Learning to Express in Knowledge-Grounded Conversation

Xueliang Zhao[1,2], Tingchen Fu[3], Chongyang Tao[4], Wei Wu[5], Dongyan Zhao[1,2][*]**, Rui Yan[3]**[*]

[1]Wangxuan Institute of Computer Technology, Peking University
[2]Center for Data Science, AAIS, Peking University
[3]Gaoling School of Artificial Intelligence, Renmin University of China
[4]Microsoft   [5]Meituan, Beijing, China
{xl.zhao,zhaody}@pku.edu.cn   ruiyan@ruc.edu.cn
{lucas.futingchen,chongyangtao,wuwei19850318}@gmail.com

Abstract

Grounding dialogue generation by extra knowledge has shown great potentials towards building a system capable of replying with knowledgeable and engaging responses. Existing studies focus on how to synthesize a response with proper knowledge, yet neglect that the same knowledge could be expressed differently by speakers even under the same context. In this work, we mainly consider two aspects of knowledge expression, namely the structure of the response and style of the content in each part. We therefore introduce two sequential latent variables to represent the structure and the content style respectively. We propose a segmentation-based generation model and optimize the model by a variational approach to discover the underlying pattern of knowledge expression in a response. Evaluation results on two benchmarks indicate that our model can learn the structure style defined by a few examples and generate responses in desired content style.

1 Introduction

Building an open domain dialogue system has attracted increasing attention from the community of AI and NLP. Despite the impressive progress, existing models are notorious for replying with generic and bland responses. To bridge the gap, researchers resort to ground dialogue generation by extra knowledge such as unstructured documents (Zhou et al., 2018c; Dinan et al., 2019). By this means, the documents serve as content sources and make a dialogue system knowledgeable regarding various concepts in a discussion.

However, existing studies focus on how to synthesize a response with proper knowledge (Dinan et al., 2019; Kim et al., 2020; Zhao et al., 2020b), but pay little attention to the fact that the same knowledge could be expressed differently even under the same context. These models usually em-

---

*Corresponding authors: Dongyan Zhao and Rui Yan.

Knowledge
• MovieName: Frozen
• Year: 2013
• Rating: Rotten Tomatoes: 89%, Metacritics: 74/100, CinemaScore: A+
• Cast: Kristen Bell as Anna, the 18-year-old Princess of Arendelle and Elsa's younger sister, Livvy Stubenrauch as 5-year-old Anna, Katie Lopez as 5-year-old Anna (singing), Agatha Lee Monn as 9-year-old Anna …
• …

	Conversations	
User1:	I was really surprised that disney chose Kristen Bell to be the voice of Anna in Frozen	I was really surprised that disney chose Kristen Bell to be the voice of Anna in Frozen
User2:	Yes, I didn't imagine it'd be her!	Yes, I didn't imagine it'd be her!
User2:	What do you think about the rating?	What do you think about the rating?
User1:	74 in Metacritics. I believe it deserves, indeed.	The rating is 74 in Metacritics. Let me say, high enough for a Disney move
User1:	And I'd give credit to three different voice actors for anna. I'm really impressed. What about you?	And I do think it was overkill to use three different voice actors for anna. Do you agree?

Table 1: A case from CMU_DoG. Given the same knowledge and context, the last two turns in left and right conversations exhibit positive and negative sentiments, respectively. Each utterance can be decomposed into knowledge-related and knowledge-irrelevant segments.

ploy a regular decoder to generate the response in an auto-regressive manner given the contextual representations of knowledge and dialogue context, which makes the generation process less explainable and controllable.

In general, the expression style of response is composed of two aspects: the structure of the response and the style of the content in each part. As the example shown in Table 1, knowledge-related phrases and clauses tend to be long, like "And I'd give credit to three different voice actors for anna.", or short, like "74 in Metacritics". Besides, they may appear at the beginning of the sentence, or at the end. For the sake of description, we decompose a response into a sequence of non-overlapping segments, each is either related to certain background knowledge and diverse in content style, or almost irrelevant to the knowledge but simply playing the role of stitching the context and carrying on the conversation. We therefore define the structure style as the distribution and number of two kinds of segments. The structure style itself is far from dominant in the sentence expression, since different speakers could convey converse attitude even if the

2258

context and the knowledge are exactly the same. So it is necessary to introduce the content style as the expression fashion within each knowledge-related segment. We further introduce two latent variables to facilitate end-to-end training, one for predicting the start and end positions of a segment, the other for deciding the category of each segment. Since the human annotations for sentence segmentation are absent and enumerating over all possibilities to maximize the likelihood of the response is time-consuming, we propose a variational framework for segmentation-based generation and induce an evidence lower bound of the likelihood.

Formally, our model follows an encoder-decoder architecture. The encoder is to obtain the contextual representation of conversational context and knowledge in a regular way. The decoder consists of three types of modules: (1) context module, for response only based on context without knowledge; (2) plain-knowledge module, for response referring knowledge but without particular style; and (3) stylized-knowledge module, for response referring knowledge and with a specific style. The context module is the only module not relying on knowledge, but simply paying attention to contextual information. Compared with plain-knowledge module, stylized-knowledge module has unique adapters, which is their primary discrepancy. When decoding, the decoder first predicts the segmentation of the response and then makes a choice in three kinds of modules to generate a single segment. Both the segmentation and the module selection are instructed under sequential latent variables.

We train our model on the Reddit Corpus published by Li et al. (2020) and evaluate our model on two benchmarks of knowledge-grounded conversation: Wizard of Wikipedia (Wizard) (Dinan et al., 2019) and CMU Document Grounded Conversation (CMU_DoG) (Zhou et al., 2018c). Evaluation results indicate that our model can significantly outperform state-of-the-art methods in the zero-resource setting (i.e., only trained on the Reddit Corpus). In addition, the performance of our model improves significantly on Wizard and CMU_DoG with the presence of only 10% training data and the segment distributions after fine-tuning are consistent with our prior knowledge about the two datasets, indicating that our model can learn the structure style with little cost. Finally, our model outperforms previous state-of-the-art models on the accuracy of performing sentiment classification us-

ing generated responses, which indicates that the model can be controlled to express knowledge with the desired content style.

Contributions in this work are three-fold: (1) exploration of the knowledge expression in knowledge-grounded conversation; (2) proposal of a variational segmentation-based generation model to discover the underlying expression style in a response; (3) empirical verification of the effectiveness of the proposed model on two benchmarks of knowledge-grounded conversation.

2   Related Work

On the vanilla encoder-decoder architecture (Shang et al., 2015; Vinyals and Le, 2015), various extensions have been made to model the structure of dialogue contexts (Serban et al., 2016, 2017; Zhang et al., 2019a); to improve diversity of responses (Li et al., 2015; Xing et al., 2017; Zhao et al., 2017; Tao et al., 2018); to control attributes of responses (Xu et al., 2019; Zhou et al., 2018a; Wang et al., 2018; See et al., 2019); and to bias responses to some specific personas (Li et al., 2016; Zhang et al., 2018). Recently, grounding dialogue generation by extra knowledge has seemed promising to bridge the gap between conversation with existing systems and conversation with humans, and the knowledge could be obtained from knowledge graphs (Zhou et al., 2018b; Moon et al., 2019; Tuan et al., 2019), retrieved from unstructured documents (Dinan et al., 2019; Lian et al., 2019; Zhao et al., 2019, 2020a; Kim et al., 2020; Li et al., 2020; Fu et al., 2022) or visual background (Mostafazadeh et al., 2017; Shuster et al., 2018; Huber et al., 2018). In this work, we study document-grounded dialogue generation. Rather than selecting knowledge relevant to dialogue context and directly exploiting pre-trained language models to generate the response, we focus on expressing knowledge in this task.

The idea of sequence modeling via segmentation (Wang et al., 2017; Kim et al., 2019) has attracted widespread attention in several natural language processing tasks. In text segmentation task, Wang et al. (2017) propose a probabilistic model for sequence modeling via their segmentation and a "Sleep-WAke Network" (SWAN) method. In machine translation, Huang et al. (2017) propose a neural phrase-based machine translation system that models phrase structures in the target language using SWAN. In data-to-text generation, Wiseman et al. (2018) develop a neural template-like genera-

tion model with a HSMM decoder, which is learned tractably by backpropagating through a dynamic program; to tackle the problem of weak Markov assumption for the segment transition probability, Shen et al. (2020) propose to explicitly segment target text into fragments and align them with their data correspondences, and jointly learn the segmentation and correspondence via dynamic programming.

