Multi-Referenced Training for Dialogue Response Generation

Tianyu Zhao[†‡] **Tatsuya Kawahara**[‡]
[†]rinna Co., Ltd. [‡]Kyoto University
zhaoty.ting@gmail.com

Abstract

In open-domain dialogue response generation, a dialogue context can be continued with diverse responses, and the dialogue models should capture such one-to-many relations. In this work, we first analyze the training objective of dialogue models from the view of Kullback–Leibler divergence (KLD) and show that the gap between the real world probability distribution and the single-referenced data's probability distribution prevents the model from learning the one-to-many relations efficiently. Then we explore approaches to multi-referenced training in two aspects. Data-wise, we generate diverse pseudo references from a powerful pretrained model to build multi-referenced data that provides a better approximation of the real-world distribution. Model-wise, we propose to equip variational models with an expressive prior, named linear Gaussian model (LGM). Experimental results of automated evaluation and human evaluation show that the methods yield significant improvements over baselines.[1]

1 Introduction

Open-domain dialogue modeling has been formulated as a seq2seq problem since Ritter et al. (2011) and Vinyals and Le (2015) borrowed machine translation (MT) techniques (Koehn et al., 2007; Sutskever et al., 2014) to build dialogue systems, where a model learns to map from *one* context to *one* response. In MT, *one*-to-*one* mapping is a reasonable assumption since an MT output is highly constrained by its input. Though we may use a variety of expressions to translate the same input sentence, these different translations still highly overlap with each other lexically and semantically

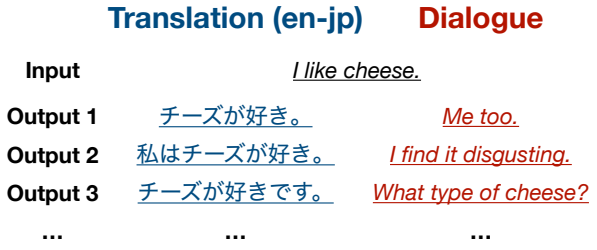

Figure 1: Examples of multiple valid outputs given the same input in machine translation and dialogue.

(see the translation example in Figure 1), and learning from one output reference is often sufficient for training a good MT system (Kim and Rush, 2016). In dialogues, however, the same input can be continued with multiple diverse outputs which are different in both the used lexicons and the expressed semantic meanings (see the dialogue example in Figure 1). Learning from barely one output reference ignores the possibility of responding with other valid outputs and is thus insufficient for building a good dialogue system.

The current dialogue modeling paradigm is largely derived from MT research, and it trains dialogue models with one output reference given each input. In this paper, we will investigate why single-referenced training harms our dialogue models and how to apply multi-referenced training.

2 Why Multi-Referenced Training Matters?

A dialogue context X can be continued with a set of different responses $\{Y_1, \cdots, Y_i, \cdots\}$. In the training of a response generation model, we expect to model the *real probability distribution* $P(\boldsymbol{Y}|X)$ with *model probability distribution* $P_\theta(\boldsymbol{Y}|X)$ for each context X, where θ is the model parameters. In most scenarios, however, we can only rely on

[1]Code and data are available at https://github.com/ZHAOTING/dialog-processing/tree/master/src/tasks/response_gen_multi_response.

Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 190–201
July 29–31, 2021.

a data set $D = \{(X^{(j)}, Y_1^{(j)})\}_j^{|D|}$,[2] where only one valid response is presented. This results in a *data probability distribution* $P_{\mathrm{D}}(\boldsymbol{Y}|X)$ that is very different from $P(\boldsymbol{Y}|X)$. In fact, $P_{\mathrm{D}}(\boldsymbol{Y}|X)$ is an one-hot vector where the first element is 1 while others are 0.

Emprical training objective As a result, we optimize a model to match the *model probability distribution* and the *data probability distribution*. From the view of Kullback–Leibler divergence (KLD), we can see it as to minimize $D_{\mathrm{KL}}(P_{\mathrm{D}}||P_\theta)$:

$$-\sum_i P_{\mathrm{D}}(Y_i|X) \log \frac{P_\theta(Y_i|X)}{P_{\mathrm{D}}(Y_i|X)},$$

which is identical to minimize the following target function after ignoring terms that are not related to the model parameter θ:

$$\begin{aligned}\mathcal{L}_{\mathrm{D}}(X, \boldsymbol{Y}) &= -\sum_i P_{\mathrm{D}}(Y_i|X) \log P_\theta(Y_i|X) \\ &= -\sum_i \mathbb{1}\{i = 1\} \log P_\theta(Y_i|X) \\ &= -\log P_\theta(Y_1|X).\end{aligned}$$

The resulting objective is the negative log likelihood (NLL) loss function commonly used in the implementation of dialogue models.

Ideal training objective We hope to minimize the KLD between the *model probability distribution* and the *real probability distribution*, $D_{\mathrm{KL}}(P||P_\theta)$:

$$-\sum_i P(Y_i|X) \log \frac{P_\theta(Y_i|X)}{P(Y_i|X)},$$

which is identical to minimize:

$$\mathcal{L}^*(X, \boldsymbol{Y}) = -\sum_i P(Y_i|X) \log P_\theta(Y_i|X).$$

However, $\mathcal{L}^*$ is intractable because 1) there are often an enormous number of valid responses, and 2) we cannot obtain the real probability of a certain response $P(Y_i|X)$.

