SIDECONTROL: Controlled Open-domain Dialogue Generation via Additive Side Networks

Wanyu Du **Yangfeng Ji**
Department of Computer Science
University of Virginia
Charlottesville, VA 22904
{wd5jq,yangfeng}@virginia.edu

Abstract

Transformer-based pre-trained language models boost the performance of open-domain dialogue systems. Prior works leverage Transformer-based pre-trained language models to generate texts with desired attributes in two general approaches: (1) gradient-based methods: updating all latent representations of pre-trained models with gradients from attribute models; (2) weighted-decoding methods: re-ranking beam candidates from pre-trained models with attribute functions. However, gradient-based methods lead to high computation cost and can easily get overfitted on small training sets, while weighted-decoding methods are inherently constrained by the low-variance high-bias pre-trained model. In this work, we propose a novel approach to control the generation of Transformer-based pre-trained language models: the SIDECONTROL framework, which leverages a novel control attributes loss to incorporate useful control signals, and is shown to perform well with very limited training samples. We evaluate our proposed method on two benchmark open-domain dialogue datasets, and results show that the SIDECONTROL framework has better controllability, higher generation quality and better sample-efficiency than existing gradient-based and weighted-decoding baselines. [1]

1 Introduction

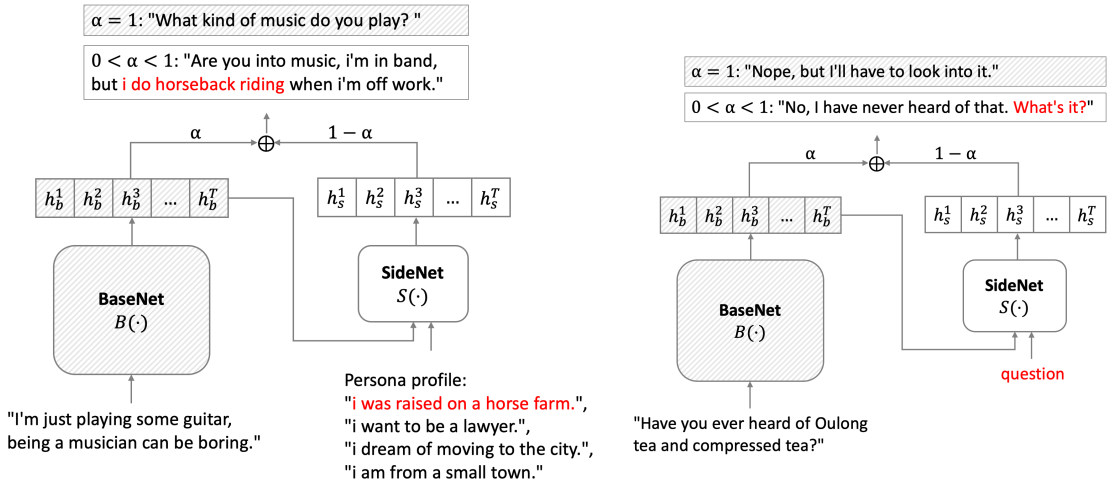
With the advance of Transformer-based pre-trained language models (Radford et al., 2019; Raffel et al., 2020; Brown et al., 2020; Zhang et al., 2020), many dialogue systems (Zhang et al., 2020; Roller et al., 2020; Shuster et al., 2020) have shown promising performance in challenging open-domain conversations with humans. However, for controlled dialogue generation, prior works mainly focus on building LSTM-based class-conditional generative model on specific datasets with task-specific design on model architecture (Wen et al., 2015; Ke et al., 2018; Chen et al., 2019; See et al., 2019) or policy learning strategy (Kawano et al., 2019; Hsueh and Ma, 2020; Takayama and Arase, 2020; Varshney et al., 2021). In this work, we explore effective method for controlled generation on Transformer-based dialogue systems, with the goal of adding controllability functionality into state-of-the-art Transformer-based dialogue systems with lower computation cost, less training data and more flexible control mechanism.

Prior works on controlled text generation for Transformer-based pre-trained language models can be categorized into two general approaches: (1) gradient-based methods and (2) weighted-decoding methods. The gradient-based methods (Dathathri et al., 2019; Goswamy et al., 2020; Lin and Riedl, 2021) propose a plug-and-play language model following $p(x|a) \propto p(a|x)p(x)$, which plugs an attribute model $p(a|x)$ with a pre-trained language model $p(x)$ to control generation. The gradients from $p(a|x)$ are used to guide the latent representations of pre-trained models encoding more control attribute information. The weighted-decoding methods (Ghazvininejad et al., 2017; Baheti et al., 2018; Holtzman et al., 2018; Yang and Klein, 2021) modify the sampling weights with attribute functions in beam search at each decoding timestep to control generation. Essentially, the attribute functions are used to re-rank the original beam candidates generated by the pre-trained language models. The main idea of both gradient-based methods and weighted decoding methods is the flexibility: users can design any attribute models or functions for different controlled generation tasks and apply the attribute model or function to any state-of-the-art pre-trained language models for generating high quality texts.

However, weighted decoding methods (Ghazvininejad et al., 2017; Baheti et al., 2018;

[1] Our code implementation and data sources can be found here: https://github.com/wyu-du/Controlled-Dialogue-Generation.

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 2175–2194
November 7–11, 2021.



(a) SideNet for Knowledge Document Control

(b) SideNet for Semantic Label Control

Figure 1: General architecture of the SIDECONTROL framework.

Holtzman et al., 2018; Yang and Klein, 2021) are limited by the low-variance high-biased pre-trained language models, since they do not update the pre-trained language models. If the pre-trained model yields commonly observed words rather than target attribute words in the beam candidates list, it is difficult for the attribute functions to re-rank and find the target words during generation. Although gradient-based methods (Dathathri et al., 2019; Goswamy et al., 2020; Lin and Riedl, 2021) do not have this limitation since they update the latent representations of pre-trained models during inference, the gradient propagation at each decoding timestep involves heavy computation, which results in slow response speed to users. In addition, the controllability performance of gradient-based methods relies on the attribute model. If the attribute model gets overfitted on a small training set, the gradient from this attribute model will just lead to meaningless updates.