3 Approach

We start by providing the problem formalization and overview of the proposed model in Sec.3.1. Then in Sec.3.2 we describe the design for each components. Lastly, we elaborate how to optimize the components with variational inference and weak supervision in Sec.3.3.

3.1 Overview

Suppose that we have a dataset $\mathcal{D} = \{(U_i, K_i, R_i)\}_{i=1}^N$, where $\forall i \in \{1, \ldots, N\}$, K_i serves as background knowledge of the dialogue (U_i, R_i) with $K_{i,j}$ being the j-th sentence, U_i is the context of the dialogue with $U_{i,j}$ the j-th utterance, and R_i is the response. To bias the expression to a specific structure style, we further assume that there are a few examples $\mathcal{D}_{sty} = \{(U_i, K_i, R_i)\}_{i=1}^M$ provided by users depicting the required style for knowledge expression. Note that we have $N \gg M$, since corpus in a specific expression style is rare and difficult to acquire. The goal is to learn a generation model $p_\theta(R|U, K)$ (θ denotes the parameters of the model) from $\mathcal{D}$, to generate a response R following $p_\theta(R|U, K)$ given a new dialogue context U and the associated knowledge K. Different from previous KGC generation model, we allow users to either (1) bias the structure style of $P_\theta(R|U, K)$ to $\mathcal{D}_{sty}$ with little cost; or (2) switch the content style of knowledge expression in R.

Figure 1 gives an overview of the proposed model, which is based on the encoder-decoder architecture. The encoder generates the contextual representations of the dialogue and knowledge, while the decoder generates the segments one after another. $\mathbf{h}_t^N$ encodes the dialogue context up to timestep $t-1$ with N denoting the number of decoder layers. Given $R = (r_1, \cdots, r_t, \cdots, r_{l_r})$ with r_t referring the t-th token of R whose length is supposed to be l_r, the variable $Z = \{z_t\}_{t=1}^{l_r}$ is utilized to control the choice of module of each

segment (**Module Indicator**), and its historical information is encoded by $\{\mathbf{c}_t\}_{t=0}^{l_r}$. $M = \{m_t\}_{t=1}^{l_r}$ is a sequence of binary variables and used to determine the boundary of each segment (**Boundary Indicator**). Specifically, $m_t = 1$ indicating that the current segment is already completed and a new segment should be created at the next timestep. Otherwise, $m_t = 0$ and the current segment remains unfinished. The generative process is disassembled into two steps: (1) determine the type of a new segment based on previously generated text and previous segment types; (2) generate within the current segment until the binary variable $m_t = 1$.

3.2 Model Architecture

Context and Knowledge Encoding. We exploit the pre-trained BART (Lewis et al., 2020) as the backbone of our architecture, which is pre-trained using a variety of denoising objectives and achieves state-of-the-art results on a range of text generation tasks. Given the dialogue context $U = (U_1, \cdots, U_n)$, we simply concatenate them as $(u_1, \cdots, u_{l_u})$. Similarly, we concatenate the associated knowledge $K = (K_1, \cdots, K_m)$ as $(k_1, \cdots, k_{l_k})$. l_u and l_k are the length of dialogue context and background knowledge respectively. The input of the encoder is then defined as:

$$I = [\text{BOS}]k_1 \ldots k_{l_k}[\text{EOS}]u_1 \ldots u_{l_u}[\text{EOS}]. \quad (1)$$

The input I is truncated or padded to the maximum capacity and then passes through the stacked self-attention layers and results in a knowledge-aware context representation $\mathbf{C}$, and a context-aware knowledge representation $\mathbf{K}$. Specifically, the context-aware knowledge representation is defined as $\mathbf{K} = [\mathbf{h}_1^{enc}, \cdots, \mathbf{h}_{l_k+1}^{enc}]$ where $\mathbf{h}_t^{enc}$ is the last layer of BART encoder at time t. Similarly, the knowledge-aware context representation is defined as $\mathbf{C} = [\mathbf{h}_{l_k+2}^{enc}, \cdots, \mathbf{h}_{l_k+l_u+2}^{enc}]$.

Prior of Module Indicator. We use the sequential discrete latent variable $Z = \{z_t\}_{t=1}^{l_r}$ to decide which module to invoke at each timestep. The transition of z_t occurs only when a segment is completed, which is decided by the binary boundary variable M. The prior quantifies the distribution of z_t before we observe the segment, and it is reasonable to assume that the prior of z_t depends on previous module choices $z_{<t}$ and previously generated text. Then the transition of Z is defined as:
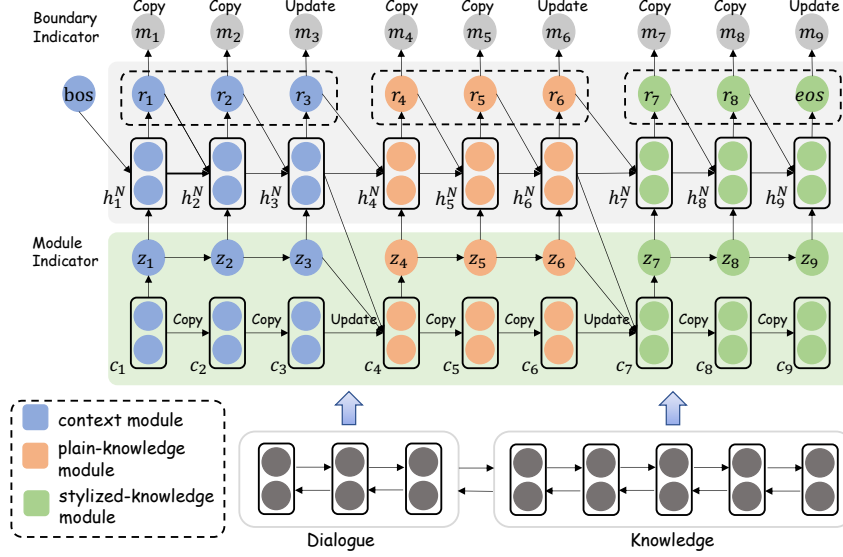
Figure 1: Architecture of the proposed model. "Copy" means $m_t = 0$ and the state of the module indicator c_{t+1} remains unchanged. "Update" indicates that $m_t = 1$ and the state has been updated to include the information from the previous segment.

$$p_{\theta_z}(z_t|r_{<t}, z_{<t}, m_{t-1}) = m_{t-1} \cdot \tilde{p}(z_t|\mathbf{c}_t)$$
$$+ (1 - m_{t-1}) \cdot \delta(z_t = z_{t-1}), \quad (2)$$

where δ is a Kronecker delta function. $\mathbf{c}_t$ encodes all previous latent states $z_{<t}$ and generated text $r_{<t}$ as follows:

$$\mathbf{c}_t = m_{t-1} \cdot f_{z-\text{rnn}}(\tilde{\mathbf{z}}_{t-1}, \mathbf{c}_{t-1}) + (1-m_{t-1}) \cdot \mathbf{c}_{t-1}. \quad (3)$$

$\tilde{\mathbf{z}}_{t-1} = [\mathbf{e}_{t-1}; \mathbf{h}_{t-1}^{N,dec}]$ with $\mathbf{e}_{t-1}$ the embedding of z_{t-1} and $\mathbf{h}_{t-1}^{N,dec}$ the representation of last generated token. Specifically, $m_{t-1} = 0$ means that the next timestep t is still in the same segment as the previous timestep $t - 1$ and thus the latent variable z_t should not be updated. Otherwise, it means that current segment is completed and z_t is updated with the transition function $\tilde{p}(z_t|\mathbf{c}_t)$. Because we only have $N_{sty} + 2$ options when choosing a module, where N_{sty} is the number of different user-defined styles in addition to 2 default styles, so in this model, the latent variable z_t ranges in natural integer to denote corresponding style type. Specifically, $z_t = 0$ denotes choosing the context expression module to generate a knowledge-irrelevant segment; $z_t = 1$ tells the model to choose the knowledge expression module without specially customized style; we leave the $z_t \geq 2$ to be user-defined so as to select the knowledge expression module combined with customized style. The transition function $\tilde{p}(z_t|\mathbf{c}_t)$ is

then implemented as a multinomial distribution parameterized by $\text{Softmax}(f_{z-\text{mlp}}(\mathbf{c}_t))$[1].