The problem and proposed solutions The gap between $\mathcal{L}_{\mathrm{D}}$ and $\mathcal{L}^*$ is caused by the difference between $P_{\mathrm{D}}(\boldsymbol{Y}|X)$ and $P(\boldsymbol{Y}|X)$, and it prevents dialogue models from learning one-to-many mappings efficiently. To alleviate this problem, we propose methods to allow for multi-referenced training in two aspects.

[2]For simplicity, we define a response in D as the first response to its context, and thus its subscript is 1. We will omit the superscript in the rest of the paper.

- Data-wise, we replace the original data distribution $P_{\mathrm{D}}(\boldsymbol{Y}|X)$ with an approximated real distribution $P_\phi(\boldsymbol{Y}|X)$ by generating up to 100 pseudo references from a *teacher model* parameterized by ϕ. We show that using the newly created data yields significant improvement.
- Model-wise, we argue that a model requires an encoder of large capacity to capture sentence-level diversity, and thus we propose to equip the variational hierarchical recurrent encoder-decoder (VHRED) model with a linear Gaussian model (LGM) prior. The proposed model outperforms VHRED baselines with unimodal Gaussian prior and Gaussian Mixture Model (GMM) prior in evaluation experiments.

3 Related Works

3.1 Knowledge Distillation

In the context of machine translation, Kim and Rush (2016) proposed that a *teacher model*'s knowledge can be transferred to a *student model* on a sequence level. They showed that transferring sequence-level knowledge is roughly equal to training on sequences generated by the *teacher model* as references. However, *one* generated reference given each input is sufficient for transferring the teacher's MT knowledge, while we will show in following experiments that training with multiple generated references can yield far better results in dialogue response generation. This confirms our earlier hypothesis that the one-to-many nature is an important characteristic that distinguishes open-domain dialogue modeling from other tasks such as machine translation.

In task-oriented dialogues, Peng et al. (2019) proposed to transfer knowledge from multiple teachers for multi-domain task-oriented dialogue response generation via policy distillation and word-level output distillation. Tan et al. (2019) applied a similar approach to multilingual machine translation. Kuncoro et al. (2019) transferred syntactic knowledge from recurrent neural network grammar (RNNG, Dyer et al., 2016) models to a sequential language model.

3.2 Data Augmentation and Manipulation

The multi-referenced training approach can be seen as a data augmentation method. Prior works on data augmentation in text generation tasks often operate on a word level while our method performs

sentence-level augmentation. Niu and Bansal (2019) proposed to apply semantic-preserving perturbations to input words for augmenting data in dialogue tasks. Zheng et al. (2018) investigated generating pseudo references by compressing existing multiple references into a lattice and picking new sequences from it. Hu et al. (2019) used finetuned BERT (Devlin et al., 2019) as the data manipulation model to generate word substitutions via reinforcement learning.

Another line of research focuses on filtering high-quality training examples for dialogue response generation. Csáky et al. (2019) proposed to remove generic responses using an entropy-based approach. Shang et al. (2018) trained a data calibration network to assign higher instance weight to more appropriate responses.

3.3 Expressive Dialogue Models

Besides manipulating the training data, dialogue researchers have attempted to strengthen dialogue models' capacity for capturing complex relations between the input context and the output responses. Zhou et al. (2017) incorporated mechanism embeddings $\mathbf{m}$ into a seq2seq model for dialogue response generation. The mechanism-aware model decodes a response by selecting a mechanism embedding $\mathbf{m_k}$ and combining it with context encoding $\mathbf{c}$. Therefore, the model is capable of generating diverse responses by choosing different mechanisms. Zhang et al. (2018) borrowed the conditional value-at-risk (CVaR) from finance as an alternative to sentence likelihood (which is negated $\mathcal{L}_{\text{D}}$) for optimization. Optimizing the CVaR objective can be seen as rejecting to optimize on easy instances whose model probabilities are larger than a threshold α. Qiu et al. (2019) proposed a two-step VHRED variant for modeling one-to-many relation. In the first step, they forced the dialogue encoding vector $\mathbf{c}$ to store common features of all response hypotheses $Y_{2:N+1}$ by adversarial training. In the second step, they trained the latent variable $\mathbf{z}$ to capture response-specific information by training with a multiple bag-of-words (MBoW) loss. These three methods will be compared with the proposed model in this work as they have focused on modeling one-to-many relations in dialogue response generation.

Gao et al. (2019) relied on vocabulary prediction to model sentence-level discrepancy. Chen et al. (2019) utilized a mechanism-based architecture and proposed a posterior mapping method to select the most proper mechanism. Gu et al. (2019) proposed to train latent dialogue models in the framework of generative adversarial network (GAN). They optimized the model by minimizing the distance between its prior distribution and its posterior distribution via adversarial training.

4 Preliminary

4.1 Models

HRED We use the hierarchical recurrent encoder decoder (HRED, Serban et al., 2016) as the baseline model, where a hierarchical RNN-based encoder $\mathcal{E}_\theta(\cdot)$ encodes the context X and produces an encoding vector $\mathbf{c}$, and an RNN-based decoder $\mathcal{D}_\theta(\cdot)$ takes $\mathbf{c}$ as input and computes the conditional probability of a response $P_\theta(Y_i|X)$ as the product of word probabilities.

$$\begin{aligned}\mathbf{c} &= \mathcal{E}_\theta(X)\\ P_\theta(Y_i|X) &= \textstyle\prod_{l=1}^{L} P_\theta(Y_{i,l}|Y_{i,:l-1}, X)\\ &= \textstyle\prod_{l=1}^{L} \mathcal{D}_\theta(Y_{i,l}|Y_{i,:l-1}, \mathbf{c}),\end{aligned}$$

where $Y_{i,j}$ stands for the j-th word in Y_i and L is the length of Y_i.

