Speculative Sampling in Variational Autoencoders for Dialogue Response Generation

Shoetsu Sato **Naoki Yoshinaga** **Masashi Toyoda**
Institute of Industrial Science, the University of Tokyo
{shoetsu,ynaga,toyoda}@tkl.iis.u-tokyo.ac.jp

Masaru Kitsuregawa
Institute of Industrial Science, the University of Tokyo
National Institute of Informatics
kitsure@tkl.iis.u-tokyo.ac.jp

Abstract

Variational autoencoders have been studied as a promising approach to model one-to-many mappings from context to response in chat response generation. However, they often fail to learn proper mappings. One of the reasons for this failure is the discrepancy between a response and a latent variable sampled from an approximated distribution in training. Inappropriately sampled latent variables hinder models from constructing a modulated latent space. As a result, the models stop handling uncertainty in conversations. To resolve that, we propose speculative sampling of latent variables. Our method chooses the most probable one from redundantly sampled latent variables for tying up the variable with a given response. We confirm the efficacy of our method in response generation with massive dialogue data constructed from Twitter posts.

1 Introduction

In early neural-based approaches to chat dialogue modeling, conventional encoder-decoder frameworks (Cho et al., 2014; Sutskever et al., 2014) tended to generate safe responses (Li et al., 2016). The main reason was that these frameworks model response generation as one-to-one projections from an utterance to a response, while many probable responses often exist in open-domain conversations.

The use of conditioned variational autoencoders (CVAE) is a promising approach for resolving the problem (Sohn et al., 2015; Serban et al., 2016). In these models, latent variables sampled from approximated distributions are expected to serve as a clue to handle the uncertainty in probable responses. The uncertainty can correspond to topics, domains, or styles that are not explicitly controlled.

However, the training of variational models is known to be unstable in chat response generation. When the training fails, latent variables are ignored and the models are reduced to the conventional encoder-decoders (Bowman et al., 2016). It is also possible for latent variables to work too aggressively and lead the models to generate responses that are less relevant to the contexts.

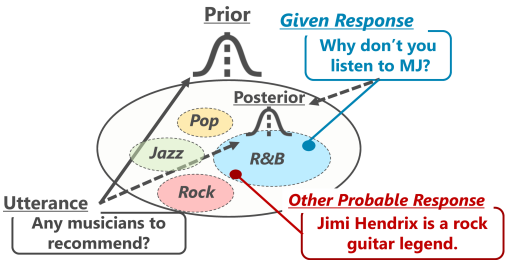


Figure 1: The posterior can produce a variable leading to another probable response: illustrative example.

Although many existing studies have tried to improve variational models (Kingma et al., 2016; Zhao et al., 2017; Shen et al., 2018; Gu et al., 2018; Fu et al., 2019) (§ 2), we postulate that there still remains a problem that degrades the models; during training, a sampled latent variable can be inappropriate to represent a given response, due to 1) immature parameters in early stages of training and 2) a trade-off in training objectives (Figure 1).

We hypothesize that the discrepancy between an unreliable latent variable and a given response can hinder models from structuring a modulated latent space. To address the problem, we propose **speculative sampling** of latent variables, a simple model-agnostic method to help variational models implicitly handle the uncertainty in conversations.

In experiments, we evaluated our method on massive open-domain dialogue data taken from Twitter. Automatic and human evaluations on the generated responses confirmed that our method improved both sensibleness and specificity of responses.

The contributions of this paper are as follows.

- We pointed out the problem of variational models that inappropriate latent variables in training can disorganize the latent space.

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 4739–4745

- We proposed a simple and model-agnostic method for modulating the latent space by sampling proper latent valiables in training.
- We empirically confirmed that our method improved the quality of generated responses both in automatic and human evaluation.

2 Related work

CVAE-based models have been studied as one of the promising solutions to the safe response problem in chat response generation (Sohn et al., 2015; Serban et al., 2016). However, the difficulty in optimization has been studied mainly from a machine learning perspective; models with a sufficient number of parameters can ignore latent variables and work similarly to conventional encoder-decoder models (a.k.a. KL vanishing). Thus, many studies have proposed methods to control the optimization of variational models (Bowman et al., 2016; Zhao et al., 2017; Kingma et al., 2016; Shen et al., 2018; Li et al., 2018; Gu et al., 2018; Gao et al., 2019; He et al., 2019). They mainly focused on regularization, the architecture, and the training schedule.