To build an effective and efficient controlled open-domain dialogue system, we propose the SIDECONTROL framework, which treats the pre-trained lanaguage model as a feature extractor and train light-weight side networks to encode complementary information from control attributes. In addition, we introduce a novel control attributes loss to guide the side network during training. As shown in Figure 1, the final output representation is a mixture of a base representation from the pre-trained language model and a side representation from the side network. The mixture coefficient α is learned during training, and is used to balance the prior knowledge from the base network and the task-specific control attributes signals from the side network. From the encoding perspective, the SIDECONTROL framework not only can be applied to any pre-trained language models, but also supports diverse format attributes control (e.g. dialogue act, external knowledge document). From the decoding perspective, the SIDECONTROL framework has low computation cost, since it directly samples from its optimized class-conditional language model $p(x|a)$ without additionally updating latent representations during generation. From the sample-efficiency perspective, the SIDECONTROL framework achieves good performance with a few thousand training samples by leveraging the control loss.

We summarize the contributions of this work as follows:

1. we propose a new controlled dialogue generation framework with novel control attributes losses to support different forms of attributes control (e.g. dialogue act, external knowledge document);

2. we conduct empirical experiments to show the sample-efficiency of the SIDECONTROL framework, which can achieve good performance with only 100 $\sim$ 1000 training samples;

3. we conduct empirical experiments to validate that the SIDECONTROL framework has better controllability, better text quality, and lower decoding cost compared to gradient-based methods and weighted-decoding methods.

2 SideNet for Controlled Generation

Firstly, we introduce the SIDECONTROL framework in subsection 2.1, which presents the general idea of using a small side network to coordinate the generation process based on large-scale pre-trained language models (Zhang et al., 2020; Roller et al., 2020; Shuster et al., 2020). Then we provide two realizations of side networks for two types of control attributes: (1) external knowledge document in subsection 2.2, (2) semantic label in subsection 2.3.

2.1 General Framework

Given a dialogue context which contains a fixed number of previous utterances $\mathbf{X} = \{\boldsymbol{x}_i\}_{i=1}^{N}$, where N is the total number of tokens in the given dialogue context, and a control attribute $\boldsymbol{a}$ which represents the desired controllable attributes, the goal is to build a model conditioned on $\mathbf{X}$ and $\boldsymbol{a}$ that can generate a response which best approximates the ground-truth human response $\mathbf{Y} = \{\boldsymbol{y}_t\}_{t=1}^{T}$:

$$\begin{aligned} p(\mathbf{Y} \mid \mathbf{X}, \boldsymbol{a}) &= \prod_{t=1}^{T} p(\boldsymbol{y}_t \mid \boldsymbol{y}_{1:t-1}, \boldsymbol{x}_{1:N}, \boldsymbol{a}) \\ &= \prod_{t=1}^{T} p(\boldsymbol{y}_t \mid \boldsymbol{h}_t) \end{aligned} \quad (1)$$

where $\boldsymbol{h}_t$ is the last hidden state of the generative model at decoding timestep t.

The SIDECONTROL framework consists of a large base network $B(\cdot)$ providing rich feature representations and a small side network $S(\cdot)$ encoding control attribute(s), as illustrated in Figure 1. The base network $B(\cdot)$ can be any pre-trained language models (Zhang et al., 2020; Roller et al., 2020; Shuster et al., 2020). Given dialogue context $\boldsymbol{x}_{1:N}$ as the input to the base network, we just take last hidden states $\{\boldsymbol{h}_b^t\}_{t=1}^{T}$ for the response $\{\boldsymbol{y}_t\}_{t=1}^{T}$ from the base network as our base representations:

$$\boldsymbol{h}_b^1, \ldots, \boldsymbol{h}_b^T = B(\boldsymbol{x}_{1:N}) \quad (2)$$

The side network $S(\cdot)$ is a light-weight neural network, which encodes the control attribute $\boldsymbol{a}$ into base representations $\{\boldsymbol{h}_b^t\}_{t=1}^{T}$:

$$\boldsymbol{h}_s^1, \ldots, \boldsymbol{h}_s^T = S(\boldsymbol{a}, \boldsymbol{h}_b^1, \ldots, \boldsymbol{h}_b^T) \quad (3)$$

Finally, we keep the base representation $\boldsymbol{h}_b^t$ fixed, and add the side representation $\boldsymbol{h}_s^t$ upon it to obtain the final combined representation $\boldsymbol{h}_t$ for the current token $\boldsymbol{y}_t$:

$$\boldsymbol{h}_t = \alpha \cdot \boldsymbol{h}_b^t + (1 - \alpha) \cdot \boldsymbol{h}_s^t \quad (4)$$

$$p(\boldsymbol{y}_t \mid \boldsymbol{h}_t) = \text{softmax}(\mathbf{W}_{\text{vocab}} \boldsymbol{h}_t) \quad (5)$$

where $\mathbf{W}_{\text{vocab}}$ is learnable parameters, and the mixture coefficient α is also learned during training, which aims to encode both useful prior knowledge from pre-trained language models and important attribute information from target dataset for controlled generation. We provide detailed implementations for the side network $S(\cdot)$ and mixture coefficient α in subsection 2.3 and subsection 2.2.

The *main challenge* in this framework is to teach the side network $S(\cdot)$, such that it can provide complementary information of control signals via $\boldsymbol{h}_s^t$ during generation, since the pre-trained language models can already generate fluent responses. To address this challenge, we intentionally freeze the parameters of the base network $B(\cdot)$ when training the side network. Otherwise, it is essentially training a large neural network model even deeper than $B(\cdot)$. Second, we introduce the control attribute loss $\mathcal{L}_{control}$, which is designed to teach the side network explicitly encoding control signals to improve the controllability of the model. The final objective is a combination of class-conditional language modelling loss $\mathcal{L}_{cclm}$ and task-specific control attributes loss $\mathcal{L}_{control}$:

$$\mathcal{L} = \mathcal{L}_{cclm} + \lambda \cdot \mathcal{L}_{control} \quad (6)$$

where λ is a task-specific hyper-parameter, and detailed implementations of $\mathcal{L}_{cclm}$ and $\mathcal{L}_{control}$ are described in subsection 2.2 and subsection 2.3, $\mathcal{L}_{control}$ has different implementation when controlling different forms of attributes.