VHRED For a given context, the HRED produces a fixed-length encoding vector $\mathbf{c}$ and relies on it to decode various responses. However, the one-to-many mapping in dialogues is often too complex to capture with a single vector $\mathbf{c}$. Serban et al. (2017) proposed variational HRED (VHRED) and used a stochastic latent variable $\mathbf{z}$ that follows a multivariate Gaussian distribution to strengthen the model's expressiveness.

$$\begin{aligned}\boldsymbol{\mu}, \boldsymbol{\sigma} &= \text{MLP}_\theta(\mathbf{c})\\ \mathbf{z} &\sim \text{Gaussian}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2\boldsymbol{I})\\ P_\theta(Y_i|X) &= \textstyle\prod_{l=1}^{L} \mathcal{D}_\theta(Y_{i,l}|Y_{i,:l-1}, \mathbf{c}, \mathbf{z}),\end{aligned}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}^2\boldsymbol{I}$ are parameters of the Gaussian distribution. In order to mitigate the infamous *posterior collapse* problem in variational models, it is common to apply tricks such as annealing KLD loss (Bowman et al., 2016) and minimizing a bag-of-words (BoW) loss (Zhao et al., 2017).

VHRED with GMM prior Gu et al. (2019) showed that the performance of the vanilla VHRED is limited by the single-modal nature of Gaussian distribution, and thus they proposed to use as prior

a Gaussian Mixture Model (GMM) with K components to capture multiple modes in $\mathbf{z}$'s probability distribution, such that $\mathbf{z}$ is sampled in the following way:

$$\boldsymbol{\mu}_k, \boldsymbol{\sigma}_k, \pi_k = \text{MLP}_{\theta,k}(\mathbf{c})$$
$$\mathbf{z} \sim \text{GMM}(\{\boldsymbol{\mu}_k, \boldsymbol{\sigma}_k^2 \boldsymbol{I}, \pi_k\}_{k=1}^K),$$

where π_k is the weight of the k-th component. We refer to the VHRED with K-component GMM prior as VHRED$_{gmmK}$.

GPT2 We finetune a pre-trained medium-sized GPT2 (Radford et al., 2019) on dialogues and use it as the *teacher model* to obtain $P_\phi(\boldsymbol{Y}|X)$ as an approximation of $P(\boldsymbol{Y}|X)$. GPT2 has been shown to reach low perplexity on real-world texts, and it can generate high-quality responses (Wolf et al., 2019; Zhang et al., 2019). Therefore, we expect it to provide a relatively accurate approximation of the real-world distribution.

4.2 Data

We use the DailyDialog corpus (Li et al., 2017) to investigate the effects of the proposed methods. We make a roughly 0.8:0.1:0.1 session-level split for training, validation, and test, respectively.[3]

4.3 Metrics

Automated Metrics We use perplexity on the test data as the metric for intrinsic evaluation. For extrinsic evaluation, we choose BLEU-2 and three types of word embedding similarities (Embedding Extrema, Embedding Average, Embedding Greedy) to measure the closeness between a hypothesis and the corresponding ground-truth reference. For diversity evaluation, we choose to count the number of generated unigram and bigram types at a corpus-level.

Dialogue Response Evaluator Besides the automated metrics above, we also use RoBERTa-eval, a model-based dialogue response evaluator, to approximate human judgement (Zhao et al., 2020). RoBERTa-eval computes the appropriateness (a real value from 1 to 5) of a response hypothesis by conditioning on its context instead of by comparing with its reference. It has been shown to correlate with human judgement significantly better than automated metrics. The authors reported Pearson's ρ = 0.64 and Spearman's ρ = 0.66 on the DailyDialog corpus.

[3]See the Appendix for more details about the data set.

Human Evaluation Following Adiwardana et al. (2020), we ask Amazon MTurk human annotators to evaluate each response on two criteria, sensibleness and specificity. Both metrics take binary values, and we use their average (knowns as Sensibleness and Specificity Average, SSA) to assess the overall quality.

5 Proposal: Enhancing Data for Multi-Referenced Training

To enhance the training data, we try to close the gap between $P_{\text{D}}(\boldsymbol{Y}|X)$ and $P(\boldsymbol{Y}|X)$. Since all probability mass is on a single response in $P_{\text{D}}(\boldsymbol{Y}|X)$, the gap can be closed by assigning some mass to other valid responses. We use a finetuned GPT2$_{md}$ to generate N hypotheses as valid responses, and let the probability mass to be assigned to them uniformly. It results in $P_\phi(\boldsymbol{Y}|X)$ wherein N elements have $\frac{1}{N}$ probability. The new training objective is:

$$\tilde{\mathcal{L}}^*(X, \boldsymbol{Y}) = -\tfrac{1}{N}\textstyle\sum_{i=2}^{N+1} \log P_\theta(Y_i|X),$$

where we assume responses Y_2 to Y_{N+1} are generated responses.

Training with the new loss function can be achieved by directly replacing the ground-truth responses in the training data with the hypotheses.[4]

Sequences generated by beam search often highly overlap both lexically and semantically (Li et al., 2016). Therefore, we use nucleus sampling with top probability 0.95 (Holtzman et al., 2019) to generate 100 hypotheses as for each context in the training data.