To design a latent space where the relevance and diversity of probable outputs are reflected geometrically, Gao et al. (2019) proposed SPACEFUSION. Among the aforementioned studies, their approach shares with us a similar goal of organizing the latent space. Kruengkrai (2019) proposed to sample multiple latent variables in text modeling, similarly to our method. However, the intention is different as their method was for better approximation of the expected reconstruction term in training.

3 Speculative Latent Variables Sampling

The main concept of variational response generation models is to handle the uncertainty in conversations as the randomness of a latent variable z sampled from the model's distribution. With parameters θ and ϕ, the model first approximates prior and posterior distributions $p_\theta(z|x)$ and $q_\theta(z|x,y)$ from the utterance x and response y in a conversation. Then, the model samples a latent variable z, from the distributions and feeds it to the decoder to compute the probability of the response $p_\phi(y|x,z)$. Ideally, the latent variable $z_p \sim p_\theta(z|x)$ and $z_q \sim q_\theta(z|x,y)$ are representations that can generate all probable responses to x and the given response y, respectively. In training, the following objective, which combines the reconstruction loss and Kullback-Leibler (KL) divergence D_{KL}, is maximized:

$$\begin{aligned}\log p(y|x) &= \log \int_z p_\phi(y|x,z)p_\theta(z|x)dz \\ &\geq \mathbb{E}_{q_\theta(z|x,y)}[\log p_\phi(y|x,z_q)] \\ &\quad - D_{\mathrm{KL}}\left(q_\theta(z|x,y)|p_\theta(z|x)\right). \qquad (1)\end{aligned}$$

Here, in training, what if a latent variable z_q sampled from the posterior distribution $q_\theta(z|x,y)$ is inappropriate to represent the response y? Although the distribution is approximated under the observation of the response, this is still possible because the parameters for the approximation are incomplete during training. Furthermore, optimizing KL-divergence does not necessarily help the reconstruction of y; the posterior distribution is promoted to be similar to the prior distribution that also covers other probable responses. As a result of this discrepancy, the training becomes skewed. The model can lose track of the correspondence between sampled latent variables and responses to be generated, and its latent space are disorganized.

To address the problem, we propose **speculative sampling** of latent variables, a simple method to disentangle the discrepancy. Specifically, we sample k latent variables $\{z_0, z_1, \cdots, z_{k-1}\}$ from the posterior distribution, and compute the loss for each variable in training. We then compute gradients only from the latent variable that has the least loss among the sampled variables. This simple modification prevents models from tying up unreliable variables with given responses.[1]

4 Experimental Setup

4.1 Models

We evaluate the effect of our proposed method in dialogue response generation. We used a subword-based Transformer-based Conditional Variational Autoencoder (T-CVAE) (Wang and Wan, 2019) that we implemented with fairseq (v0.8.0)[2] (Ott et al., 2019), as the core architecture for the dialogue models. T-CVAE is a combination of Transformer (Vaswani et al., 2017) and CVAE (Kingma et al., 2014), both of which are strong baselines

[1] Our method was inspired by dynamic oracle (Goldberg and Nivre, 2012) that allows a shift-reduce dependency parser to choose an easy-to-decode oracle operation among all the possible oracle operations that will ultimately reach the gold tree. Analogously, we aim to provide the most probable latent variables that can reach a given response in training.

[2] https://github.com/pytorch/fairseq

commonly employed for text generation. We followed the major hyperparameters of Transformer-base (Vaswani et al., 2017). We show detailed hyperparameters in Appendix.

The compared models are as follows.

T-CVAE: vanilla T-CVAE (Wang and Wan, 2019).

T-CVAE + Cyclical annealing cyclically adjusts the weight to the KL-divergence loss in training (Fu et al., 2019). We set one epoch as one cycle.

SPACEFUSION adds losses for fusing the vector space of inputs and outputs (Gao et al., 2019).[3]

T-CVAE + BoW loss adds the bag-of-word (BoW) loss to the training objective in Eq. 1. This is a constraint that ties up a latent variable with the bag-of-words of a given response (Zhao et al., 2017).

T-CVAE + Monte Carlo (MC) sampling samples five latent variables and use the average for computing the training loss (Kruengkrai, 2019).[4]

T-CVAE + Speculative sampling: refer to § 3.[5]

The models can be divided into three categories: 1) controlling the training schedule (**Cyclical annealing**), 2) adding constraints on the latent space (**SPACEFUSION** and **BoW loss**), and 3) changing the sampling method in training (**Monte Carlo sampling** and **Speculative sampling**).