2.2 Knowledge Document Control

When having external knowledge documents as the control attributes, such as persona profile (Dinan et al., 2020), Wikipedia articles (Dinan et al., 2018), etc., the format of control attribute is sequences of tokens $\boldsymbol{a} = \{\boldsymbol{k}_i\}_{i=1}^{K}$, where K is the total number of tokens in the external knowledge document. In this case, we model the knowledge document representation with a single-layer bi-directional LSTM:

$$\boldsymbol{h}_k^1, ..., \boldsymbol{h}_k^K = \text{BiLSTM}(\boldsymbol{a}) \quad (7)$$

The side network is designed to align the controlled knowledge document representation $\{\boldsymbol{h}_k^i\}_{i=1}^{K}$ with

the base representation $\boldsymbol{h}_b^t$ at each decoding timestep. We compute the cross-attention between $\{\boldsymbol{h}_k^i\}_{i=1}^K$ and $\boldsymbol{h}_b^t$ following (Bahdanau et al., 2014):

$$e_i^t = v^T \cdot \tanh\left(\mathbf{W}_k \boldsymbol{h}_k^i + \mathbf{W}_b \boldsymbol{h}_b^t + b_{kb}\right) \quad (8)$$

$$a_i^t = \text{softmax}(e_i^t) \quad (9)$$

$$\boldsymbol{c}_k^t = \sum_{i=1}^{K} a_i^t \cdot \boldsymbol{h}_k^i \quad (10)$$

where $\mathbf{W}_k \in \mathbb{R}^{D \times D}$, $\mathbf{W}_b \in \mathbb{R}^{D \times D}$ and $b_{kb} \in \mathbb{R}^D$ are learnable parameters. The attention a^t is a probability distribution over the controlled knowledge document that tells the decoder where to look at when generating the next word, and the context vector $\boldsymbol{c}_k^t$ represents what has been read from the controlled knowledge document representation at decoding timestep t. The final side representation $\boldsymbol{h}_s^t$ incorporates the context vector $\boldsymbol{c}_k^t$ into the base representation $\boldsymbol{h}_b^t$:

$$\boldsymbol{h}_s^t = \tanh\left(\mathbf{W}_c[\boldsymbol{c}_k^t; \boldsymbol{h}_b^t] + b_c\right) \quad (11)$$

where we concatenate $\boldsymbol{c}_k^t$ and $\boldsymbol{h}_b^t$, and $\mathbf{W}_c \in \mathbb{R}^{2D \times D}$ and $b_c \in \mathbb{R}^D$ are learnable parameters.

Since the controlled knowledge document is different per utterance, we implement the mixture coefficient α based on the side representation $\boldsymbol{h}_s^t$ and base representation $\boldsymbol{h}_b^t$ at decoding timestep t:

$$\alpha_t = \sigma(\mathbf{W}_\alpha[\boldsymbol{h}_s^t; \boldsymbol{h}_b^t] + b_\alpha) \quad (12)$$

$$\boldsymbol{h}_t = \alpha_t \cdot \boldsymbol{h}_b^t + (1 - \alpha_t) \cdot \boldsymbol{h}_s^t \quad (13)$$

where we concatenate $\boldsymbol{h}_s^t$ and $\boldsymbol{h}_b^t$, and $\mathbf{W}_\alpha \in \mathbb{R}^{2D \times 1}$ and $b_\alpha \in \mathbb{R}$ are learnable parameters.

In order to encourage the decoder generating more words from the knowledge document, we adopt the copy mechanism from (See et al., 2017) to formulate $\mathcal{L}_{cclm}$:

$$\beta = \sigma(\mathbf{W}_\beta[\boldsymbol{c}_k^t; \boldsymbol{h}_b^t] + b_\beta) \quad (14)$$

$$p(\boldsymbol{y}_t \mid \boldsymbol{h}_t) = \beta p(\boldsymbol{y}_t|\boldsymbol{h}_t) + (1 - \beta) \sum_{i=1}^{K} a_i^t \quad (15)$$

$$\mathcal{L}_{cclm} = -\sum_{t=1}^{T} \log p(\boldsymbol{y}_t^* \mid \boldsymbol{h}_t) \quad (16)$$

where we concatenate $\boldsymbol{c}_k^t$ and $\boldsymbol{h}_b^t$, and $\mathbf{W}_\beta \in \mathbb{R}^{2D \times 1}$ and $b_\beta \in \mathbb{R}$ are learnable parameters. $\boldsymbol{h}_t$ comes from Equation 13. $\boldsymbol{y}_t^*$ is the ground-truth word at decoding timestep t. $\sum_{i=1}^{K} a_i^t$ is the summation of attention distribution over the knowledge document at current decoding timestep t, which will assign higher probability for attended knowledge document words in the final word probability distribution.

The control attributes loss for this task is used to encourage generating more non-repetitive words from the knowledge document. We adopt the coverage mechanism from (See et al., 2017) to formulate $\mathcal{L}_{control}$:

$$\mathcal{L}_{control} = \sum_{t=1}^{T} \sum_{i=1}^{K} \min(a_i^t, \sum_{t'=0}^{t-1} a_i^{t'}) \quad (17)$$

where $a_i^{t'}$ is the attention weight of knowledge document word $\boldsymbol{k}_i$ at previous decoding time step t'. $\mathcal{L}_{control}$ penalizes the overlap between current attention distribution and previous attention distributions, which prevents the model repeatedly attending to the same word in the knowledge document. For more details about the copy mechanism and coverage mechanism, please refer to the original paper (See et al., 2017).

2.3 Semantic Label Control

When having a semantic label as the control attribute, such as dialogue act (Li et al., 2017), emotion (Rashkin et al., 2019), etc., we implement the side network as a simple feed-forward neural network:

$$\boldsymbol{h}_s^t = \tanh\left(\mathbf{W}_d[\mathbf{W}_a \boldsymbol{a}; \boldsymbol{h}_b^t] + b_d\right) \quad (18)$$

$$\boldsymbol{h}_t = \alpha \cdot \boldsymbol{h}_b^t + (1 - \alpha) \cdot \boldsymbol{h}_s^t \quad (19)$$

$$\mathcal{L}_{cclm} = -\sum_{t=1}^{T} \log p(\boldsymbol{y}_t^* \mid \boldsymbol{h}_t) \quad (20)$$

where we concatenate $\mathbf{W}_a \boldsymbol{a}$ and $\boldsymbol{h}_b^t$, $\mathbf{W}_a \in \mathbb{R}^{1 \times D}$ is an embedding matrix that maps the discrete label $\boldsymbol{a}$ to a continuous representation, $\mathbf{W}_d \in \mathbb{R}^{2D \times D}$ and $b_d \in \mathbb{R}^D$ are learnable parameters. The mixture coefficient $\alpha \in [0, 1]$ is a global parameter which is learned during training, in order to encode both useful prior knowledge from pre-trained language models and control signals from semantic label. $\boldsymbol{y}_t^*$ is the ground-truth word at decoding timestep t.