5.1 Training with Hypotheses

In this part, we compare baseline HRED models trained with only ground truth (GT) and with different numbers of hypotheses. Since using N hypotheses makes the training data N times larger, we accordingly adjust the maximum number of training epochs. We found that all the models can converge in the given epochs. [5]

As shown in Table 1, replacing 1 GT with 1 hypothesis yields a boost on most metrics. Further increasing the number of hypotheses will continue to improve the model's performance. It is worth noting that when the number of hypotheses

[4]We will refer to the original response as ground truth and the generated responses as hypotheses. A reference can be either a ground-truth response or a hypothesis response.

[5]See the Appendix for experimental settings and statistics of model size and training cost.

Model	Param (in M)	Trn Time (in sec.)	Data	ppl	BLEU-2	Embedding Similarity Ext	Avg	Grd	Reval	D1	D2
Teacher model											
$GPT2_{md}$	338.39	3000	1 GT	21.16	8.67	41.02	65.17	48.44	4.28	4372	23430
Single-referenced training (baseline w/o KD)											
HRED	8.04	150	1 GT	29.00	6.46	39.40	60.80	43.92	3.42	1914	7369
Single-referenced training (baseline tok-KD, §5.2)											
$HRED_{tok\text{-}KD}$	8.04	700	1 GT	27.68	6.90	39.83	62.33	45.11	3.45	1820	7118
Single-referenced training (baseline seq-KD, §5.1)											
HRED	8.04	150	1 hyp	35.08	6.62	39.66	61.96	44.75	3.61	1914	7369
Multi-referenced training (proposed seq-KD, §5.1)											
HRED	8.04	150	5 hyp	23.10	7.13	40.23	62.43	45.44	3.82	1788	7267
			20 hyp	21.15	**7.38**	**40.52**	**62.53**	**45.64**	3.87	1707	6945
			100 hyp	**20.93**	7.28	40.26	62.22	45.30	**3.89**	1704	6794

Table 1: Experimental results of data enhancement. **Param** shows the number of model parameters in M (2^{20}); **Trn Time** shows the approximate time of training on 1 GT data for 1 epoch; **GT** – ground truth; **hyp** – hypotheses; **ppl** – perplexity; **Ext** – Embedding Extrema; **Avg** – Embedding Average; **Grd** – Embedding Greedy; **Reval** – RoBERTa-eval score; **D1** – the number of generated unigram types in the entire test data; **D2** – the number of generated bigram types in the entire test data.

is increased from 20 to 100, the performance gain is limited. This suggests that as training data increases, the model's capacity might have become a bottleneck.

5.2 Comparing with Knowledge Distillation

The proposed data enhancement can be considered as a multi-sequence sequence-level knowledge distillation (seq-KD), and it has been shown to significantly outperform single-sequence seq-KD (i.e. the 1 hyp setting). We would also like to compare it with token-level KD (tok-KD), where the student HRED learns to match its softmax output with the teacher GPT2 on every token (Kim and Rush, 2016). The model is referred to as $HRED_{tok\text{-}KD}$.

While tok-KD outperforms single-sequence seq-KD in some metrics according to Table 1, the proposed multi-sequence seq-KD is much better than tok-KD in all metrics. Other drawbacks of tok-KD include: 1) It requires the student model to have the same vocabulary as the teacher model; 2) The teacher model has to predict the probability distribution for every output token and thus makes the training extremely slow.

6 Proposal: Enhancing Model for Multi-Referenced Training

We have previously seen the HRED's performance gain when we increase the number of hypotheses from 1 to 20, but it starts to degrade when we further increase the number to 100. A conjecture is that the model's capacity is insufficient to learn too complex input-output relations.

6.1 Larger-Sized Model

The simplest way to increase a model's capacity is to use more hidden units and layers. Since the baseline HRED has 1 hidden layer with 500 hidden units, we experimented with larger HREDs, which are 1) $HRED_{l}$ with 2 layers and 1000 hidden units per layer and 2) $HRED_{xl}$ with 2 layers and 2000 hidden units per layer. As shown in Table 2, $HRED_{l}$ slightly outperforms the original HRED but a larger $HRED_{l}$ yields worse results in some metrics. It suggests that increasing model size is not a consistent way to improve performance.

6.2 Variational Model

VHRED and $VHRED_{gmm}$ have the potential to learn one-to-many relations better since they can generate different output sequences by sampling different values from its encoding distributions. However, their performance is not even comparable with the baseline HRED according to Table 2. We also found the performance of VHRED and $VHRED_{gmm5}$ with larger latent variable size and more components to be worse, which is partially due to the fact that their KLD losses are positively correlated with the latent variable size and thus are unbalanced with their reconstruction losses. These