Prior of Boundary Indicator. The boundary indicator $M = \{m_t\}_{t=1}^{l_r}$ depicts the segmental structure of the response, with $m_t = 1$ indicates that a new segment should start at time $t + 1$. Presumably, the prior of m_t could be inferred from $r_{\leq t}$ and z_t. We model the distribution $p_{\theta_m}(m_t|r_{\leq t}, z_t)$ by a Bernoulli distribution parameterized by $\sigma(f_{m-\text{mlp}}([\mathbf{e}_t; \mathbf{h}_t^{N,dec}]))$, where σ denotes the sigmoid function.

Stylized Generation. As mentioned above, the generation process involves scheduling different modules according to z_t. Here we give a systematic description of the generation process. The decoder accepts the token generated last timestep r_{t-1} as input, performs transformation in N decoder layers, finally obtains a dense representation.

We use $\mathbf{h}_t^l$ to denote the hidden state after the l-th layer at timestep t, which is a shorthand for $\mathbf{h}_t^{l,dec}$ for brevity. Specially, $\mathbf{h}_t^0$ is the output of the embedding layer. When $z_t = 0$, it implies that knowledge encoding is unnecessary for current segment so $\mathbf{h}_t^l$ is defined as:

$$\mathbf{h}_t^l = \text{DecoderLayer}(\mathbf{h}_t^{l-1}, \mathbf{H}_{t-1}^{l-1}, \mathbf{C}), \quad (4)$$

where $\mathbf{H}_{t-1}^l = [\mathbf{h}_1^l, \cdots, \mathbf{h}_{t-1}^l]$ is a sequence of decoder hidden states in previous timesteps, and

---

[1] We use $f_{*-\text{mlp}}$ to denote a multi-layer perceptron network in this paper.

C is the context representation mentioned above. DecoderLayer$(\cdot, \cdot, \cdot)$ is implemented as pre-trained BART decoder layer where $\mathbf{h}_t^{l-1}$ first plays self-attention on $\mathbf{H}_{t-1}^{l-1}$ then performs cross-attention on **C**. The probability $p(r_t|r_{<t}, z_t = 0)$ is defined as a multinomial distribution parameterized by Softmax$(f_{r-\mathrm{mlp}}(\mathbf{h}_t^N))$, where $\mathbf{h}_t^N$ encodes the generated tokens up to timestep $t-1$. When $z_t = 1$, the implementation of decoder layer is analogous to the $z_t = 0$ case except that we replace **C** with **K**, since knowledge is needed:

$$\mathbf{h}_t^l = \mathrm{DecoderLayer}(\mathbf{h}_t^{l-1}, \mathbf{H}_{t-1}^{l-1}, \mathbf{K}). \quad (5)$$

To generate a segment with a particular customized style when $z_t \geq 2$, we introduce some adapters to bias the generation efficiently following Houlsby et al. (2019). Specifically, the hidden state $\mathbf{h}_t^l$ is defined as:

$$\mathbf{h}_t^l = \mathrm{DecoderLayer}_{adp}(\mathbf{h}_t^{l-1}, \mathbf{H}_{t-1}^{l-1}, \mathbf{K}), \quad (6)$$

where DecoderLayer$_{adp}(\cdot, \cdot, \cdot)$ denotes the transformer decoder layer with adapters inserted. To make the style fine-grained and adjustable, each style has a unique set of adapters. Different styles have no adapter in common. In addition, our model has the ability to learn to express in any style, as long as a discriminator for the desired style is provided.[2]

3.3 Learning Details

We introduce auxiliary distributions $q_{\phi_m}(M|R) = \prod_{t=1}^{l_r} q_{\phi_m}(m_t|R)$ and $q_{\phi_z}(Z|M, R) = \prod_{t=1}^{l_r} q_{\phi_z}(z_t|M, R)$, which serve as an approximation to the intractable posterior of the boundary indicator M and the module indicator Z. We then apply variational approximation which gives the following evidence lower bound objective [3](ELBO) (Hoffman et al., 2013):

$$\log p_\theta(R|U, K)$$
$$\geq \mathbb{E}_{q_{\phi_m}(M|R)} \left(\mathbb{E}_{q_{\phi_z}(Z|M,R)} \sum_{t=1}^{l_r} \log p_\theta(r_t|r_{<t}, z_t) \right.$$
$$\left. - \sum_{t=1}^{l_r} m_{t-1} \cdot D_{\mathrm{KL}}\big(q_{\phi_z}(z_t|M,R) \| p_{\theta_z}(z_t)\big) \right) \quad (7)$$
$$- \sum_{t=1}^{l_r} D_{\mathrm{KL}}\big(q_{\phi_m}(m_t|R) \| p_{\theta_m}(m_t)\big),$$

---

[2]The proposed method is also able to control the content style of knowledge-irrelevant segmentation by introducing extra adapters. But we focus on knowledge expression in this work.

[3]We always have $m_0 = 1$

where $p_{\theta_z}(z_t)$ and $p_{\theta_m}(m_t)$ stand for $p_{\theta_z}(z_t|r_{<t}, z_{<t}, m_{t-1})$ and $p_{\theta_m}(m_t|r_{\leq t}, z_t)$ respectively, and $D_{\mathrm{KL}}(\cdot\|\cdot))$ refers to Kullback–Leibler divergence. Detailed derivations are presented in the appendix.

Based on the intuition that the response provides hints about the segmentation, we construct the posterior distribution $q_{\phi_m}(m_t|R)$ as a Bernoulli distribution parameterized by $\sigma(f'_{m-\mathrm{mlp}}(\psi_t))$. ψ_t is a feature extracted from a bi-directional LSTM $\psi(R)$. Since the module indicator is kept unchanged within a segment, the posterior distribution $q_{\phi_z}(z_t|M, R)$ is conditioned on the boundary indicator m_{t-1} and defined as:

$$q_{\phi_z}(z_t|M, R) = m_{t-1} \cdot \tilde{q}(z_t|\psi_t) + (1 - m_{t-1}) \cdot \delta(z_t = z_{t-1}), \quad (8)$$

where $\delta(\cdot)$ denotes Dirac delta function and the transition function $\tilde{q}(z_t|\psi_t)$ is implemented as a multinomial distribution parameterized by Softmax$(f'_{z-\mathrm{mlp}}(\psi_t))$. Once we have the posterior distribution, we apply Gumbel-Softmax (Jang et al., 2016) to take samples of m_t and z_t.

Weak Supervision on M and Z. We first use StanfordNLP toolkit (Manning et al., 2014) to parse every response in the training set as a sequence of segments, and use $\tilde{M} = \{\tilde{m}_t\}_{t=1}^{l_r}$ to denote the results of segmentation labeling. The pseudo label of module choice $\tilde{Z} = \{z_t\}_{t=1}^{l_r}$ is tagged in a similar way to multiclass classification, determined by (1) the similarity between each segment and knowledge and (2) the classification confidence of the style discriminator. More details about the construction of $\tilde{Z}$ and $\tilde{M}$ are provided in the appendix.

With $\tilde{Z}$ and $\tilde{M}$, the loss function of weak supervision is defined as:

$$\mathcal{L}_m = -\sum_{t=1}^{l_r} \log p_{\theta_m}(\tilde{m}_t|r_{\leq t}, \tilde{z}_t),$$
$$\mathcal{L}_z = -\sum_{t=1}^{l_r} \tilde{m}_{t-1} \cdot \log p_{\theta_z}(\tilde{z}_t|r_{<t}, \tilde{z}_{<t}, \tilde{m}_{t-1}). \quad (9)$$

The learning algorithm is summarized and provided in the appendix.

4 Experiments

4.1 Datasets

We test our model on benchmarks of knowledge-grounded dialogue generation, including Wizard of

Training Data	Models	Wizard Seen				Wizard Unseen				CMU_DoG			
		PPL	F1	D-1	D-2	PPL	F1	D-1	D-2	PPL	F1	D-1	D-2
Reddit Corpus	BART	40.1	18.4	0.076	0.355	42.9	18.4	0.049	0.237	75.8	9.8	0.021	0.131
	ZRKGC	41.1	18.9	0.055	0.246	42.7	18.8	0.037	0.179	53.8	12.2	0.015	0.094
	Our Model	**35.9**	**19.3**	**0.082**	**0.383**	**38.4**	**19.2**	**0.060**	**0.292**	60.4	12.2	**0.028**	**0.186**
Reddit Corpus + 10% annotated data	BART	32.7	18.9	0.073	0.357	35.0	18.8	0.049	0.235	49.5	10.1	0.019	0.110
	ZRKGC	29.1	19.1	0.072	0.309	31.6	18.9	0.048	0.209	38.0	13.7	0.010	0.062
	Our Model	**28.6**	**20.4**	0.073	**0.366**	**30.7**	**20.0**	0.052	**0.270**	40.8	**14.4**	0.015	**0.122**

Table 2: Automatic evaluation results. Numbers in bold mean that the improvement to the best performing baseline is statistically significant (t-test with p-value < 0.05).