The control attributes loss $\mathcal{L}_{control}$ for this task is used to modify the final latent representations so that the model can generate responses with the target control attribute. However, it is difficult to directly measure how much control attribute information has been encoded into the side representation. Therefore, we approximate it using a independent

attribute classifier $p(\boldsymbol{a}|\boldsymbol{h}_{1:T})$. When training the side network, we keep the attribute classifier fixed and feed the side representations $\{\boldsymbol{h}_s^t\}_{t=1}^T$ into the classifier. The classifier will return a loss between the current side representation and the target control attribute $\boldsymbol{a}^*$, and optimizing this loss will update the side representation $\boldsymbol{h}_s^t$ towards obtaining a higher $p(\boldsymbol{a}^*|\boldsymbol{h}_{1:T})$:

$$p(\boldsymbol{a} \mid \boldsymbol{h}_{1:T}) = \text{softmax}(\mathbf{W}_{\text{clf}} \frac{\sum_{t=1}^T \boldsymbol{h}_s^t}{T}) \quad (21)$$

$$\mathcal{L}_{control} = -\log p(\boldsymbol{a}^* \mid \boldsymbol{h}_{1:T}) \quad (22)$$

Note that $\mathbf{W}_{\text{clf}} \in \mathbb{R}^{D \times K}$ is independently learned on the same training set based on the base representation $\{\boldsymbol{h}_b^t\}_{t=1}^T$, but is fixed when we update the side network.

3 Experiments

3.1 Evaluation Methods

In this work, we focus on evaluating the controllability and text quality of different controlled generation methods. Additionally, we prefer to have lower decoding cost and better modularity in order to apply the proposed method into more possible applications. Therefore, we use the following automatic metrics to evaluate the performance:

Controllability[2]: this is our main metric. It aims at evaluating whether the proposed method can successfully generate the target controlling attributes.

1. For the semantic label control task, we use the *classification accuracy* computed by an independently trained BERT classifier (Devlin et al., 2019).

2. For the knowledge document control task, we use the *cosine similarity* between the word vectors of external knowledge document and model generated response. We use the pre-trained GloVe embedding (Pennington et al., 2014) to model the word vectors.

Text Quality: it aims at evaluating how well the model learns to generate responses that match the ground-truth references, where we use model perplexity (PPL) computed on the test set [3], BLEU (Papineni et al., 2002) and METEOR (Banerjee and Lavie, 2005) to approximate it.

Decoding Cost: it evaluates the generation efficiency of the proposed method. Given the same set of 10 dialogue contexts, we compute the decoding time per token across different methods, a faster decoding time indicates the method is more efficient during generation.

Modularity: it evaluate how well the side network can be applied to different base networks. We compare model performance under two different types of pre-trained language models: DialoGPT (Zhang et al., 2020) and BlenderBot (Roller et al., 2020). Ideally, we expect as good or even better performance when switching the base network from DialoGPT to BlenderBot, since BlenderBot has been trained on larger dialogue corpus that is likely to provide more informative base representations.

3.2 Competitive Baselines

We compare the SIDECONTROL framework with the following competitive baselines:

DialoGPT-Ori: the original pre-trained language model for open-domain dialogue generation, DialoGPT (Zhang et al., 2020) DialoGPT is a Transformer-based language model. It is the baseline for all other controlled generation methods.

DialoGPT-FT: direct fine-tuning the DialoGPT on the target dialogue dataset. It is used as a strong baseline for evaluating the generation quality of the generative model.

DialoGPT-PPLM: the Plug-and-Play Language Model (PPLM) (Dathathri et al., 2019) with DialoGPT as the base pre-trained language model. It is a strong gradient-based baseline.

DialoGPT-FUDGE: the Future Discriminators for Generation (FUDGE) (Yang and Klein, 2021) with DialoGPT as the base pre-trained language model. It is a strong weighted decoding baseline.

DialoGPT-SideControl: our SIDECONTROL framework with DialoGPT as the base pre-trained language model. It is used to validate the effectiveness of our side network compared with other controlled generation baselines.

BlenderBot-Ori: the original BlenderBot (Roller et al., 2020), which is a Transformer-based sequence-to-sequence model showing state-of-the-art performance on some challenging open-domain dialogue datasets.

BlenderBot-SideControl: our SIDECONTROL framework with BlenderBot as the base pre-trained

[2] We provide implementation details in Appendix A

[3] Note that PPLM and FUDGE do not update the generative model and are applied only during generation, which means their model perplexity will be the same with their base network, i.e. DialoGPT-Ori, therefore we do not report their model perplexity in performance results.

	Controllability	Text Quality				Decoding Cost
METHOD	SIMILARITY ↑	PPL ↓	BLEU-1 ↑	BLEU-2 ↑	METEOR ↑	TIME ↓
DialoGPT-Ori	0.6382	68.63	12.95	1.22	0.0526	0.0786 s/tok
DialoGPT-FT	0.6732	15.22	17.27	2.05	0.0675	0.0721 s/tok
DialoGPT-FUDGE	0.6684	-	10.26	0.60	0.0514	0.0510 s/tok
DialoGPT-PPLM	0.6858	-	11.30	0.94	0.0646	0.5208 s/tok
DialoGPT-SideControl	**0.7526**	**14.34**	13.46	1.96	0.0988	0.0824 s/tok
BlenderBot-Ori	0.7455	90.89	9.38	0.54	0.0908	0.0384 s/tok
BlenderBot-SideControl	**0.7841**	**14.34**	10.10	1.20	0.0964	0.0608 s/tok

Table 1: Knowledge document control performances under full training set of ConvAI2, where $\lambda = 10^{-5}$ for $\mathcal{L}_{control}$ in DialoGPT-SideControl and BlenderBot-SideControl.