4.2 Datasets and Preprocessing

To evaluate the ability of models to generate diverse responses, the dataset needs to contain various topics and styles. Following existing studies (Ritter et al., 2011; Serban et al., 2017; Adiwardana et al., 2020; Su et al., 2020), we constructed massive English and Japanese dialogue datasets from social media conversations. Concretely, we exploited Twitter posts while treating a post and the subsequent replies as a conversation.

We used posts in 2017 and 2018 for both training and development, posts in 2019 for testing. They were randomly sampled from our Twitter archive (Nishi et al., 2016) collected via the Twitter API.[6] We filtered out noisy posts with a rule-based filtering following Adiwardana et al. (2020). The numbers of English conversations were 19,627,263 for training, 196,253 for development, and 97,433 for testing. The numbers of Japanese conversations were 18,116,756 for training, 191,890 for development, and 96,276 for testing.

We employed `multi-bleu.perl` in Moses toolkit (v4.0)[7] for tokenizing English text. This tokenization was applied only for generated outputs to compute automatic evaluation metrics. We employed MeCab[8] for tokenizing Japanese text.

From the training data, we trained subword tokenization models through unigram language modeling (Kudo and Richardson, 2018) and CBOW vectors (Mikolov et al., 2013) for initialization of the model's embedding layers.

For human evaluation, we manually chose 100 conversations from the Japanese test data. This was because randomly sampled conversations 1) can be difficult to understand for evaluators due to the lack of contexts or knowledge, and 2) can contain utterances where possible responses are not diverse (*e.g.*, greetings or yes/no questions). Using such conversations for human evaluation not only increases annotation costs, but also makes it difficult to analyze differences between models. We will also release these conversations as a challenging set that enables developers to evaluate the ability of models for diversification with a low cost.

4.3 Evaluation Metrics

For automatic evaluation, we employed several common metrics: case-sensitive BLEU (Papineni et al., 2002) in Moses[9] and dist-n (Li et al., 2016). Additionally, we compared the KL-divergence of trained models to investigate how the resolution of KL vanishing affected generated responses.

We also conducted human evaluation with similar metrics to Adiwardana et al. (2020). Annotators provided scores of 1) **sensibleness** and 2) **specificity** from 1 to 5 for each anonymized response.[10]

5 Results

This section reports results of automatic (§ 5.1) and human evaluations (§ 5.2) of generated responses on the Twitter datasets, and then analyzes the models' outputs (§ 5.3).

[3]Note that the covariance of Gaussian distribution of this model is not parametrized, and thus, only this model is slightly different from other models based on T-CVAE.

[4]In T-CVAE, latent variables are combined with the last decoder state before softmax. Thus, we simply averaged the latent variables instead of averaging the decoder states.

[5]From the validation loss, we chose five as the number of sampled latent variables.

[6]`https://developer.twitter.com/`

[7]`https://github.com/moses-smt/mosesdecoder`

[8]`https://github.com/taku910/mecab`

[9]`http://www.statmt.org/moses`

[10]When generated responses are too noisy for the evaluators not to evaluate the specificity, the specificity is scored as zero.

	BLEU	dist-1	dist-2	KLD
Reference	-	6.20	41.25	-
T-CVAE	0.71	0.71	3.55	0.00
Cyclical annealing	0.68	0.72	3.68	0.09
SPACEFUSION	**0.76**	0.60	2.87	-
BoW loss	0.30	1.58	10.99	24.04
MC sampling	0.09	**3.02**	**21.54**	9.81
Speculative sampling				
$K = 2$	0.62	0.76	4.31	0.75
$K = 5$	0.51	0.89	5.48	1.96
$K = 10$	0.47	0.90	6.06	2.77
$K = 20$	0.43	0.96	6.74	3.53
$K = 40$	0.41	0.98	7.03	3.98

Table 1: Automatic evaluation results for English data.

	BLEU	dist-1	dist-2	KLD
Reference	-	4.11	30.60	-
T-CVAE	2.48	1.14	4.24	0.00
Cyclical annealing	2.73	1.19	4.24	0.09
SPACEFUSION	2.86	1.50	4.74	-
BoW loss	1.97	1.56	8.74	24.07
MC sampling	0.53	**1.79**	**18.25**	11.83
Speculative sampling				
$K = 2$	2.69	1.28	4.95	0.62
$K = 5$	**2.91**	1.57	6.48	1.57
$K = 10$	2.70	1.54	7.00	2.19
$K = 20$	2.40	1.51	7.28	2.91
$K = 40$	2.38	1.55	7.75	3.24

Table 2: Automatic evaluation results for Japanese data.