Wikipedia (Wizard) (Dinan et al., 2019) and CMU Document Grounded Conversations (CMU_DoG) (Zhou et al., 2018c). We choose the Reddit Corpus published by (Li et al., 2020) as $\mathcal{D}$ for pre-training. Since it is abundant in expression style as a corpus from online forum, the two latent variables could be well initialized. We use part of the training data of Wizard and CMU_DoG as $\mathcal{D}_{sty}$ respectively, for these two datasets are distinctive in expression style and differ from each other. The dialogues in CMU_DoG tend to be causal and short, with most utterances irrelevant to knowledge while the responses in Wizard are usually long and knowledgeable, as some phrases are directly extracted from wiki articles.

More details about the datasets are described in the appendix.

4.2 Experimental Setup

In this paper, we mainly consider two experimental setups, corresponding to the two aspects of knowledge expression. To explore how our model can be used to control the distribution of different kinds of segments (knowledge-related and knowledge-irrelevant), we first train the model on the Reddit Corpus and then fine-tune it on a small amount of examples in Wizard and CMU_DoG, respectively.[4] To verify whether our model can generate the knowledge-related segments in the desired style, we still train the model on the Reddit Corpus, and use a style tag to control the generation process. In this experimental setup, we are primarily concerned with generating with two kinds of styles, positive and negative, where $z_t = 2 \cdot \min(1, z_t)$ tells the model to generate a response in positive sentiment and $z_t = 3 \cdot \min(1, z_t)$ is for response in negative sentiment.

Evaluation Metrics. Following Dinan et al. (2019), we choose PPL and unigram F1 as the

metrics to evaluate the appropriateness. We further use Distinct-1/2 (D-1/2), which are calculated as ratios of distinct unigrams and bigrams in responses respectively, to evaluate the distinctness. We also employ classification accuracy as the evaluation metrics for style control experiments.[5] Due to space limitation, we provide automatic evaluation results on more metrics (i.e., BLEU-1, METEOR, and ROUGE-L) in the appendix.

To further verify whether our model could learn structure style and content style, we randomly sample 300 examples from Test Seen of Wizard, and the test set of CMU_DoG respectively, and recruit 6 well-educated native speakers to do qualitative analysis on the responses generated by our model and all baselines. The annotators need to judge the quality of the responses from four aspects (i.e., *fluency*, *context coherence*, *knowledge relevance* and *style consistency*), and assign a score from $\{0, 1, 2\}$ (representing "bad", "fair" and "good" respectively) to each response for each aspect. The agreement among all annotators is measured via Fleiss' kappa (Fleiss, 1971). More details about the setup of human evaluation as well as the results on learning content style are provided in the appendix.

4.3 Baselines

For the exploration of structure style, we select the following models as baselines: (1) *BART* (Lewis et al., 2020): a model that achieves state-of-the-art performance on various text generation tasks. Note that our model degrades into BART once we remove the module indicator Z and the boundary indicator M; (2) *Zero-resource Knowledge-grounded Conversation (ZRKGC)* (Li et al., 2020)[6]: a model that is based on UniLM (Dong et al., 2019) and optimized with Generalized EM method.

For the content style, we consider the following models as baselines: (1) *Emotional Chatting*

---

[4]We provide the evaluation results of training on the whole Wizard and CMU_DoG in the appendix.

[5]We exploit Roberta trained on the SST-2 training set (Socher et al., 2013) as the evaluator.

[6]https://github.com/nlpxucan/ZRKGC

Models	Wizard Seen					CMU_DoG				
	Fluency	Context Coherence	Knowledge Relevance	Style Consistency	Kappa	Fluency	Context Coherence	Knowledge Relevance	Style Consistency	Kappa
BART	1.68	1.56	1.52	1.34	0.64	1.62	1.57	1.55	1.31	0.63
ZRKGC	1.62	1.59	1.55	1.36	0.65	1.61	1.53	1.65	1.56	0.66
Our Model	1.71	1.64	1.66	1.77	0.60	1.61	1.66	1.63	1.76	0.74

Table 3: Human evaluation results on learning structure style.

Machine (ECM) (Zhou et al., 2018a)[7]: a model which can generate appropriate responses not only content-relevant but also emotional consistent; (2) variant of *DialoGPT* (Zhang et al., 2019b): we add a sentiment indicating token at the first of the sequence and explore whether such simple heuristics works for controlling knowledge expression; (3) *CTRL* (Keskar et al., 2019) [8] : a large-scale model trained on conditional codes to govern the style and content of generation.

Our model and all baselines are trained on the identical Reddit Corpus to maintain fairness.

4.4 Results on Learning Structure Style

In this section, we demonstrate the effectiveness of our segmentation-based generation framework in both low-resource setting and zero-resource setting and empirically verify that our model can learn structure style with a few annotated examples.

In zero-resource setting, we trained our model on the Reddit Corpus published by Li et al. (2020) and tested on Wizard and CMU_DoG respectively. Automatic evaluation results are shown in Table 2. It could be observed that: (1) our model significantly outperforms ZRKGC and BART on most metrics and achieves the new state-of-the-art performance on Wizard. It is impressive that our model exceeds BART in CMU_DoG especially since the proposed model degrades into BART without two sequential latent variables Z and M. The result serves as strong evidence for the effect of two latent variables, which enable the model to learn complex expression style in Reddit Corpus to handle flexible expression in CMU_DoG. By contrast, BART is far from satisfying with only a regular decoder. (2) our model exceeds ZRKGC significantly in terms of Distinct metrics, for ZRKGC mainly focuses on leveraging external knowledge sources for response generation, but falls short on expression diversity. In the low-resource setting, after training our model on the Reddit Corpus, we then fine-tune it with only 10% training size of Wizard and CMU_DoG

respectively (i.e., $\mathcal{D}_{sty}$ in Sec 3.1) to adjust $p(z_t)$ and $p(m_t)$ to a new structure style. When provided with only 10% training data, our model gets obvious improvement ($\sim 1\%$ increase in F1) in contrast with BART ($\sim 0.5\%$ increase in F1) and ZRKGC ($\sim 0.2\%$ increase in F1), proving that the proposed model can learn more sophisticated structure style through quick adjustment on a specific dataset with little cost.[9]

Human Evaluation. Table 3 shows human evaluation results on learning structure style. It could be observed that: (1) our model is significantly superior to others on *style consistency*, indicating that the model can learn a consistent expression style with very little data. (2) our model has better performance on *context coherence* and *knowledge relevance*, tallying with its impressive performance in the low-resource scenario.

Fine-tuning with Limited Annotated Data. We first train the model on the Reddit Corpus and then fine-tune it with the amount of annotated data (e.g., Wizard and CMU_DoG) gradually increasing from 2% to 10%. To have a more intuitive understanding of the effects of latent variables Z and M, we compare the proposed model with BART, which generates the response with a single decoder. The evaluation results are shown in Figure 2. It can be concluded from the result that: (1) our model can learn the expression style of a particular dataset more efficiently. As the training data increases, our model has a more significant improvement in terms of the F1 metric; (2) our model performs better in meager resources since there is a considerable gap between our model and BART when the training data is close to 0%; (3) the expression style of CMU_DoG can be learned with less data because the model has a significant change in performance after using 2% CMU_DoG training data.