	Controllability	Text Quality				Decoding Cost
METHOD	ACCURACY ↑	PPL ↓	BLEU-1 ↑	BLEU-2 ↑	METEOR ↑	TIME ↓
DialoGPT-Ori	0.4307	55.09	7.78	0.66	0.0333	0.0921 s/tok
DialoGPT-FT	0.4358	8.95	14.35	2.30	0.0523	0.0786 s/tok
DialoGPT-FUDGE	0.4701	-	14.40	1.59	0.0411	0.0535 s/tok
DialoGPT-PPLM	0.5994	-	14.22	1.25	0.0506	2.4171 s/tok
DialoGPT-SideControl	0.5376	12.79	16.37	1.90	0.0526	0.0990 s/tok
BlenderBot-Ori	0.4605	110.05	12.21	1.10	0.0775	0.0603 s/tok
BlenderBot-SideControl	**0.6865**	**8.16**	14.49	1.36	0.0680	0.0995 s/tok

Table 2: Semantic label control performances under full training set of DailyDialog, where $\lambda = 10^{5}$ for $\mathcal{L}_{control}$ in DialoGPT-SideControl and BlenderBot-SideControl.

language model. It is used to show the high modularity of our side network.

3.3 Knowledge Document Control

In this task, given the previous dialogue context and the external knowledge document for the current speaker, the model will generate one utterance that is relevant both to the context and to the knowledge document. We provide the detailed experiment setups in Appendix B.

Dataset. We use the ConvAI2 dataset (Dinan et al., 2020) for the knowledge document control task. We set the previous 4 utterances as the dialogue context. Each utterance is linked to its corresponding persona profile. Since the test set of ConvAI2 has not been made public, we use the original training set to construct our training set, and split the first 80% original validation set as our validation set and the remaining 20% original validation set as our testing set. In total, we have 153,082 training samples, 38,271 validation samples and 11,590 testing samples.

Performances under Full Data. Table 1 shows that DialoGPT+SideControl outperforms all other baselines in controllability, which validates the effectiveness of the SIDECONTROL framework. For

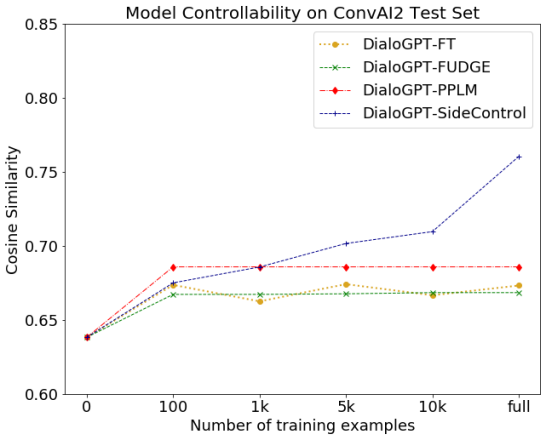


Figure 2: Controllability under different number of training examples in ConvAI2 dataset.

the quality of the generated texts, we find that both FUDGE and PPLM perform worse than the original pre-trained language model, while the SIDECONTROL shows improved quality because of the $\mathcal{L}_{cclm}$ during training. We also notice that direct fine-tuning gives the best performance in BLEU-1 and BLEU-2, but worse controllability compared with the SIDECONTROL. This is because direct fine-tuning only focuses on optimizing the language modelling loss, and does not take the control attributes information into account. For the decod-

ing cost, our SIDECONTROL is around 6x faster than PPLM during generation, which shows its efficiency during inference. Finally, we find that the performance improvement in controllability and text quality also hold when we apply the SIDECONTROL to BlenderBot, which shows the flexible modularity of the side network.

Performances under Small Data. With the goal of testing the sample-efficiency of the SIDECONTROL framework, we train all baselines under smaller datasets, where we randomly sample 100, 1000, 5000 and 10000 training samples from the original training set to train the model, and evaluate the model performance using the full testing test. Figure 2 shows the controllability performance under different training sizes, and we provide detailed text quality performance in Appendix E. We find that SIDECONTROL only underperforms PPLM in 100 training samples, since PPLM uses non-parametric bag-of-words features as its attribute model while SIDECONTROL uses a BiLSTM as its attribute model. And 1000 training samples are sufficient enough for SIDECONTROL to achieve comparable performance with PPLM. In addition, SIDECONTROL constantly achieves performance improvement when increasing the training size.

Ablation Study. To verify the effectiveness of the control loss $\mathcal{L}_{control}$, we conduct ablation study by trying out different values of λ in Equation 6. We provide partial results in Table 3 and full results in Appendix D. When $\lambda = 0$, the model becomes a vanilla language model and takes no information from the side network, which leads to a low performance in controllability. When $\lambda \neq 0$, the model incorporates control attributes information from the side network, which leads to an improved performance in controllability. However, incorporating side information will lead to a slight increase in model perplexity.

3.4 Semantic Label Control

In this task, given the previous dialogue context and the current dialogue act, the model will generate one utterance that is relevant to the context and also satisfies the current dialogue act. We provide the detailed experiment setups in Appendix C.

Dataset. We use the DailyDialog dataset (Li et al., 2017) for the semantic label control task. We set the previous 5 utterances as the dialogue context and follow the standard train/validation/test splition of the original dataset to construct our generation dataset. In total, we obtain 35,781 training samples, 3,388 validation samples and 3,123 testing samples.

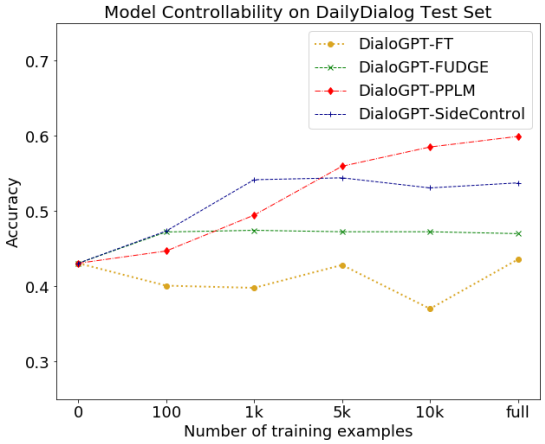


Figure 3: Controllability under different number of training examples in DailyDialog dataset.