5.1 Automatic Evaluation

Table 1 and 2 show the results of automatic evaluation. For English data, we observed the trade-off between the BLEU and dist-n scores more clearly. Among the compared models, **Cyclical annealing** slightly improved dist-n scores. Although the gains obtained by **SPACEFUSION** varied across languages, we did not observe large impacts on the results in both languages. It can vary by adjusting hyperparameters. While **BoW loss** and **MC sampling** resolved KL vanishing and achieved remarkably high dist-n scores, the BLUE scores were degraded. We will discuss the reason in § 5.2.

Although we set $K = 5$ as the hyperparameter of the proposed model for human evaluation, we also evaluated the model with different k to explore its effect. In all settings, the dist-n scores were consistently improved while keeping the BLUE score compared to **T-CVAE**. Note that the vanilla T-CVAE corresponds to the proposed model with $k = 1$. Interestingly, the larger K we chose, the dist-2 and KL-divergence became higher. This result supports our hypothesis discussed in § 3 – providing probable latent variables in training can help models construct an organized latent space.

Note that the proposed method did not significantly increase the training time per epoch, as gradients were only computed for the most probable latent variable. On our server with four NVIDIA Quadro P6000 GPUs, the increase was 25% for $K = 5$ compared to T-CVAE.

5.2 Human Evaluation

Table 3 shows the results of human evaluation for the Japanese data. The results were similar to those shown in Table 2. **SPACEFUSION** and **speculative sampling** achieved relatively high sensibleness (i.e., relatedness to the context). The specificity of **BoW loss** and **Speculative sampling** were remarkably higher than other models while the sensibleness of **BoW loss** was degraded from **T-CVAE**. The low specificity of **MC sampling** was due to the low sensibleness; we allowed the evaluators to assign low specificity to responses when they were too noisy to evaluate.

	Sensibleness	Specificity	Avg.
Reference	4.67	4.33	4.50
T-CVAE	3.58	1.35	2.46
Cyclical annealing	3.58	1.29	2.44
SPACEFUSION	3.66	1.42	2.54
BoW loss	3.04	**1.58**	2.31
MC sampling	1.42	0.70	1.06
Speculative sampling	**3.94**	1.52	**2.73**

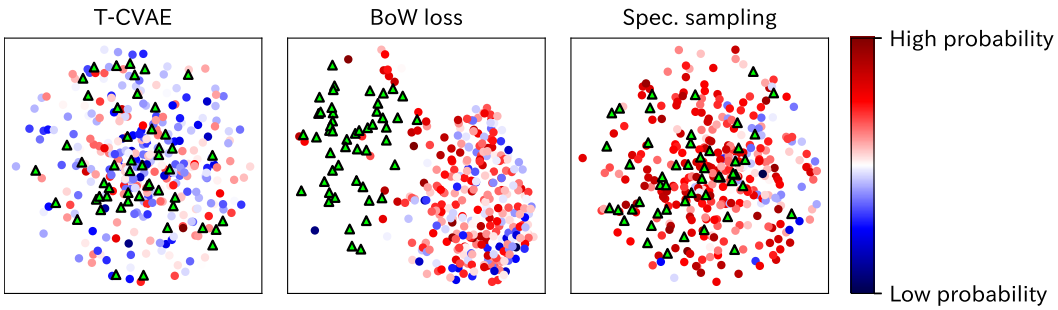
Table 3: Human evaluation results for Japanese data. Pearson correlation between evaluators was 0.69.

We consider the reason for the decrease in sensibleness compared to **T-CVAE** as follows. **BoW loss** worked too strongly as a constraint on the latent space and the distributions became enlarged. In **MC sampling**, latent variables close to the mean of the posterior distribution were more likely to be trained. As a result, in testing, it is possible for the models to sample latent variables from unreliable regions that were not optimized enough.

Overall, the high specificity in the baseline models tended to result in low sensibleness in return. Meanwhile, **Speculative sampling** achieved comparable results in both sensibleness and specificity.

