context decoder and response decoder parts of it. We made modifications to it for the construction of our own DiaalogDiffAE model.

4.2 Evaluation method

To evaluate our generation results, we introduce the BLEU score [21], the BOW Embedding [22], and the distinct scores [23] as our metrics, which are also used in DialogWAE's evaluation [6]. BLEU

score evaluates the overlapping extent between the generated and reference responses. BOW measures the cosine similarity of the word embeddings between the generated and reference responses. Dist evaluates the response diversity, in which intra-dist evaluates the diversity in a single response and inter-dist evaluate the diversity for different response samples under the same context.

4.3 Experimental details and results

In our model, We set the diffusion network to be 10 layer u-net with 200 sample steps for every sampling. We trained our network with 80 global epochs, and in each global epoch we perform 10 times of training phase 1, 10 times of diffusion training, and 1 time of training phase 2 (training phases described in section 3). We evaluate our models on the test set with each test example sampling 3 responses for the calculation of precise and recall metrics. The change of loss functions on the training set, the evaluation matrices, and the output examples on the testing sets with regard to different epochs in the training processes are shown in Fig. 2, Fig. 3, and Table. 1, respectively.

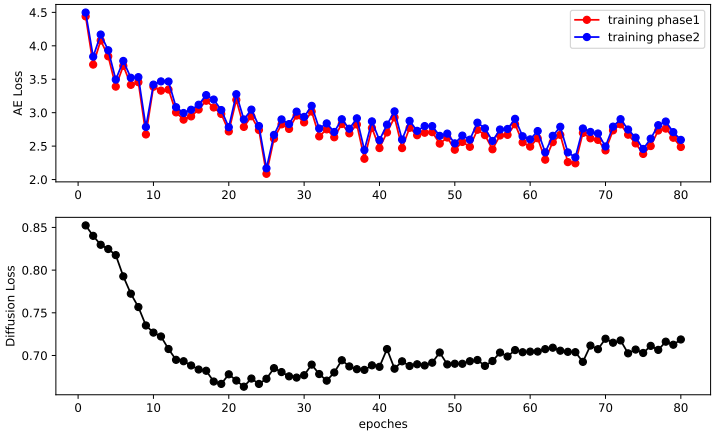

Figure 2: Loss function changes in a training process

Table 1: Response examples from our DialogDiffAE model after different training epochs.

Context & Reference Response	Response (epoch 5)
how much is it ? this one sells for $ 39. 99 . can i test it out ? of course . it sounds great . i ' ll take it .	- it's a bit of the <unk> . that's right . i'm sorry . i'll take it . yes , it's a bit of the <unk> . i'm sorry . i'll take it .
Response (epoch 40)	**Response (epoch 80)**
- $ 200 per month . that's very good . i'll take it . yes , you can . but you'll have to ... is it the only thing i've got to pay for it ?	- $ 200 . 50 . that's a lot of money . sure , you can have a few minutes . i ... i'll have them for my own .

During the training process, the diffusion loss decreases to its bottom after only 20 epochs while the AE losses continuously decrease until about epoch 50. However, although the metrics may show that the model is working good enough after 40 epochs of training, human check of the actual outputs suggest that the later model do perform better than the previous ones in generating logically accepted responses.

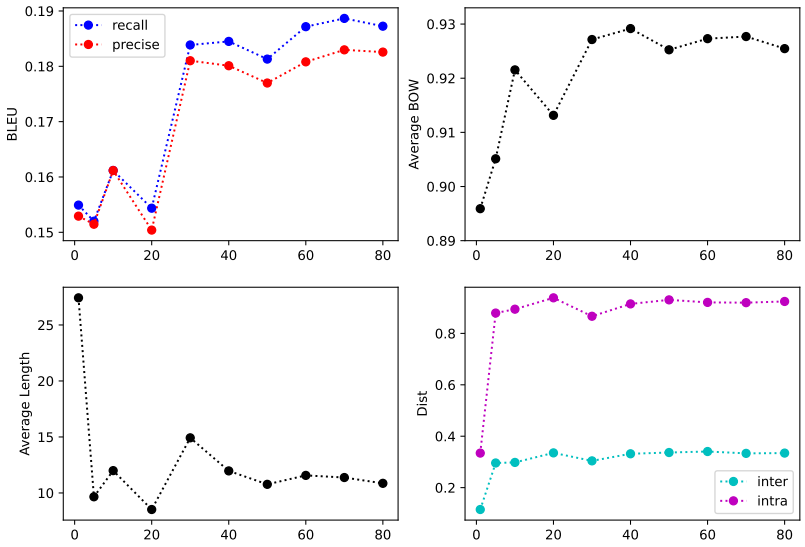

Figure 3: BLEU, BOW, response length, and distinct score changes with regard to different epochs

We then compare the result, including the metrics scores and some response examples generated by the 80-epoch DialogDiffAE model with the baseline, DialogWAE model [6], with the results shown in Table. 2 and Table. 3.

Table 2: Metrics comparison between DialogWAE [6] and our DialogDiffAE.