Performances under Full Data. Table 2 demonstrates that SIDECONTROL has better text quality than FUDGE and PPLM, since we explicitly optimize $\mathcal{L}_{cclm}$ during training. For the controllability, PPLM achieves the best performance with a sacrifice of inference efficiency, while SIDECONTROL can achieve comparable performance in controllability with around 24x faster decoding time. Finally, the performance improvements in controllability and text quality still hold when we switch the base network from DialoGPT to BlenderBot, which demonstrates that the side network is flexible to be applied to different types of pre-trained language models. And surprisingly, BlenderBot can even provide the state-of-the-art performance in controllability.

Performances under Small Data. We also compare across the model performance under different training sizes following the same setup with the knowledge document control task, and provide detailed text quality performance in Appendix F. Figure 3 illustrates that SIDECONTROL achieves better controllability than PPLM when training size is under 1000. This is because PPLM uses a data-driven classifier as its attribute model in this task, and its attribute model gets overfitted on the 100 training samples, which results in poor controllability performance. Similarly, FUDGE has the same overfitting issue for its attribute discriminator on these small training sets, and gets unsatisfied

λ	SIMILARITY ↑	PPL ↓
$\lambda = 0$	0.7273	**14.24**
$\lambda = 10^{-4}$	0.7306	14.65
$\lambda = 10^{-5}$	**0.7526**	14.34

Table 3: Ablation study of DialoGPT-SideControl on knowledge document control, where the model is trained under the full training set of ConvAI2, and tested under the full testing set of ConvAI2.

λ	ACCURACY ↑	PPL ↓
$\lambda = 0$	0.4950	**12.37**
$\lambda = 10^{3}$	0.5232	12.59
$\lambda = 10^{5}$	**0.5376**	12.79

Table 4: Ablation study of DialoGPT-SideControl on semantic label control, where the model is trained under the full training set of DailyDialog, and tested under the full testing set of DailyDialog.

METHOD	FLUENCY ↑	RELEVANCY ↑
DialoGPT-PPLM	3.832	3.188
DialoGPT-FUDGE	4.016	3.348
DialoGPT-SideControl	**4.108**	**3.816**

Table 5: Human evaluation of fluency and context relevancy on semantic label control task.

	Wins %		
	PPLM	FUDGE	SideControl
PPLM	-	55.25%	**57.61%**
FUDGE	44.76%	-	**54.46%**
SideControl	42.39%	45.54%	-

Table 6: Human evaluation of attribute relevancy on semantic label control task.

controllability performance. Although SIDECONTROL also pre-trains a classifier on the 100 training samples to guide the update of side representation, its final representation is a combination of base and side representation. We believe incorporating prior knowledge from the base representation helps SIDECONTROL alleviates the overfitting issue on small training set.

Ablation Study. We also try out different values of λ to study the effect of control loss $\mathcal{L}_{control}$, as shown in Table 4. Full ablation study results are provided in Appendix D. When $\lambda = 0$, the model takes no control attributes signals from the side network during training, which results in a low controllability performance. When $\lambda \neq 0$, the controllability performance of the model is improved but with a slight increase in model perplexity. Both Table 3 and Table 4 verify the effectiveness of control loss $\mathcal{L}_{control}$ in improving the controllability of pre-trained language models.

Human Evaluation. To validate the good performance of SIDECONTROL, we follow prior works (Dathathri et al., 2019; Li et al., 2019) and deploy a set of human evaluations to compare the text quality and controllability between several methods. For the text quality, we ask human annotators to evaluate the fluency and context relevancy of the generated responses on a scale of 1-5, where a higher score indicates better quality. For the controllability, we use A/B testing following (Li et al., 2019) and compare all model pairs (e.g. PPLM vs. SIDECONTROL) [4]. For all human evaluations, we randomly sample 50 dialogue contexts, and collect the corresponding model generated responses. Human annotators are recruited using Amazon Mechanical Turk and each response has 5 annotations. In total, we collect 2250 human annotations. Table 5 shows the results of text quality evaluation, and SIDECONTROL achieves the best fluency and context relevancy than PPLM and FUDGE. Table 6 shows the results of controllability evaluation, and SIDECONTROL wins over PPLM and FUDGE in 57% and 54% respectively. Both text quality and controllability evaluation show that SIDECONTROL can generate more fluent, context-relevant and attribute-relevant responses than PPLM and FUDGE.

4 Related Works

There are three major categories of controllable text generation models: class-conditional language model (Keskar et al., 2019; Kawano et al., 2019), plug-and-play language model (Dathathri et al., 2019) and weighted decoding (Ghazvininejad et al., 2017; Baheti et al., 2018; Holtzman et al., 2018; Yang and Klein, 2021).

Class-Conditional Language Model. Class-conditional language models train a conditional generative model from scratch, and guide the generation with explicit control codes provided in the training data. Keskar et al. (2019) trains a 1.63 billion-parameter Conditional Transformer Language (CTRL) model by prepending control codes in front of raw texts. But training the CTRL

[4] We show the same dialogue context, current dialog act and two responses generated by model A and model B respectively, and ask human annotators to select the response which is more related to the current dialog act among: model A, model B, both and neither.

(Keskar et al., 2019) requires 140 GB of training data, which may not be affordable for some low-resource languages. Kawano et al. (2019) builds a controllable neural conversation model by leveraging an adversarial learning framework that alternatively trains between a class-conditional language model and a multi-class discriminator, where the discriminator is used to help the generative model produce responses with appropriate dialogue act. But the control code is modelled as discrete variable in this work, which limits the controllability capacity of the dialogue model.

Plug-and-Play Language Model. Guiding generation with gradients from additional attribute models is another popular approach. Dathathri et al. (2019) introduce a plug-and-play language model (PPLM) which combines the pre-trained language model $p(x)$ with attribute models $p(a|x)$ to approximate the contional generative model $p(x|a)$. At each decoding timestep, all hidden representations of the pre-trained language model are shifted with gradients towards a higher $p(x|a) \propto p(a|x)p(x)$. The attribute models of PPLM are either in the form of bag-of-words or single layer classifiers, which requires much less training data than learning a conditional generative model. The following works (Goswamy et al., 2020; Lin and Riedl, 2021; Madotto et al., 2020) further propose more fine-grained attribute models and generation strategies for specific task, such as emotional text generation (Goswamy et al., 2020), story generation (Lin and Riedl, 2021) and conversation generation (Madotto et al., 2020). But since the plug-and-play language models have to compute gradient from attribute model and update hidden representations at each decoding timestep, the generation process is very time-consuming, which leads to high decoding cost.