Refashioning of Knowledge-related Segments. To know how our model adjusts to different

---

[7] https://github.com/thu-coai/ecm
[8] https://github.com/salesforce/ctrl

[9] After trained with 10% annotated data, the diversity of our model decreases for it fits the specific expression style of Wizard or CMU_DoG.
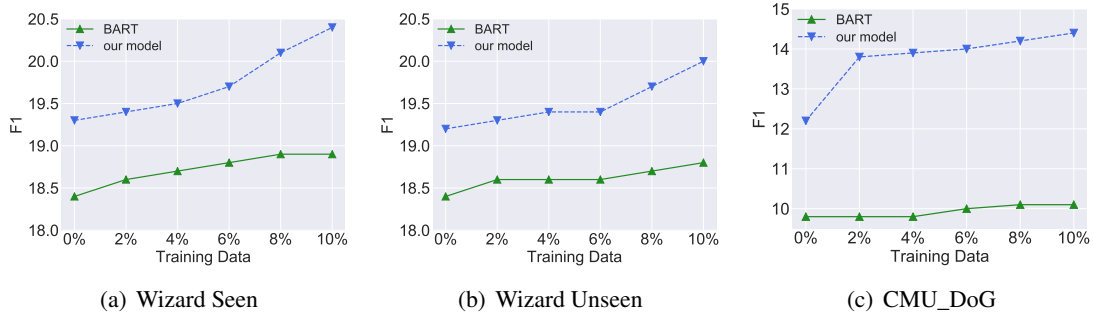
	(a) Wizard Seen	(b) Wizard Unseen	(c) CMU_DoG

Figure 2: Performance of different models wrt. training data size.

Models	Wizard Seen				Wizard Unseen				CMU_DoG			
	positive		negative		positive		negative		positive		negative	
	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc
ECM	10.5	55.8	10.2	60.7	10.1	55.7	10.1	57.6	7.6	41.5	8.3	55.4
DialoGPT	12.1	54.1	12.1	46.9	12.0	56.0	12.0	45.0	9.2	44.9	9.2	55.1
CTRL	15.3	71.9	14.9	55.3	14.9	75.0	14.6	52.3	9.3	70.2	9.2	61.7
Our Model	**19.7**	70.3	**19.2**	70.7	**19.4**	73.1	**19.2**	69.9	**12.7**	74.8	**12.2**	68.0

Table 4: Evaluation results on sentiment control. Numbers in bold mean that the improvement to the best performing baseline is statistically significant (t-test with p-value < 0.05).
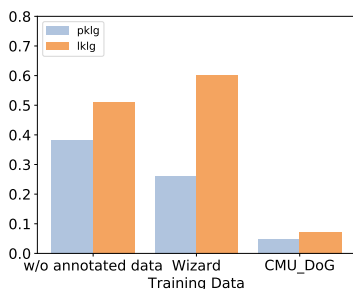

Figure 3: The effect of fine-tuning on different data.

datasets, we compare the knowledge-related segments before and after trained with annotated data from two aspects: (1) the average proportion of knowledge-related segments ($pklg$) in a sentence; (2) the average proportion of words belonging to knowledge-related segments ($lklg$). The motivation behind is that the frequency and length of these two kinds of segments generally indicates how well the latent variable is learned to capture the knowledge expression structure. We identify these two kinds of segmentation by comparing their lexical similarities with the knowledge. Figure 3 reports the results. The results indicate that our model could learn the underlying structure style of both datasets, with the great difference of $pklg$ and $lklg$ before and after fine-tuning as evidence. After fine-tuning with Wizard data, $pklg$ drops to 0.26 while the $lklg$ grows up a bit, indicating that the knowledge-related segments generated by our model are fewer and longer, which tallies with the fact that the responses in Wizard are probably directly copied from background knowledge. How-

ever, after CMU_DoG data is fed to the model, both $pklg$ and $lklg$ shrink drastically, which agrees with the fact that crowd-sourcing workers converse more liberally online and the responses are less relevant to the background knowledge.

4.5 Results on Learning Content Style

We further investigate whether the proposed model could express knowledge with the desired sentiment. Specifically, we introduce two sets of style adapters to endow knowledge expression in two different sentiments, namely positive and negative. So in this scenario, it is required that responses are not only coherent with context but also limited in positive or negative sentiment. To apply ECM on knowledge-grounded conversation, we label the sentiment category for each response with a classifier pre-trained on the SST (Socher et al., 2013) training set. For DialoGPT, we similarly annotate each response with a sentiment category and append the sentiment token before the context tokens. The evaluation results are shown in Table 4. We can conclude that: (1) The proposed model outperforms all baseline models in terms of all metrics, which indicates that our model can control the sentiment of knowledge expression and guarantee high quality of the generated responses; (2) Simply adding a sentiment indicating token at the beginning of the sequence can not effectively control the style of knowledge expression, as the performance of DialoGPT on sentiment control is poor; (3) Although ECM is designed for sentiment con-

trol, it still fails to perform well in this task, proving that sentiment control in the knowledge-grounded conversation is rather difficult. Besides, ECM can only control the sentiment of the whole response but is helpless to manage every knowledge-related segment at a fine-grained level.

5 Conclusions

We explore knowledge expression in knowledge-grounded conversation and break down the expression style of a response into the structure of the response (structure style) and the style of the content in each part (content style). We propose a variational segmentation-based generation model to discover the underlying expression style in response. Specifically, we introduce two latent variables to model these two aspects of expression style respectively and induce an evidence lower bound of the likelihood. Evaluation results on two benchmarks of the task indicate that our model can learn the structure style with little cost and generate responses in desired content style without any human-annotated data.

Ethical Considerations

It's crucial for an open-domain dialogue system to be able to automatically detect and discover the underlying structural pattern of a sentence. With the ability to handle a variety of expression styles, whether positive or negative, serious or casual, our work suggests that we are getting closer to the goal of creating an artificial intelligent dialogue system that can freely communicate with humans, which is beyond the wildest dreams of most AI and NLP researchers. However, a detailed survey should be undertaken in advance to consider the immediate audience's and developers' interests, as well as any potential stakeholder groups.

Furthermore, knowledge-grounded dialogue systems have the potential to fabricate facts and distribute rumors and false information, particularly when the source of external background knowledge is unreliable. If the knowledge candidate set is contaminated by fake news, the response generated by the dialogue system is likely to suffer from the "hallucination" issue. Controlling the source of knowledge sentences, such as paragraphs extracted from the wiki, authoritative news sites, or authoritative product documents, is a necessary and practical strategy.

Acknowledgement

References

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. Wizard of wikipedia: Knowledge-powered conversational agents. In *ICLR*.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*, pages 13042–13054.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

Tingchen Fu, Xueliang Zhao, Chongyang Tao, Ji-Rong Wen, and Rui Yan. 2022. There are a thousand hamlets in a thousand people's eyes: Enhancing knowledge-grounded dialogue with personal memory. *arXiv preprint arXiv:2204.02624*.

Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. 2013. Stochastic variational inference. *Journal of Machine Learning Research*, 14(5).

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.

Po-Sen Huang, Chong Wang, Sitao Huang, Dengyong Zhou, and Li Deng. 2017. Towards neural phrase-based machine translation. *arXiv preprint arXiv:1706.05565*.

Bernd Huber, Daniel McDuff, Chris Brockett, Michel Galley, and Bill Dolan. 2018. Emotional dialogue generation using image-grounded language models. In *CHI*, page 277. ACM.

Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*.

Byeongchang Kim, Jaewoo Ahn, and Gunhee Kim. 2020. Sequential latent knowledge selection for knowledge-grounded dialogue. *arXiv preprint arXiv:2002.07510*.

Taesup Kim, Sungjin Ahn, and Yoshua Bengio. 2019. Variational temporal abstraction. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 11570–11579.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models. *NAACL*, pages 110–119.

Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and Bill Dolan. 2016. A persona-based neural conversation model. In *ACL*, pages 994–1003.

Linxiao Li, Can Xu, Wei Wu, Yufan Zhao, Xueliang Zhao, and Chongyang Tao. 2020. Zero-resource knowledge-grounded dialogue generation. *arXiv preprint arXiv:2008.12918*.

Rongzhong Lian, Min Xie, Fan Wang, Jinhua Peng, and Hua Wu. 2019. Learning to select knowledge for response generation in dialog systems. *arXiv preprint arXiv:1902.04911*.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60.

Seungwhan Moon, Pararth Shah, Anuj Kumar, and Rajen Subba. 2019. Opendialkg: Explainable conversational reasoning with attention-based walks over knowledge graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 845–854.

Nasrin Mostafazadeh, Chris Brockett, Bill Dolan, Michel Galley, Jianfeng Gao, Georgios Spithourakis, and Lucy Vanderwende. 2017. Image-grounded conversations: Multimodal context for natural question and response generation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 462–472.

Abigail See, Stephen Roller, Douwe Kiela, and Jason Weston. 2019. What makes a good conversation? how controllable attributes affect human judgments. *arXiv preprint arXiv:1902.08654*.