Model	BLEU recall	BLEU precise	BOW	Average Length*	Inter-dist	Intra-dist
DialogWAE	**0.195224**	0.157977	**0.941461**	22.681210	**0.519404**	0.744020
DialogDiffAE	0.187251	**0.182571**	0.925465	**10.860384**	0.334369	**0.924270**

*note that because we noticed that the average lengths of the DailyDialog sentences seldom exceed 20, we would prefer smaller response lengths.

As shown in Table. 2 and Table. 3, the responses generated by DialogDiffAE are more stable and concise, while the responses generated by DialogWAE are longer and have more diversity. Such differences result in the higher precise BLEU scores, lower recall BLEU scores, and lower inter-dist scores for our DialogDiffAE compared to the scores of DialogWAE. When checking the exact examples (Table. 3), we noticed that DialogDiffAE's output fits the logical sequence better based on human evaluation, although the output may have no phrase or meaning overlap with the reference responses.

5 Analysis

Although our DialogDiffAE has shown advantages in giving logically expected responses, one limitation of this model is the identical responses it generates for most of the test examples under different random diffusion samples (see example 2). Such phenomenon means either the diffusion model always somehow generates the similar response strategy, or the diffusion model didn't really learn the general language relation between the context and responses, so that the decoder downplays the weight assigned for the diffusion output. We propose that the second case has more probability.

Table 3: Comparison of example responses between the DialogWAE [6] and our DialogDiffAE

Responses	Example1	Example2
Context	A: oh , it ' s almost eleven twenty . B: thank you . A: you ' re welcome . the rain is quite heavy , isn ' t it ? B: yeah . i was in a hurry and i forgot my umbrella today . A: i have one . we could share it . which way are you going ? B: (to be generated)	A: hello ? B: hi , mr . smith . this is mary . is jenny there ? A: no , i ' m afraid not , mary . may i take a message for her ? B: yes , please tell her that band practice has been moved to tuesday night at eight thirty . A: o . k . no problem . anything else ? B: (to be generated)
DialogWAE	**s1**: yes , please . i am a little bit of a bit of a bit of a bit of a few days . i ' m afraid i ' m not sure i ' ll get a ... **s2**: no , not at all . i ' m going to be able to get a new one - bedroom apartment . i ' m afraid i can ' t find it . i ' m going to have to ... **s3**: no , i ' ll take a look at the moment . i ' ll take a look at the <unk> . i ' ll take the right bus to get the right ...	**s1**: can ' t be a call , please . i ' ll call you back to the airport and see if you ' ll have to wait for a few minutes . **s2**: this one . the only one is at the moment of the next week . **s3**: it ' s a bit , but i ' m not sure .
DialogDiffAE	**s1**: i would like to take a taxi . **s2**: i would like to go to the airport . **s3**: oh , that ' s great . i ' ll be glad to do that .	**s1**: yes , i ' ll be there at 6 : 30 . that ' s fine . i ' ll call you later . **s2**: yes , i ' ll be there at 6 : 30 . is that all right ? **s3**: yes , i ' ll be there at 6 : 30 . is that all right ?

We tried to examine the influence of training parameters (update times in each training phases) on the diffusion model significance and the outputs' diversity. We changed the training times in every training phases (10, 5, 2 and 10, 4, 4 for training diffusion model, training phase 1, and training phase 2) and compared their metrics in Table. 4.

Table 4: Metrics comparison of different tarining parameters for DialogDiffAE.

training cases*	BLEU recall	BLEU precise	BOW	Inter-dist	Intra-dist
10, 10, 1	0.187251	0.182571	0.925465	0.334369	0.924270
10, 5, 2	0.191873	0.182305	0.928594	0.357239	0.920639
10, 4, 4	**0.214381**	**0.204169**	**0.929632**	**0.365636**	**0.935918**

*numbers represents the training times in AE training phase 1, diffusion training, and AE training phase 2.

The (10, 4, 4) training case shows all scores superior than the other cases, which not only generates more expected responses but also give responses with slightly higher diversity. Such exploration suggests that placing more weights on training the context encoder B does not downplay the performance of the diffusion, and can help the whole model learn better dialogue skills through the better trained context encoder B. The optimal training process in our model is still waiting to be explored.

However, training case (10, 4, 4)'s improvement on the response sampling diversity is still minor. Most of the outputs it generates for the same context under different sampling is still identical. We

also tried to increase the diffusion model's influence by start with 10 epochs of pre-training or finish with 10 epochs of post-training on the diffusion network and encoder A. We compare the evaluation metrics in Table. 5.

Table 5: Metrics comparison of whether to pre-train the encoder A & diffusion part for DialogDiffAE.

training strategy	BLEU recall	BLEU precise	BOW	Inter-dist	Intra-dist
Original	0.187251	**0.182571**	**0.925465**	0.334369	0.924270
Pre-train	0.174980	0.164200	0.923884	0.362123	0.897216
Post-train	**0.187492**	0.176687	0.924400	**0.380557**	**0.935400**

The pre-training of the encoder A and diffusion network will not make the diffusion output perform better in generating expected outputs. We propose that it is because without the training of encoder B, although the diffusion model can get large weight in the decoder for the first 10 epochs of training, it will not learn generating suitable result for the concatenation with well trained encoder B outputs after the whole 80 training epochs. In the end, the weight of the diffusion network may still be decreased during the following training steps. The post-training of the encoder A and diffusion network will generate more diversified responses, as shown by increased difference between BLEU recall and BLEU precise, and increased Inter-dist. But this training strategy pays for the lower performance in every single sampling (decreased BLEU precise). A trade-off exists between the response diversity and the response accuracy.