Weighted Decoding. Weighted decoding runs a more expensive beam search where the sampling probability distribution is altered by desired control attributes, such as topic, sentiment, etc. Ghazvininejad et al. (2017) design a set of style features on controlling topic, sentiment, and repetitive words, and re-compute the beam score of each token with a combination of the original beam score and the style feature score. A recent work (Yang and Klein, 2021) introduces a Future Discriminator for Generation (FUDGE) that trains a binary discriminator for the control attribute prediction and re-scores the probability distribution of the original pre-trained language model with the discriminator prediction via Bayesian factorization. The major limitation of weighted decoding methods is that, if the pre-trained language model is a high-bias estimator, which assigns low probability for desired attribute words and high probability for commonly observed but unrelated words, re-scoring or re-ranking such a "high-biased" distribution cannot guarantee the generation of desired attributes.

The SIDECONTROL framework differs the above methods as follows: (1) the side network only requires access to last hidden states of the base network. Both class-conditional language models (Keskar et al., 2019) and plug-and-play language models (Dathathri et al., 2019) require access to every hidden states of the pre-trained language model, which limits its application under certain pre-trained model. (2) the side network learns a residual on top of pre-trained language models, which is suitable for small datasets. Directly fine-tuning (Ziegler et al., 2019) large pre-trained language models will cause overfitting issues on some small datasets, and weighted-decoding methods (Ghazvininejad et al., 2017; Yang and Klein, 2021) only modify the final vocabulary distribution of pre-trained models but do not learn model parameters to better adapt to the target task.

5 Conclusions

In this work, we propose a new method for controlled dialogue generation: adding a small side network to incorporate useful control signals into the pre-trained language models. We design control attributes loss to teach the side network learning useful control signals. Empirical experiments show that our method is effective even with $100 \sim 1000$ training samples. Besides, our side network supports diverse forms of attributes control and can be flexibly applied to any pre-trained language models, which extends its possible application to other general controlled text generation tasks.

Acknowledgments

The authors thank the anonymous reviewers for their useful comments and the UVa NLP group for helpful discussions.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Ashutosh Baheti, Alan Ritter, Jiwei Li, and Bill Dolan. 2018. Generating more interesting responses in neural conversation models with distributional constraints. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3970–3980, Brussels, Belgium. Association for Computational Linguistics.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Siqi Bao, Huang He, Fan Wang, Hua Wu, and Haifeng Wang. 2020. PLATO: Pre-trained dialogue generation model with discrete latent variable. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 85–96, Online. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Wenhu Chen, Jianshu Chen, Pengda Qin, Xifeng Yan, and William Yang Wang. 2019. Semantically conditioned dialog response generation via hierarchical disentangled self-attention. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3696–3709, Florence, Italy. Association for Computational Linguistics.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2019. Plug and play language models: A simple approach to controlled text generation. *CoRR*, abs/1912.02164.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Emily Dinan, Varvara Logacheva, Valentin Malykh, Alexander Miller, Kurt Shuster, Jack Urbanek, Douwe Kiela, Arthur Szlam, Iulian Serban, Ryan Lowe, et al. 2020. The second conversational intelligence challenge (convai2). In *The NeurIPS'18 Competition*, pages 187–208. Springer.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2018. Wizard of wikipedia: Knowledge-powered conversational agents. In *International Conference on Learning Representations*.

Marjan Ghazvininejad, Xing Shi, Jay Priyadarshi, and Kevin Knight. 2017. Hafez: an interactive poetry generation system. In *Proceedings of ACL 2017, System Demonstrations*, pages 43–48, Vancouver, Canada. Association for Computational Linguistics.

Tushar Goswamy, Ishika Singh, Ahsan Barkati, and Ashutosh Modi. 2020. Adapting a language model for controlled affective text generation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2787–2801, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Ari Holtzman, Jan Buys, Maxwell Forbes, Antoine Bosselut, David Golub, and Yejin Choi. 2018. Learning to write with cooperative discriminators. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1638–1649, Melbourne, Australia. Association for Computational Linguistics.

Cheng-Hsun Hsueh and Wei-Yun Ma. 2020. Semantic guidance of dialogue generation with reinforcement learning. In *Proceedings of the 21th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 1–9, 1st virtual meeting. Association for Computational Linguistics.

Seiya Kawano, Koichiro Yoshino, and Satoshi Nakamura. 2019. Neural conversation model controllable by given dialogue act based on adversarial learning and label-aware objective. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 198–207, Tokyo, Japan. Association for Computational Linguistics.

Pei Ke, Jian Guan, Minlie Huang, and Xiaoyan Zhu. 2018. Generating informative responses with controlled sentence function. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1499–1508, Melbourne, Australia. Association for Computational Linguistics.

Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019.

CTRL: A conditional transformer language model for controllable generation. *CoRR*, abs/1909.05858.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization.

Margaret Li, Jason Weston, and Stephen Roller. 2019. ACUTE-EVAL: improved dialogue evaluation with optimized questions and multi-turn comparisons. *CoRR*, abs/1909.03087.

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

Zhiyu Lin and Mark Riedl. 2021. Plug-and-blend: A framework for controllable story generation with blended control codes. *CoRR*, abs/2104.04039.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization.

Andrea Madotto, Etsuko Ishii, Zhaojiang Lin, Sumanth Dathathri, and Pascale Fung. 2020. Plug-and-play conversational models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2422–2433, Online. Association for Computational Linguistics.

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

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

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2019. Towards empathetic open-domain conversation models: A new benchmark and dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5370–5381, Florence, Italy. Association for Computational Linguistics.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Kurt Shuster, Eric M Smith, et al. 2020. Recipes for building an open-domain chatbot. *arXiv preprint arXiv:2004.13637*.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Abigail See, Stephen Roller, Douwe Kiela, and Jason Weston. 2019. What makes a good conversation? how controllable attributes affect human judgments. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1702–1723, Minneapolis, Minnesota. Association for Computational Linguistics.