5.3 Analysis and Discussion

To investigate the latent space learned by the models, for each utterance-response pair in testing data,



Utterance: 野菜を美味しく食べる方法を募集します (I'm looking for delicious ways to eat vegetables.)
Response: 野菜を切って洗います。ドレッシングをかけます。美味しく食べます。 (Cut and wash vegetables. Put dressing. Enjoy.)

Figure 2: Visualization of sampled latent variables and generation probabilities for an utterance-response pair in test data. Dots and triangle denote variables sampled from **prior** and **posterior** distributions, respectively.

we sampled 300 and 50 latent variables from the prior and the posterior distribution of the compared models, respectively. And then, each model computed log probabilities to generate the reference response from the sampled variables. The probabilities were normalized for each model.

Figure 2 plots the latent variables and the probabilities for the three representative models – **T-CVAE**, **BoW loss**, and **Speculative sampling** by using t-SNE (Maaten and Hinton, 2008) for dimension reduction.[11] This clearly shows the difference in latent spaces among the compared models.

As shown in Figure 1, the latent space of variational models should meet the conditions: 1) the geometry of the latent space reflects the meaning of responses (i.e., similar latent variables generate similar responses) and 2) the prior and posterior distributions overlap each other. If the former is not met, generalization in training for the latent space becomes complicated, and models tend to ignore latent variables (i.e., KL vanishing). If the latter is not met, variables leading to the reference response are less likely to be sampled in testing, leading models to noisy outputs (i.e., too large KLD).

In our settings, **T-CVAE** did not satisfy the former condition; the closeness of variables to the posterior (green triangles) was irrelevant to the probabilities. Conversely, **BoW loss** did not satisfy the latter condition; although there existed a region corresponding to the reference responses, variables were rarely sampled from the region in testing. **Speculative sampling** tended to satisfy both conditions; while preventing KL vanishing, our method did not put an explicit restriction to the latent space, which successfully made the two distributions close by optimizing KL divergence.

[11] https://github.com/huguyuehuhu/fastTSNE

Utterance	急募 喉の痛みの緩和方法 (*Any ideas on how to relieve sore throat?*)
Reference	マヌカハニーを舐める (*Eat Manuka honey.*)
T-CVAE	病院に行った方がいいですよ。 (*You should go to a hospital.*)
Cyclical annealing	お大事にしてください...! (*I hope you get well soon.*)
SPACEFUSION	お大事になさってください... (*I hope you get well soon.*)
BoW loss	胃腸炎にならなくていいと思います。 (*I think you don't have to have gastroenteritis.*)
MC sampling	自分のやつ! (*Your own!*)
Speculative sampling	ビタミンCを摂るといいよ。 (*Take vitamin C.*)

Table 4: Examples of generated outputs.

Output Examples Table 4 shows example outputs. Despite using the variational model, safe responses were observed. Meanwhile, the responses generated by the models with high KL-divergence (**BoW loss** and **MC sampling**) were more specific but less sensible. **Speculative sampling** tended to make responses with topic-specific words (*e.g.*, "vitamin C"), while keeping the sensibleness.

6 Conclusions

In this study, we aimed to help dialogue models construct a organized latent space that can capture implicit uncertainty in conversations. We proposed speculative sampling of latent variables, a method for mitigating the discrepancy in training between sampled latent variables and corresponding responses. Experimental results in a response generation test with massive Twitter dialogue data confirmed that our proposed method improved both sensibleness and specificity of generated responses. We will release all code and IDs of Twitter posts.[12]

[12] https://github.com/jack-and-rozz/speculative_sampling

Acknowledgements

The research was supported by NII CRIS collaborative research program operated by NII CRIS and LINE Corporation.

Ethical considerations

The Twitter data we used in this paper was collected through the official Twitter APIs and is in compliance with with Twitter's terms of service. To allow Twitter users to delete posts or change them to private access, we will only release the IDs of posts.

References

Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, et al. 2020. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL 2016)*, pages 10–21.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pages 1724–1734.

Hao Fu, Chunyuan Li, Xiaodong Liu, Jianfeng Gao, Asli Celikyilmaz, and Lawrence Carin. 2019. Cyclical annealing schedule: A simple approach to mitigating KL vanishing. 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) (NAACL 2019)*, pages 240–250.

Xiang Gao, Sungjin Lee, Yizhe Zhang, Chris Brockett, Michel Galley, Jianfeng Gao, and Bill Dolan. 2019. Jointly optimizing diversity and relevance in neural response generation. 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) (NAACL-HLT 2019)*, pages 1229–1238.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings.