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*, 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*, pages 3295–3301.

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. In *ACL*, pages 1577–1586.

Xiaoyu Shen, Ernie Chang, Hui Su, Cheng Niu, and Dietrich Klakow. 2020. Neural data-to-text generation via jointly learning the segmentation and correspondence. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7155–7165.

Kurt Shuster, Samuel Humeau, Antoine Bordes, and Jason Weston. 2018. Engaging image chat: Modeling personality in grounded dialogue. *arXiv preprint arXiv:1811.00945*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Chongyang Tao, Shen Gao, Mingyue Shang, Wei Wu, Dongyan Zhao, and Rui Yan. 2018. Get the point of my utterance! learning towards effective responses with multi-head attention mechanism. In *IJCAI*, pages 4418–4424.

Yi-Lin Tuan, Yun-Nung Chen, and Hung-yi Lee. 2019. Dykgchat: Benchmarking dialogue generation grounding on dynamic knowledge graphs. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1855–1865.

Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *arXiv preprint arXiv:1506.05869*.

Chong Wang, Yining Wang, Po-Sen Huang, Abdelrahman Mohamed, Dengyong Zhou, and Li Deng. 2017. Sequence modeling via segmentations. In *International Conference on Machine Learning*, pages 3674–3683. PMLR.

Yansen Wang, Chenyi Liu, Minlie Huang, and Liqiang Nie. 2018. Learning to ask questions in open-domain conversational systems with typed decoders. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2193–2203.

Sam Wiseman, Stuart M Shieber, and Alexander M Rush. 2018. Learning neural templates for text generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3174–3187.

Chen Xing, Wei Wu, Jie Liu, Yalou Huang, Ming Zhou, and Wei-Ying Ma. 2017. Topic aware neural response generation. In *AAAI*, pages 3351–3357.

Can Xu, Wei Wu, Chongyang Tao, Huang Hu, Matt Schuerman, and Ying Wang. 2019. Neural response generation with meta-words. *arXiv preprint arXiv:1906.06050*.

Hainan Zhang, Yanyan Lan, Liang Pang, Jiafeng Guo, and Xueqi Cheng. 2019a. Recosa: Detecting the relevant contexts with self-attention for multi-turn dialogue generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3721–3730.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? *arXiv preprint arXiv:1801.07243*.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2019b. 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*, pages 654–664.

Xueliang Zhao, Chongyang Tao, Wei Wu, Can Xu, Dongyan Zhao, and Rui Yan. 2019. A document-grounded matching network for response selection in retrieval-based chatbots. In *IJCAI*, pages 5443–5449.

Xueliang Zhao, Wei Wu, Chongyang Tao, Can Xu, Dongyan Zhao, and Rui Yan. 2020a. Low-resource knowledge-grounded dialogue generation. In *International Conference on Learning Representations*.

Xueliang Zhao, Wei Wu, Can Xu, Chongyang Tao, Dongyan Zhao, and Rui Yan. 2020b. Knowledge-grounded dialogue generation with pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3377–3390.

Hao Zhou, Minlie Huang, Tianyang Zhang, Xiaoyan Zhu, and Bing Liu. 2018a. Emotional chatting machine: Emotional conversation generation with internal and external memory. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. 2018b. Commonsense knowledge aware conversation generation with graph attention. In *IJCAI*, pages 4623–4629.

Kangyan Zhou, Shrimai Prabhumoye, and Alan W Black. 2018c. A dataset for document grounded conversations. *arXiv preprint arXiv:1809.07358*.

A Appendix

A.1 Derivation of ELBO

$$\log p(R|U, K)$$

$$= \log \sum_{(M,Z)} p(R, M, Z)$$

$$= \log \sum_{(M,Z)} q(M, Z|R) \frac{p(R, M, Z)}{q(M, Z|R)}$$

$$= \log \mathbb{E}_{(M,Z)\sim q(M,Z|R)} \frac{p(R, M, Z)}{q(M, Z|R)} \quad (10)$$

$$\geq \mathbb{E}_{(M,Z)\sim q(M,Z|R)} \log \frac{p(R, M, Z)}{q(M, Z|R)}$$

$$= \mathbb{E}_{(M,Z)\sim q(M,Z|R)} \log p(R|M, Z)$$

$$- \mathbb{E}_{(M,Z)\sim q(M,Z|R)} \big(\log q(M, Z|R)$$

$$- \log p(M, Z) \big).$$

According to the mean-filed approximation, $q(M, Z) \approx q(M)q(Z)$. Therefore, $\mathbb{E}_{(M,Z)\sim q(M,Z|R)} \log p(R|M, Z)$ and $\mathbb{E}_{(M,Z)\sim q(M,Z|R)} \big(\log q(M, Z|R) - \log p(M, Z) \big)$ can be re-written as:

$$\mathbb{E}_{(M,Z)\sim q(M,Z|R)} \log p(R|M, Z)$$

$$= \mathbb{E}_{M\sim q(M|R)} \big(\mathbb{E}_{Z\sim q(Z|M,R)} \sum_{t=1}^{l_r} \log p(r_t|r_{<t}, z_t) \big))$$

$$(11)$$

$$\mathbb{E}_{(M,Z)\sim q(M,Z|R)} \big(\log q(M, Z|R) - \log p(M, Z) \big)$$

$$= \mathbb{E}_{M\sim q(M|R)} \Big(\mathbb{E}_{Z\sim q(Z|M,R)} \big(\log q(M|R) - \log p(M) \big)$$

$$+ \mathbb{E}_{Z\sim q(Z|M,R)} \big(\log q(Z|M, R) - \log p(Z) \big) \Big)$$

$$= \mathbb{E}_{M\sim q(M|R)} \Big(\log q(M|R) - \log p(M) \Big)$$

$$+ \mathbb{E}_{M\sim q(M|R)} \Big(\mathbb{E}_{Z\sim q(Z|M,R)} \big(\log q(Z|M, R) - \log p(Z) \big) \Big)$$

$$= \sum_{t=1}^{l_r} \Big(\mathbb{E}_{M\sim q(M|R)} \big(\log q(m_t) - \log p(m_t) \big) \Big)$$

$$+ \mathbb{E}_{M\sim q(M|R)} \Big(\sum_{t=1}^{l_r} m_{t-1} \cdot \mathbb{E}_{Z\sim q(Z|M,R)} \big(\log q(z_t) - \log p(z_t) \big) \Big)$$

$$= \sum_{t=1}^{l_r} D_{\text{KL}}(q(m_t)\|p(m_t))$$

$$+ \mathbb{E}_{M\sim q(M|R)} \Big(\sum_{t=1}^{l_r} m_{t-1} \cdot D_{\text{KL}}(q(z_t)\|p(z_t)) \Big).$$

$$(12)$$

A.2 Details about the Construction of $\tilde{M}$ and $\tilde{Z}$

In this section, we provide more details about of construction of $\tilde{M}$ and $\tilde{Z}$. For every response in the training set, we parse it as a syntax tree using StanfordNLP toolkit (Manning et al., 2014). The syntax tree we obtain is in a hierarchical and nested structure. The root node of the tree represents the whole response sentence and the root node of every subtree represents a corresponding phrase, a small part of a sentence. For example, if a phrase could be divided into three parts, then the node representing the phrase has three child nodes and each represents a part of the phrase. After we acquire the parsing tree, segmentation is then carried out recursively. To be concrete, we traverse the parsing tree by deep-first search order. Every time we arrive at a node, compute the similarity[10] between the knowledge and the phrase represented by the node. If the similarity is above the threshold μ_{seg}, we mark the phrase as a segment and search in this branch terminates. Else we continue to search the child nodes of the current node to segment at a more refined level. We use $\tilde{M} = \{\tilde{m}_t\}_{t=1}^{l_r}$ to denote the results of segmentation labeling.

The pseudo label of module choice $\tilde{Z} = \{z_t\}_{t=1}^{l_r}$ is tagged in a similar way to multiclass classification. Specifically, for a segment $(r_s, \cdots, r_e)$ where s and e are the start and end position of a segment respectively. If the similarity between this segment and the knowledge falls below a threshold μ_{knl}, its pseudo label $(z_s, \cdots, z_e)$ will be set to 0. Otherwise we send the segment to a series of style discriminators one after another until the classification confidence given by a discriminator is above μ_{sty_i} and pseudo module choice label will be set to $i+1$. If all discriminators fail to classify the segment at a confidence greater than μ_{sty_i}, $(z_s, \cdots, z_e)$ are all 1, indicating knowledge should be expressed without particular style.