Model	Param (in M)	Trn Time (in sec.)	Data	ppl	BLEU-2	Embedding Similarity Ext	Avg	Grd	Reval	D1	D2
Teacher model											
GPT2$_{md}$	338.39	3000	1 GT	21.16	8.67	41.02	65.17	48.44	4.28	4372	23430
Baseline model											
HRED	8.04	150	100 hyp	20.93	7.28	40.26	62.22	45.30	3.89	1704	6794
Baseline larger model (§6.1)											
HRED$_{l}$	21.04	170	100 hyp	20.81	7.36	40.66	62.53	45.48	3.90	1734	7032
HRED$_{xl}$	52.52	190	100 hyp	**20.69**	7.21	40.43	62.51	45.65	3.85	1743	6986
Baseline variational model (§6.2)											
VHRED	11.02	160	100 hyp	56.54	5.39	38.49	62.38	44.59	3.25	2124	10903
VHRED$_{gmm5}$	11.36	160	100 hyp	50.44	5.44	38.77	62.55	44.79	3.33	2058	10879
Proposed variational model (§6.3)											
VHRED$_{lgm5}$	11.36	160	1 GT	39.97	6.10	40.30	64.03	45.92	3.33	1934	8789
			1 hyp	50.44	6.12	40.26	64.17	46.05	3.50	1989	9427
			5 hyp	30.85	6.61	41.31	65.31	47.19	3.73	1825	8522
			20 hyp	29.74	6.82	41.33	65.29	47.39	3.76	1786	8395
			100 hyp	28.76	6.79	41.31	65.18	47.19	3.76	1777	8364
VHRED$_{lgm20}$	12.52	160	1 GT	46.46	6.70	41.12	64.98	46.83	3.64	1907	8941
			1 hyp	46.45	6.65	41.10	64.95	46.77	3.64	1895	8869
			5 hyp	29.18	6.99	41.80	65.72	47.68	3.82	1725	7757
			20 hyp	26.93	7.07	42.29	66.13	48.01	3.86	1604	7255
			100 hyp	26.40	7.31	**42.31**	**66.32**	**48.32**	3.91	1677	7641
VHRED$_{lgm100}$	18.67	160	100 hyp	26.25	**7.39**	42.28	66.19	48.16	**3.92**	1612	7302
Prior works (§6.4)											
MHRED	8.51	300	100 hyp	24.27	6.59	39.65	61.64	44.79	3.80	1829	7729
HRED$_{CVaR}$	8.04	150	100 hyp	20.92	7.32	40.49	62.43	45.53	3.88	1738	6908
VHRED$_{MBoW}$	11.02	900	100 hyp	51.74	5.68	38.71	62.81	45.07	3.41	2334	12116

Table 2: Experimental results of model enhancement.

results suggest that existing variational baselines are not expressive enough and difficult to optimize.

6.3 VHRED with Linear Gaussian Model (LGM) Prior

To allow for stronger expressiveness, we propose a linear Gaussian model (LGM) prior. Instead of relying on a single Gaussian latent variable, we exploit K Gaussian latent variables $\mathbf{z}_1$ to $\mathbf{z}_K$ and use their linear combination to encode a dialogue:

$$\begin{aligned} \boldsymbol{\mu}_k, \boldsymbol{\sigma}_k, \pi_k &= \text{MLP}_{\theta,k}(\mathbf{c}) \\ \mathbf{z}_k &\sim \text{Gaussian}(\boldsymbol{\mu}_k, \boldsymbol{\sigma}_k^2 \boldsymbol{I}) \\ \mathbf{z} &= \textstyle\sum_{k=1}^{K} \pi_k \mathbf{z}_k, \end{aligned}$$

and we refer to the VHRED with K-variable LGM prior as VHRED$_{lgmK}$.

This simple modification significantly improves VHRED's performance according to results in Table 2. We experimented with K in $\{5, 20, 100\}$ and found the performance improvement to be consistent with more hypotheses and larger K.

Regarding how the interaction between a model's expressiveness (i.e. K) and the amount of hypotheses affects model performance, we notice that:

- When K is small (K = 5), we can hardly obtain performance gain by training with more hypotheses (from 20 to 100).
- When we increase K to 20, further performance gain is achievable. It suggests that the performance bottleneck can be widened to allow for learning from more hypotheses.
- When we increase K to 100, the performance gap between VHRED$_{lgm20}$ and VHRED$_{lgm100}$ is very small. It suggests that we may need more hypotheses to exploit the expressiveness of VHRED$_{lgm100}$.

6.4 Comparing with Prior Works

Three models from prior works are also used for comparison in Table 2, including the mechanism-

Model	Human Scores (in %)		
	Sensible	Specific	SSA
Trained on 1-GT data			
HRED	59.50	60.00	59.75
VHRED$_{gmm5}$	38.50	56.00	47.25
VHRED$_{lgm20}$	52.50	63.50	58.00
Trained on 100-hypotheses data			
HRED	68.50	67.00	67.75
VHRED$_{gmm5}$	44.50	66.50	55.50
VHRED$_{lgm20}$	**72.50**	**74.00**	**73.25**

Table 3: Results of human evaluation on 3 models trained on 2 types of data.

aware model (MHRED, Zhou et al., 2017), the conditional value-at-risk model designed for learning different dialogue scenarios (HRED$_{CVaR}$, Zhang et al., 2018), and the two-step variational model (VHRED$_{MBoW}$, Qiu et al., 2019). Their details have been discussed in Section 3.3.

For the VHRED$_{MBoW}$ model, We only implemented the second step (multiple BoW loss part) because the paper has not provided sufficient details for implementing its first step, and the reported results suggest that the model still works well without the first step processing (Qiu et al., 2019).

As shown in Table 2, these models are not competitive in the multi-referenced setting, and two of them cannot even beat the baseline HRED.

7 Human Evaluation

Besides automated evaluation, we also conduct human evaluation to provide a more accurate assessment of model performance. We sample 100 dialogues randomly from the test data and generate responses using 3 models (HRED, VHRED$_{gmm5}$, VHRED$_{lgm5}$) trained on 2 types of data (the 1-GT data and the 100-hypotheses data). We ask 4 Amazon MTurk human workers to annotate the sensibleness and the specificity of the 600 $(context, response)$ pairs. The collected data reach good inter-rater agreement (Krippendorff's $\alpha > 0.6$). Then we calculate the average of the two metrics (SSA, Adiwardana et al., 2020) as introduced in Section 4.3.