Yoav Goldberg and Joakim Nivre. 2012. A dynamic oracle for arc-eager dependency parsing. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012)*, pages 959–976.

Xiaodong Gu, Kyunghyun Cho, Jung-Woo Ha, and Sunghun Kim. 2018. Dialogwae: Multimodal response generation with conditional wasserstein auto-encoder. *arXiv preprint arXiv:1805.12352*.

Junxian He, Daniel Spokoyny, Graham Neubig, and Taylor Berg-Kirkpatrick. 2019. Lagging inference networks and posterior collapse in variational autoencoders. In *Proceedings of the International Conference on Learning Representations (ICLR 2019)*.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR 2015)*.

Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. 2014. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems (NIPS 2014)*, volume 27.

Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. 2016. Improved variational inference with inverse autoregressive flow. In *Advances in neural information processing systems (NIPS 2016)*, pages 4743–4751.

Canasai Kruengkrai. 2019. Better exploiting latent variables in text modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019)*, pages 5527–5532.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP 2018)*, pages 66–71.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. 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 (NAACL-HLT 2016)*, pages 110–119.

Juncen Li, Ping Luo, Fen Lin, and Bo Chen. 2018. Conversational model adaptation via KL divergence regularization. In *Proceedigs of the 32nd AAAI Conference on Artificial Intelligence (AAAI 2018)*, pages 5213–5219.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26 (NIPS 2013)*, pages 3111–3119.

Ryosuke Nishi, Taro Takaguchi, Keigo Oka, Takanori Maehara, Masashi Toyoda, Ken-ichi Kawarabayashi, and Naoki Masuda. 2016. Reply trees in twitter: data analysis and branching process models. *Social Network Analysis and Mining*, 6(1):26.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations) (NAACL 2019)*, pages 48–53.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002)*, pages 311–318.

Alan Ritter, Colin Cherry, and William B Dolan. 2011. Data-driven response generation in social media. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP 2011)*, pages 583–593.

Iulian Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedigs of the 30th AAAI Conference on Artificial Intelligence (AAAI 2016)*.

Iulian Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI 2017)*.

Xiaoyu Shen, Hui Su, Shuzi Niu, and Vera Demberg. 2018. Improving variational encoder-decoders in dialogue generation. In *Proceedings of 32nd AAAI Conference on Artificial Intelligence (AAAI 2018)*, pages 5456—-5463.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. 2015. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems 28 (NIPS 2015)*, volume 28, pages 3483–3491.

Hui Su, Xiaoyu Shen, Sanqiang Zhao, Zhou Xiao, Pengwei Hu, Randy Zhong, Cheng Niu, and Jie Zhou. 2020. Diversifying dialogue generation with non-conversational text. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*, pages 7087–7097.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS 2014)*, pages 3104–3112.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, pages 5998–6008.

Tianming Wang and Xiaojun Wan. 2019. T-cvae: Transformer-based conditioned variational autoencoder for story completion. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI 2019)*, pages 5233–5239.

Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. 2017. 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) (ACL 2017)*, pages 654–664.

# encoder/decoder layers	6	Label smoothing rate	0.1
# attention heads	8	Dropout rate	0.1
Dim. of embeddings	512	Init. learning rate	1e-3
Dim. of Transformer	2048	(warmup)	1e-7
Vocab. size	16k	Beam size	5
Max. tokens in batch	27k	Max. training steps	250k

Table 5: Hyperparameters of models.

A Detailed Experimental Settings

Table 5 shows the hyperparameters of the compared models. We used Adam Optimizer (Kingma and Ba, 2015) with $\beta_1 = 0.9$ and $\beta_2 = 0.98$. The learning rate started from 10^{-7} and linearly increased to 10^{-3} for warm-up during the first 4,000 step. And then, the learning rate was decayed to 10^{-9} with inverse square-root scheduling.

We applied dropout to: 1) input embeddings combined with positional embeddings, 2) outputs from feed-forward layers, 3) outputs from self-attention layers, and 4) outputs from encoder-decoder attention layers. The parameters of models were initialized by Xavier initializer (Glorot and Bengio, 2010).

We randomly sampled 1,000,000 sentences from the training data to train CBOW vectors and subword tokenization models, due to the computational costs. For this training, we adopted WORD2VEC[13] and Sentencepiece[14] with default hyperparameters.

[13] https://code.google.com/archive/p/word2vec/

[14] https://github.com/google/sentencepiece