A.3 Learning Algorithm

The learning algorithm is summarized in Algorithm 1.

A.4 Details of Datasets

Training Data. We choose the Reddit Corpus published by (Li et al., 2020) as $\mathcal{D}$ for pre-training. The data contains $842,521$ context-knowledge-response triples for training and $2,737$ context-knowledge-response triples for validation. On average, each dialogue contains 3.1 utterances in both sets, and the average length of the utterance is 16.0 in training and is 16.1 in validation.

Evaluation Data. We test our model on benchmarks of knowledge-grounded dialogue generation,

---

[10]We use unigram Precision to calculate the similarity.

Algorithm 1 Learning Algorithm

1: **Input:** Training data $\mathcal{D}$, thresholds for weak supervision μ_{seg}, μ_{knl} and μ_{sty}, discriminator$\{Dis_i\}_{i=1}^{N_{sty}}$, maximum step M, adapter training step M'.
2: **for** $m \leftarrow 1$ to M **do**
3:     Sample a mini-batch $\{(U_i, K_i, R_i)\}$ from $\mathcal{D}$.
4:     Conduct segmentation on R_i to get $\tilde{M}$.
5:     **for** $i \leftarrow 1$ to N_{seg} **do**
6:         **for** $j \leftarrow 1$ to N_{sty} **do**
7:             Use Dis_j to classify response segment $(r_{s_i}, \cdots, r_{e_i})$.
8:             **if** Confidence of $Dis_j \geq \mu_{sty}$ and $(z_{s_i}, \cdots, z_{e_i})$ are not assigned **then**
9:                 $(z_{s_i}, \cdots, z_{e_i}) \leftarrow j + 1$
10:             **end if**
11:         **end for**
12:     **end for**
13:     **if** $m \leq M'$ **then**
14:         Update the adapters based on the first term in ELBO.
15:     **else**
16:         Update the parameters θ (i.e., θ_m, θ_z and the parameters in $p(r_t)$) and ϕ (i.e., ϕ_m and ϕ_z) based on ELBO and weak supervision.
17:     **end if**
18: **end for**
19: **return** Generation Model $p_\theta(R|U, K)$ with prior distribution p_{θ_m} and p_{θ_z}

including Wizard of Wikipedia (Wizard) (Dinan et al., 2019) and CMU Document Grounded Conversations (CMU_DoG) (Zhou et al., 2018c). Both datasets are split into training sets, validation sets, and test sets by the data owners. We follow Dinan et al. (2019) and conduct the pre-processing with the code published on ParlAI[11]. Topics in Wizard cover a wide range ($1,365$ in total), and each conversation happens between a wizard who has access to the knowledge about a specific topic and an apprentice who is just eager to learn from the wizard about the topic. The test set is split into two subsets. Test Seen only contains dialogues with topics that have already appeared in the training set, while topics in Test Unseen never appear in the training set and the validation set. Different from Wizard, CMU_DoG focuses on movie domain, and besides wizard-apprentice conversations, the data also contain conversations between two workers who know the document and try to discuss the content in depth. In both datasets, only the turns where knowledge is accessible are considered in response generation. Table 5 reports the statistics of the Wizard data and the CMU_DoG data

A.5 More Implementation Details

We employ a knowledge selection (KS) module to select the top 7 related sentences in knowledge. The KS module is implemented based on Roberta-base (125M) and trained on the Reddit Corpus. Specifically, we treat the sentence which has the highest F1 score with the response as the positive sample, and the negative sample is randomly sampled from all the other knowledge sentences. We train the KS module via maximum likelihood estimation (MLE) with a batch size of 64 and an initial learning rate of $1e-5$. The threshold μ_{seg}, μ_{knl}, μ_{pos} and μ_{neg}[12] in weak supervision are set as 0.9, 0.5, 0.8 and 0.8, respectively. The encoder-decoder architecture is implemented on the basis of Bart-base (139M) and trained on the Reddit Corpus with a batch size of 64 and an initial learning rate of $5e-6$. The parameters for prior and posterior distributions of Z and M (i.e., θ_z, θ_m, ϕ_z and ϕ_m) are initialized randomly, and optimized with a learning rate of $1e-4$. The parameters for adapters are initialized randomly and optimized with a learning rate of $2e-3$. We only train the adapters for the first 1000 steps. We utilize gated recurrent units (GRUs) as the basic units in $f_{z-\mathrm{rnn}}$. We set the hidden size and the number of layers of RNN in our model (i.e., $f_{z-\mathrm{rnn}}$ and $\psi(\cdot)$) as 128 and 1 respectively. The embedding size for Z is set as 128 and the adapter size is set as 64. When fine-tuning the model on the Wizard and CMU_DoG datasets, the learning rate and the batch size are set as $5e-5$ and 32 respectively. We employ greedy search in response decoding. All models are learned with Adam (Kingma and Ba, 2015) optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We increase the learning rate linearly for the first 200

---

[11] https://github.com/facebookresearch/ParlAI/blob/master/projects/wizard_of_wikipedia

[12] We consider positive and negative sentiment style in our experiments.

		Wizard of Wikipedia				CMU_DoG		
		Train	Valid	Test Seen	Test Unseen	Train	Valid	Test
# Utterances		166,787	17,715	8,715	8,782	74,717	4,993	13,646
# Conversations		18,430	1,948	965	968	3,373	229	619
# Topics/Documents		1,247	599	533	58	30	30	30
Avg. # of Turns		9.0	9.1	9.0	9.1	22.2	21.8	22.0

Table 5: Statistics of the Wizard data and the CMU_DoG data.

Training Data	Models	Wizard Seen				Wizard Unseen				CMU_DoG			
		PPL	BLEU-1	ROUGE-L	METEOR	PPL	BLEU-1	ROUGE-L	METEOR	PPL	BLEU-1	ROUGE-L	METEOR-L
Reddit Corpus	BART	40.1	0.206	0.164	0.099	42.9	0.202	0.167	0.099	75.8	0.141	0.117	0.060
	ZRKGC	41.1	0.225	0.163	0.099	42.7	0.220	0.164	0.100	53.8	0.161	0.128	0.075
	Ours	35.9	0.235	0.168	0.095	38.4	0.232	0.169	0.095	60.4	0.166	0.131	0.069
Reddit Corpus +10% training data	BART	32.7	0.203	0.174	0.103	35.0	0.199	0.175	0.103	49.5	0.141	0.122	0.067
	ZRKGC	29.1	0.227	0.175	0.101	31.6	0.222	0.176	0.100	38.0	0.173	0.139	0.083
	Ours	28.6	0.237	0.181	0.103	30.7	0.231	0.176	0.101	40.8	0.182	0.139	0.080

Table 6: More Results about Automatic Evaluation.

steps and decrease it thereafter proportionally to the inverse square root of the step number. Early stopping on validation is adopted as a regularization strategy. All models are trained on a $8\times$RTX 2080 Ti machine.

A.6 More Results about Automatic Evaluation

Table 6 reports more results about the automatic evaluation, from which we can see that our model still outperforms the baselines.

A.7 Human Evaluation

We randomly sample 300 examples from Test Seen of Wizard, and the test set of CMU_DoG respectively, and recruit 6 well-educated native speakers to do qualitative analysis on the responses generated by our model and all baselines. For each of the 300 examples, an annotator is provided with the context, the ground-truth knowledge, model responses and the associated style types. For evaluation of structure style, we defined two kinds of structure styles based on two datasets, namely the Wizard-like style S_{wizard} and the CMU_DoG-like style S_{cmudog}. While for evaluation of content style, we roughly divide content styles in two categories, S_{pos} and S_{neg} for convenience. The responses provided by different models are randomly shuffled to hide their sources. The annotators need to judge the quality of the responses from four aspects: (1) *fluency*: whether the response is fluent without any grammatical errors; (2) *context coherence*: whether the response is coherent with the context; (3) *knowledge relevance*: whether the re-

sponse is relevant with the knowledge; and (4) *style consistency*: whether the response exhibits the desired style. Each annotator assigns a score from $\{0, 1, 2\}$ (representing "bad", "fair" and "good" respectively) to each response for each aspect. Each response obtains four scores for aforementioned four aspects, and the agreement among all annotators is measured via Fleiss' kappa (Fleiss, 1971).