The results of the human evaluation are given in Table 3. First, all three models obtain significant improvements on all three metrics by training on the multi-referenced data, which confirms the effectiveness of the proposed data enhancement method. Then, VHRED$_{lgm20}$ is better than its GMM counterpart and the HRED. And a larger performance gain is obtained for VHRED$_{lgm20}$ than other models when we train it on the multi-referenced data. The result suggests that an expressive prior is indeed necessary and useful for latent dialogue models, especially in the multi-referenced setting.

8 Analysis

8.1 Combining Ground Truth and Hypotheses

One issue that readers may be concerned about is whether it is better to combine ground truth with hypotheses than to use them separately. We take the VHRED$_{lgm20}$ as an example and conduct experiments using mixed training data. As shown in Table 4, we can get performance gain by training with mixed data. The improvement is larger when the original data is smaller (1 hypothesis) because it doubles the training data. When using 100 hypotheses, we can almost fully rely on the generated data and discard ground truth.

8.2 What do variables in LGM learn?

We combine latent variables linearly in the LGM prior. To investigate how each variable contributes, we train a standard VHRED$_{lgm20}$ on the 100-hypotheses data, but evaluate it by using only 1 variable to generate responses. Besides the metrics introduced above, we calculate the average selection probability π_k on the test data (as denoted by $\bar{\pi}_k$). Out of the results, we find four obvious patterns regarding their selection probability (avg prob.), perplexity (PPL), and RoBERTa-eval scores (Reval.). The results of these patterns are shown in Table 5.

In general, selection probability correlates positively with RoBERTa-eval score, while perplexity is less relevant to the other two metrics. For variables that have high probabilities and RoBERTa-eval scores (e.g. the 8th and the 1st), there is a performance discrepancy on other metrics, and thus we believe LGM can capture different aspects of responses. For instance, we notice that the 1st variable tends to generate generic and safe responses, while the 8th variable is likely to produce sentences with more diverse word types. A dialogue example is given in Table 6.[6] A more comprehensive inter-

[6]More examples and results can be found in the Appendix.

Use GT	# hyp.	ppl	BLEU-2	Embedding Similarity			Reval
				Ext	Avg	Grd	
✗	1	46.45	6.65	41.10	64.95	46.77	3.64
✓	1	30.12	6.70	41.48	65.01	46.91	3.71
✗	5	29.18	6.99	41.80	65.72	47.68	3.82
✓	5	27.31	7.26	42.21	66.33	48.32	3.83
✗	20	26.93	7.07	42.29	66.13	48.01	3.86
✓	20	26.46	7.25	42.00	65.81	47.71	3.88
✗	100	26.40	7.31	42.31	66.32	48.32	3.91
✓	100	26.49	7.23	42.28	65.83	47.60	3.88

Table 4: Experimental results of combining ground truth and hypotheses. (§8.1)

k	$\bar{\pi}_k$	ppl	BLEU-2	Reval
Bad prob. / bad PPL / bad Reval.				
4	0.12%	4865.8	1.77	1.51
Bad prob. / good PPL / bad Reval.				
0	0.38%	112.10	5.42	2.73
Medium prob. / bad PPL / good Reval.				
8	8.22%	2740.2	6.22	3.74
Good prob. / good PPL / good Reval.				
1	39.24%	72.34	5.52	3.59

Table 5: Experimental results of VHRED$_{lgm20}$ decoding with the k-th latent variable. (§8.2)

Dialogue Example #422	
Floor	**Context Utterance**
A	*i'm so hungry. shall we go eat now, rick?*
B	*sure. where do you want to go? are you in the mood for anything in particular?*
A	*how about some dumplings? i just can't get enough of them.*
B	*[to be predicted]*
k	**Response Utterance**
4	*tables tables tables there any any any any pale, medium rare.*
0	*ok. i don't think we have any soup at the moment.*
8	*i've heard that some dumplings are really good. but i don't know what to eat.*
1	*ok. i'll go to the restaurant.*

Table 6: Samples of VHRED$_{lgm20}$ decoding with the k-th latent variable. (§8.2)

pretation of the variables remains challenging, and we leave this to future works.

9 Conclusion

In this work, we analyzed the training objective of dialogue response generation models from the view of distribution distance as measured by Kullback–Leibler divergence. The analysis showed that single-referenced dialogue data cannot characterize the one-to-many feature of open-domain dialogues and that multi-referenced training is necessary. Towards multi-referenced training, we first proposed to enhance the training data by replacing every single reference with multiple hypotheses generated by a finetuned GPT2, which provided us with a better approximation of the real data distribution. Secondly, we proposed to equip variational dialogue models with an expressive prior, named linear Gaussian model (LGM), to capture the one-to-many relations. The automated and human evaluation confirmed the effectiveness of the proposed methods.

References

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

Samuel Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *SIGNLL 2016, The 20th SIGNLL Conference on*

Computational Natural Language Learning, pages 10–21.

Chaotao Chen, Jinhua Peng, Fan Wang, Jun Xu, and Hua Wu. 2019. Generating multiple diverse responses with multi-mapping and posterior mapping selection. In *IJCAI 2019, The 28th International Joint Conference on Artificial Intelligence*, pages 4918–4924.