Combining the analysis above, we propose that our model has achieved a better performance than the baseline DialogWAE [6] model. We suspect that this improvement comes from a combination of a suitable encoder-decoder network structure and a limited contribution from the diffusion model. The encoder B-decoder path may have tackled most of the response generation direction like a seq2seq model, and the random sampling of the diffusion model adjust the exact output in a minor way. We suggest that this limitation might be an inevitable result from our training process and the limited dataset. DailyDial [19] is a sequence to sequence dataset that doesn't offer multiple responses under the same contexts. Therefore, in the training process it's hard for the network to see the possibilities of multiple responses and increase the importance of the diffusion part.

6 Conclusion

In this project, we studied the application of diffusion model in capturing textual latent space distribution in auto-encoder models. The problem with VAE is that it works well only when the latent space distribution is close to standard normal. We hope the injection of diffusion model can help capturing arbitrarily complex latent space distribution and thus relax VAE's distribution assumption.

Our model consists of classic auto-encoder model and an additional diffusion model. We divided the training procedure into three phases to alternatively train the auto-encoder model and the diffusion model, see Section 3.3 for more details on the training procedure. We evaluated our model on BLEU recall score, BLEU precise score, BOW score, Average Length, Inter-dist score, and Intra-dist score for dialogue generation task. The results show that our DialogDiffAE model generates more stable and concise content while DialogWAE generates more diverse results (see Table 2). With best-tuned training epoch numbers, our DialogDiffAE has superior performance according to most of the evaluation matrics (see Table 4). Human evaluation reveals that DialogDiffAE's output has better logical flow than DialogWAE's. A deeper investigation suggests that the contribution of the injected diffusion model is limited compared to the contribution of encoder-decoder model, see Section 5 for more discussion on this. One likely reason is that the dataset offers single response under the same context and thus has constrained diffusion model's generation ability.

Briefly speaking, we have successfully implemented and evaluated our proposed model 1, which is able to generate sensible response for given context content and has better performance compared to the baseline DialogWAE model under most of the evaluation metrics. The main limitation of our work is that the actual diffusion model doesn't count much for the output compared to the auto-encoder model. We investigated this phenomenon in depth (see Section 5) and proposed some potential reasons. Verification of these reasons is left for future work.

References

[1] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2013.

[2] Zichao Yang, Zhiting Hu, Ruslan Salakhutdinov, and Taylor Berg-Kirkpatrick. Improved variational autoencoders for text modeling using dilated convolutions, 2017.

[3] Keshav Rungta, Geeling Chau, Anshuman Dewangan, Margot Wagner, and Jin-Long Huang. Sentence generation and classification with variational autoencoder and bert, 2022.

[4] Changwon Ok, Geonseok Lee, and Kichun Lee. Informative language encoding by variational autoencoders using transformer. *Applied Sciences*, 12(16), 2022.

[5] Xiaoyu Shen, Hui Su, Shuzi Niu, and Vera Demberg. Improving variational encoder-decoders in dialogue generation, 2018.

[6] Xiaodong Gu, Kyunghyun Cho, Jung-Woo Ha, and Sunghun Kim. Dialogwae: Multimodal response generation with conditional wasserstein auto-encoder, 2018.

[7] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020.

[8] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models, 2020.

[9] Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. *arXiv preprint arXiv:2205.14217*, 2022.

[10] Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. Diffuseq: Sequence to sequence text generation with diffusion models, 2022.

[11] Zhengfu He, Tianxiang Sun, Kuanning Wang, Xuanjing Huang, and Xipeng Qiu. Diffusionbert: Improving generative masked language models with diffusion models, 2022.

[12] Stanislau Semeniuta, Aliaksei Severyn, and Erhardt Barth. A hybrid convolutional variational autoencoder for text generation. *arXiv preprint arXiv:1702.02390*, 2017.

[13] Wenlin Wang, Zhe Gan, Hongteng Xu, Ruiyi Zhang, Guoyin Wang, Dinghan Shen, Changyou Chen, and Lawrence Carin. Topic-guided variational autoencoders for text generation. *arXiv preprint arXiv:1903.07137*, 2019.

[14] Shuyang Dai, Zhe Gan, Yu Cheng, Chenyang Tao, Lawrence Carin, and Jingjing Liu. Apo-vae: Text generation in hyperbolic space. *arXiv preprint arXiv:2005.00054*, 2020.

[15] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[16] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014.

[17] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[18] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[19] Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. DailyDialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 986–995, Taipei, Taiwan, November 2017. Asian Federation of Natural Language Processing.

[20] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics.

[21] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.

[22] Chia-Wei Liu, Ryan Lowe, Iulian V. Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. How not to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation, 2016.

[23] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models, 2015.