Results on Learning content style. Table 7 reports the human evaluation results on learning content style. The three models are trained on the Reddit Corpus. We can conclude that: (1) by introducing two latent variables and a number of adapters for different styles, our model can generate responses in desired content style (i.e., S_{pos} and S_{neg}) more accurately and achieve significant improvement on *style consistency*, which is consistent with the results in Table 4; (2) our model also outperforms ECM and DialoGPT on *fluency*, *context coherency* and *knowledge relevance* thanks to the capacity of large-scale pre-trained language models and the introduction of external knowledge respectively.

A.8 Comparison with More Baselines

We compare with models trained on full training data, and Table 8 shows the evaluation results. First, it is noted that our model outperforms KnowledGPT in terms of F1 by using only 10% training data[13] on CMU_DoG, which provides a strong support for the effectiveness of the proposed model.

---

[13]The 10% training data is randomly sampled. The result is an average value of three repetitive experiments on every dataset

Models	Wizard Seen					CMU_DoG				
	Fluency	Context Coherence	Knowledge Relevance	Style Consistency	Kappa	Fluency	Context Coherence	Knowledge Relevance	Style Consistency	Kappa
ECM	0.85	0.94	1.02	1.24	0.65	0.96	0.95	1.18	1.08	0.72
DialoGPT	1.57	1.41	1.19	1.26	0.75	1.55	1.62	1.09	1.02	0.65
Our	1.64	1.60	1.78	1.72	0.76	1.59	1.63	1.51	1.69	0.62

Table 7: Human evaluation results on learning content style.

Training Data	Models	Wizard Seen			Wizard Unseen			CMU_DoG		
		F1	D-1	D-2	F1	D-1	D-2	F1	D-1	D-2
100% annotated data	TMN	15.9	0.041	0.176	14.3	0.025	0.106	9.9	0.003	0.008
	SKT	19.3	0.085	0.300	16.1	0.056	0.188	-	-	-
	DRD	19.3	0.065	0.252	17.9	0.046	0.177	10.7	0.010	0.044
	KnowledGPT	22.0	0.141	0.431	20.5	0.094	0.290	13.5	0.023	0.113
	Our Model	22.0	0.128	0.415	20.8	0.090	0.278	15.3	0.031	0.121
Reddit Corpus	BART	18.4	0.076	0.355	18.4	0.049	0.237	9.8	0.021	0.131
	ZRKGC	18.9	0.055	0.246	18.8	0.037	0.179	12.2	0.015	0.094
	Our Model	19.3	0.082	0.383	19.2	0.060	0.292	12.2	0.028	0.186
Reddit Corpus + 10% annotated data	Our Model	20.4	0.073	0.366	20.0	0.052	0.270	14.4	0.015	0.122
Reddit Corpus + 100% annotated data	Our Model	21.9	0.134	0.453	21.2	0.103	0.302	15.5	0.041	0.134

Table 8: Automatic evaluation results.

Second, by adjusting the structure style on a small amount of data, the gap between our model and KnowledGPT is further narrowed, while the improvements on ZRKGC and BART are trivial.

A.9 Ablation over Weak Supervision

To have more insights into the impact of weak supervision on the performance of our model, we compare the proposed model with the following variants: (1)-*weak supervision on Z*: the weak supervision on module indicator Z is removed; (2)-*weak supervision on Z and M*: the weak supervision on module indicator and boundary indicator is removed. Table 9 reports the evaluation results. We can conclude that (1) the weak supervision objectives significantly improve model performance; (2) the weak supervision objectives play a more crucial role on CMU_DoG, as removing them causes a dramatic drop in performance. The reason is that this dataset has more sophisticated expression styles and it is difficult to learn these styles without auxiliary supervision signals.

A.10 Ablation over Boundary Indicator

Since the module indicator is conditioned on the boundary indicator, we are curious about what will happen if the M is removed. The ablation result is shown on Figure 4. There is an evident drop on Wizard Seen and Wizard Unseen, verifying the effect of boundary indicator in assisting the module indicator. The margin is tiny on CMU_DoG,
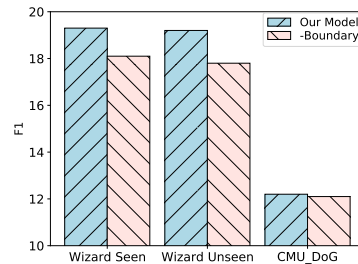

Figure 4: Ablation results on Wizard and CMU_DoG.

perhaps because its structure feature is easier to capture, so the module indicator could works properly itself.

A.11 Case Study

This section mainly studies how different models vary in knowledge expression for the same context and background knowledge. Table 10 shows an example from the test set of CMU_DoG. This example contains the background knowledge which gives a plot from the movie, and the dialogue context which is generated by discussing the content in the knowledge. We choose the following four models to generate the response in corresponding style given the dialogue context and knowledge, and all models are pre-trained with the Reddit Corpus: (1) Wizard Model for S_{wizard}: the model fine-tuned with 10% training data in Wizard; (2) CMU_DoG Model for S_{cmudog}: the model fine-tuned with 10% training data in CMU_DoG; (3) Positive Model for S_{pos}: the model forced to express knowledge with

Training Data	Models	Wizard Seen			Wizard Unseen			CMU_DoG		
		F1	D-1	D-2	F1	D-1	D-2	F1	D-1	D-2
Reddit Corpus	Our model	19.3	0.082	0.383	19.2	0.060	**0.292**	**12.2**	0.028	**0.186**
	-weak supervision on Z	19.1	0.077	0.362	19.1	0.056	0.270	10.2	0.027	0.155
	-weak supervision on Z and M	19.1	0.083	0.382	18.8	0.058	0.270	9.5	0.023	0.147
Reddit Corpus + 10% annotated data	Our model	**20.4**	0.073	0.366	**20.0**	0.052	**0.270**	**14.4**	0.015	**0.122**
	-weak supervision on Z	19.5	0.072	0.354	19.3	0.051	0.250	13.2	0.014	0.115
	-weak supervision on Z and M	19.5	0.077	0.366	19.2	0.054	0.258	13.5	0.013	0.091

Table 9: Ablation study over the weak supervision. Numbers in bold means that the improvement to variants is statistically significant (t-test with p-value < 0.05)

Knowledge	• MovieName: How to Train Your Dragon • Back at the village, Hiccup subdues a captive dragon in his final training test in front of his father instead of killing it, but Stoick inadvertently angers the dragon into attacking. • Toothless attempts to protect Hiccup in the ensuing panic but is instead captured by the Vikings. • Hiccup accidentally reveals to Stoick that Toothless is capable of locating the dragons' nest. • Stoick disowns his son and sets off for the nest with Toothless chained to the lead ship as a guide. • The Vikings expel most of the dragons but are overwhelmed by the Red Death until Hiccup, Astrid and their fellow pupils fly in riding the training dragons from the academy and provide cover fire. • Hiccup almost drowns trying to break Toothless free from a sinking ship but Stoick saves them both and then reconciles with his son. • Toothless and Hiccup destroy the Red Death but Hiccup is injured in the fight. • Hiccup regains consciousness on Berk where his lower left leg has been replaced by a prosthesis, and the Vikings and the dragons now live in harmony.
Context	User1: Now I keep thinking about phantom pain and whatnot... will he walk with his new prothesis? Was does viking prothesis look like?! User2: ...but now they live there together! Schweet. User1: Yes!
Ground Truth	Probably sharp and imposing!
Wizard Model	They live in a house together . It ' s the same as before but with toothless and hiccup now living on berk where his lower left leg has been replaced by a prosthesis.
CMU_DoG Model	What is the new prothesis ? ! It ' s a Viking
Positive Model	Vikings live in harmony now have the dragons living together and are all still alive to be united.
Negative Model	Though Vikings and the dragons live in harmony, Hiccup can no longer walk with this lower left leg.

Table 10: A case from test set of CMU_DoG.

positive sentiment; (4) Negative Model for S_{neg}: the model forced to express knowledge with negative sentiment. We can see that the knowlege expression style of the Wizard Model and CMU_DoG Model are quite different. The central part of the Wizard Model response is copied from the background knowledge, which is consistent with the style of Wizard data. The response generated by CMU_DoG Model is more casual in knowledge expression, and the content is mainly related to the conversation context. Besides, responses generated by the Positive Model exhibit evident positive sentiment, while responses generated by the Negative Model show relatively negative sentiment.