Richárd Csáky, Patrik Purgai, and Gábor Recski. 2019. Improving neural conversational models with entropy-based data filtering. In *ACL 2019, The 57th Annual Meeting of the Association for Computational Linguistics*, pages 5650–5669.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL HLT 2019, The 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186.

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A Smith. 2016. Recurrent neural network grammars. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 199–209.

Jun Gao, Wei Bi, Xiaojiang Liu, Junhui Li, and Shuming Shi. 2019. Generating multiple diverse responses for short-text conversation. In *AAAI 2019, The 33rd AAAI Conference on Artificial Intelligence*, volume 33, pages 6383–6390.

Xiaodong Gu, Kyunghyun Cho, Jung Woo Ha, and Sunghun Kim. 2019. DialogWAE: Multimodal response generation with conditional wasserstein auto-encoder. In *ICLR 2019, The 7th International Conference on Learning Representations*.

Ari Holtzman, Jan Buys, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. In *ICLR 2020, The 8th International Conference on Learning Representations*.

Zhiting Hu, Bowen Tan, Russ R Salakhutdinov, Tom M Mitchell, and Eric P Xing. 2019. Learning data manipulation for augmentation and weighting. In *NeurIPS 2019, Advances in Neural Information Processing Systems 32*, pages 15738–15749.

Yoon Kim and Alexander M Rush. 2016. Sequence-level knowledge distillation. In *EMNLP 2016, The 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR 2015, The 3rd International Conference on Learning Representations*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *ACL 2007, The 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180.

Klaus Krippendorff. 2018. *Content analysis: An introduction to its methodology*. Sage publications.

Adhiguna Kuncoro, Chris Dyer, Laura Rimell, Stephen Clark, and Phil Blunsom. 2019. Scalable syntax-aware language models using knowledge distillation. In *ACL 2019, The 57th Annual Meeting of the Association for Computational Linguistics*, pages 3472–3484.

Christophe Leys, Christophe Ley, Olivier Klein, Philippe Bernard, and Laurent Licata. 2013. Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. *Journal of Experimental Social Psychology*, 49(4):764–766.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. DailyDialog: A manually labelled multi-turn dialogue dataset. In *IJCNLP 2017, The 8th International Joint Conference on Natural Language Processing*, volume 1, pages 986–995.

Tong Niu and Mohit Bansal. 2019. Automatically learning data augmentation policies for dialogue tasks. In *EMNLP-IJCNLP 2019, The 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 1317–1323.

Shuke Peng, Xinjing Huang, Zehao Lin, Feng Ji, Haiqing Chen, and Yin Zhang. 2019. Teacher-student framework enhanced multi-domain dialogue generation. *arXiv preprint, arXiv:1908.07137*.

Lisong Qiu, Juntao Li, Wei Bi, Dongyan Zhao, and Rui Yan. 2019. Are training samples correlated? learning to generate dialogue responses with multiple references. In *ACL 2019, The 57th Annual Meeting of the Association for Computational Linguistics*, pages 3826–3835.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).

Alan Ritter, Colin Cherry, and William B Dolan. 2011. Data-driven response generation in social media. In *EMNLP 2011, The 2015 Conference on Empirical Methods in Natural Language Processing*, pages 583–593.

Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *AAAI 2016, The 30th AAAI Conference on Artificial Intelligence*, volume 16, pages 3776–3784.

Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron C. Courville, and Yoshua Bengio. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. In *AAAI 2017, The 31st AAAI Conference on Artificial Intelligence*, pages 3295–3301.

Mingyue Shang, Zhenxin Fu, Nanyun Peng, Yansong Feng, Dongyan Zhao, and Rui Yan. 2018. Learning to converse with noisy data: generation with calibration. In *IJCAI 2018, The 27th International Joint Conference on Artificial Intelligence*, pages 4338–4344.

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

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *NIPS 2014, Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014*, pages 3104–3112.

Xu Tan, Yi Ren, Di He, Tao Qin, and Tie-Yan Liu. 2019. Multilingual neural machine translation with knowledge distillation. In *ICLR 2019, The 7th International Conference on Learning Representations*.

Oriol Vinyals and Quoc Le. 2015. A neural conversational model. In *ICML Deep Learning Workshop*.

Thomas Wolf, Victor Sanh, Julien Chaumond, and Clement Delangue. 2019. TransferTransfo: A transfer learning approach for neural network based conversational agents. *arXiv preprint, arXiv:1901.08149*.

Hainan Zhang, Yanyan Lan, Jiafeng Guo, Jun Xu, and Xueqi Cheng. 2018. Tailored sequence to sequence models to different conversation scenarios. In *ACL 2018, The 56th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 1479–1488.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2019. DialoGPT: Large-scale generative pre-training for conversational response generation. *arXiv preprint, arXiv:1911.00536*.

Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. 2017. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *ACL 2017, The 55th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 654–664.

Tianyu Zhao, Divesh Lala, and Tatsuya Kawahara. 2020. Designing precise and robust dialogue response evaluators. In *ACL 2020, The 58th Annual Meeting of the Association for Computational Linguistics*, pages 26–33.

Renjie Zheng, Mingbo Ma, and Liang Huang. 2018. Multi-reference training with pseudo-references for neural translation and text generation. In *EMNLP 2018, The 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3188–3197.

Ganbin Zhou, Ping Luo, Rongyu Cao, Fen Lin, Bo Chen, and Qing He. 2017. Mechanism-aware neural machine for dialogue response generation. In *AAAI 2017, The 31st AAAI Conference on Artificial Intelligence*.

A Human Evaluation

We recieved 2400 annotations in total (4 annotators for each of the 600 $(context, response)$ pairs). We first remove annotation outliers following Leys et al. (2013). After removing 208 annotations for sensibleness and 253 for specificity, the remaining annotations have reasonable inter-rater agreement meansured by Krippendorff's α (Krippendorff, 2018) as shown in Table 7.

Item	Krippendorff's α
Sensibleness	0.76
Specificity	0.60

Table 7: Inter-rater agreement of human annotations.

Item	Statistics		
	Train	*Validation*	*Test*
sessions	9237	1157	1159
$(ctx, resp)$ pairs	59305	9906	9716

Table 8: Corpus statistics.

Training Data	Max Epochs	Max Steps
1 GT	100	5.93M
1 hyp.	100	5.93M
1 GT + 1 hyp.	50	5.93M
5 hyp.	20	5.93M
1 GT + 5 hyp.	20	7.12M
20 hyp.	10	11.86M
1 GT + 20 hyp.	10	12.45M
100 hyp.	2	11.86M
1 GT + 100 hyp.	2	11.98M

Table 9: Maximum training epochs and steps in different data settings.

B Experimental Settings

B.1 Model Implementation

For HRED and VHRED models, we implement encoders and decoders with gated recurrent unit (GRU) networks. Sentence-level encoders are bidirectional, while dialogue-level encoders and decoders are unidirectional. All the GRU networks have 1 layer and 500 hidden units. We use 30-dimensional floor embeddings to encode the switch of floor. For VHREDs, latent variables have 200 dimensions. Prior and posterior networks are implemented by feedforward networks with hyperbolic tangent activation function. While priors have different forms (unimodal Gaussian, Gaussian mixture model, and linear Gaussian model), we use unimodal Gaussian for all the posteriors. We use attentional mechanism for all decoders. All models were trained on a single `NVIDIA TITAN RTX` card. When training on K-hypotheses data, the training time per epoch is roughly K times of the reported number.

B.2 Training Details

We optimize all the models with the Adam method (Kingma and Ba, 2015). The initial learning rate is 0.001 and gradients are clipped within [-1.0, 1.0]. We decay the learning rate with decay rate 0.75 and patience 3. The training process is early stopped when the learning rate is less than 1×10^{-7}. The numbers of training epochs and steps are shown in Table 9. Batch size is 30 during training. We use up to 5 history utterances as context, and all utterances are truncated to have 40 tokens to most. We set dropout probability as 0.2 and shuffle training data every epoch for better generalization. VHREDs are optimized by maximizing their variational lower bound (Sohn et al., 2015). We apply linear KL annealing in the first 40,000 training steps.

For finetuning the GPT2 model, we use a smaller batch size of 10 to fit the model into memory. As with other hyperparameters such as learning rate and weight regularization factor, we follow the settings used by Wolf et al. (2019). And the GPT2 is finetuned on the 1-GT data for only 2 epochs.

C Extra Samples

We show three more dialogue samples in Table 10 and 11.

Dialogue #100	
Floor	**Context Utterance**
A	*i have never shopped there, but i think price club is a kind of membership store.*
B	*what does that mean?*
A	*that means you have to pay a membership fee to shop there. i have heard they have very good prices on electronics.*
B	*electronics?*
A	*yes. like stereos and [UNK].*
B	[TO BE PREDICTED]

Model	**Trn. Data**	**Response Utterance**
Ground truth	-	*how much is the membership fee?*
HRED	1 GT	*do you have any other advantages?*
VHRED$_{gmm5}$	1 GT	*how do you want to buy?*
VHRED$_{lgm20}$	1 GT	*so what? i don't have a current account.*
HRED	100 hyp.	*i see. i'll buy a new one.*
VHRED$_{gmm5}$	100 hyp.	*that's a good idea. how about the one?*
VHRED$_{lgm20}$	100 hyp.	*i see. is there a warranty?*
- the 4th variable	100 hyp.	*sure. which service do you use?*
- the 0th variable	100 hyp.	*how much money can you get it?*
- the 8th variable	100 hyp.	*i have a membership card. it's very expensive.*
- the 1st variable	100 hyp.	*i see. it's a good deal.*

Table 10: More samples on the test data. (§C)

Dialogue #500	
Floor	**Context Utterance**
A	*i am not sleepy.*
B	*it's getting late, and you have to wake up early tomorrow.*
A	*there is no way that i can fall asleep right now.*
B	*try listening to some soft music.*
A	*it won't work. i'm nowhere close to being tired.*
B	[TO BE PREDICTED]

Model	**Trn. Data**	**Response Utterance**
Ground truth	-	*i really don't care, just go to sleep.*
HRED	1 GT	*you're too tired.*
VHRED$_{gmm5}$	1 GT	*what's up?*
VHRED$_{lgm20}$	1 GT	*you shouldn't have to sleep.*
HRED	100 hyp.	*don't worry. i'll get you up.*
VHRED$_{gmm5}$	100 hyp.	*i don't know. i just want to relax.*
VHRED$_{lgm20}$	100 hyp.	*you should be tired. you can get a good sleep.*
- the 4th variable	100 hyp.	*do do let you gift you gift you gift you live you live here i sing for here friendship akimbo?*
- the 0th variable	100 hyp.	*don't be at the evening.*
- the 8th variable	100 hyp.	*you are always sleepy.*
- the 1st variable	100 hyp.	*come on. you can get a good sleep.*

Table 11: More samples on the test data. (§C)