Kurt Shuster, Da Ju, Stephen Roller, Emily Dinan, Y-Lan Boureau, and Jason Weston. 2020. The dialogue dodecathlon: Open-domain knowledge and image grounded conversational agents. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2453–2470, Online. Association for Computational Linguistics.

Junya Takayama and Yuki Arase. 2020. Consistent response generation with controlled specificity. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4418–4427, Online. Association for Computational Linguistics.

Deeksha Varshney, Asif Ekbal, and Pushpak Bhattacharyya. 2021. Modelling context emotions using multi-task learning for emotion controlled dialog generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2919–2931, Online. Association for Computational Linguistics.

Tsung-Hsien Wen, Milica Gašić, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned LSTM-based natural language generation for spoken dialogue systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1711–1721, Lisbon, Portugal. Association for Computational Linguistics.

Kevin Yang and Dan Klein. 2021. FUDGE: Controlled text generation with future discriminators. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3511–3535, Online. Association for Computational Linguistics.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing

Liu, and Bill Dolan. 2020. DIALOGPT : Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278, Online. Association for Computational Linguistics.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul F. Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *CoRR*, abs/1909.08593.

A Automatic Metrics for Controllability Evaluation

In this section we provide implementation details for how we compute classification accuracy and cosine similarity.

A.1 Dialogue Act Classifier

We train an independent dialogue act classifier to evaluate whether the current generated response matches its conditioning dialogue act. The input to the evaluation dialogue act classifier is a single response, and the output is a prediction of one of the 4 dialogues in DailyDialog, i.e. *inform*, *questions*, *directives* and *commissive*.

We construct the training corpus following the standard splition of original DailyDialog dataset, and obtain 87,170 training samples, 8,069 validation samples and 7,740 testing samples. We leverage the BERT model to provide a sequence of word representations and add a single-layer feed-forward neural network to predict the dialogue act of current sentence. We use AdamW (Loshchilov and Hutter, 2019) with learning rate 0.0001 to train this classifer. We set the batch size to 16, the total training epoch to 10 and automatically evaluate the model on the validation set very 5000 iterations. We save the model checkpoint with the lowest validation loss as the optimal model.

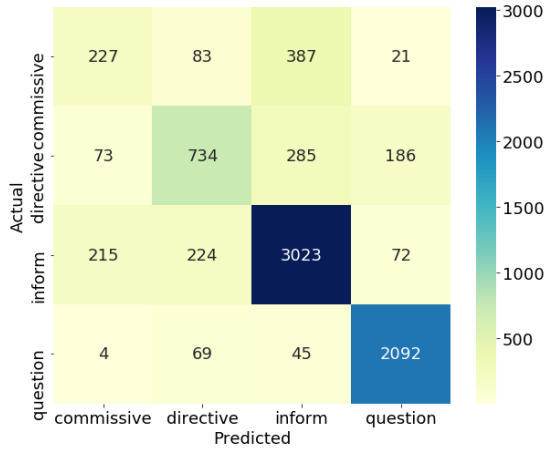


Figure 4: Confusion matrix of the evaluation dialogue act classifier.

This dialogue act classifier achieves 0.79 accuracy on the test set. Figure 4 shows the confusion matrix of this dialogue act classifier.

A.2 Computation of Cosine Similarity

To measure the similarity between the generated response and the conditioning knowledge document, we compute the cosine similarity between the wording embeddings of generated response and external knowledge document. The word embeddings are GloVe embeddings (Pennington et al., 2014) pretrained on Wikipedia 2014 and Gigaword 5, which are 100-dimension vectors and have 6 billion tokens [5].

We use the NLTK word tokenizer [6] to tokenize the texts into a set of tokens, and remove stop words based on a pre-defined stop words list in (Bao et al., 2020). Finally, we compute the cosine similarity between the two sets of word vectors.

B Experiment Setups for Knowledge Document Control

We conduct all of our experiments on single GeForce RTX 2080Ti GPU server with 11019 MB memory.

B.1 Direct Fine-tuning

We directly update all parameters of the pre-trained language model on the ConvAI2 training set without having any side network or control attributes loss. For the training of the pre-trained language model, we use AdamW (Loshchilov and Hutter, 2019) with learning rate 0.0001. We set the batch size to 2, the total training epoch to 10, and automatically evaluate the model on the validation set every 1000 iterations. We save the model checkpoint which achieves lowest validation loss as the final optimal model. For generation, we follow the setup of FUDGE, which use top-k sampling with $k = 10$.

B.2 PPLM

For the implementation of the attribute model, we use the bag-of-words attribute model proposed in the original paper (Dathathri et al., 2019) to encode external knowledge document. We run the model on the ConvAI2 dataset using the code provided by the original paper: https://github.com/uber-research/PPLM. We set the maximum generation length to 50, the number of gradient update steps to 3, the step size to 0.03, the window length to 5, the number of generated sentences to 1, $\gamma_{gm} = 0.99$, $\lambda_{KL} = 0.01$.

[5] https://nlp.stanford.edu/projects/glove/

[6] https://www.nltk.org/_modules/nltk/tokenize.html

B.3 FUDGE

For the implementation of the attribute model, we use the bag-of-words attribute model proposed in the original paper (Yang and Klein, 2021) to encode external knowledge document. We run the model on the ConvAI2 dataset using the code provided by the original paper: https://github.com/yangkevin2/naacl-2021-fudge-controlled-generation. We set the maximum generation length to 80, the weight on conditioning model to 4.0, consider top 200 outputs from DialoGPT at each decoding timestep before conditioning, and sample from top 10 outputs from DialoGPT at each decoding timestep.

B.4 SideControl

For the implementation of the side network, we use a single-layer bi-LSTM which shares the same hidden dimension with the final hidden states of the base network. We tokenize the knowledge document using the same tokenizer with the base network, and share the same word embedding with the base network as well. For the training of the side network, we use AdamW (Loshchilov and Hutter, 2019) with learning rate 0.0001. We set the batch size to 4, the total training epoch to 10, and automatically evaluate the model on the validation set every 100 iterations. For the hyper-parameter λ of the coverage loss in Equation 17, we use grid search on the validation set to obtain the optimal number. We search from the set $\lambda = \{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 0.01, 0.1\}$ and find $\lambda = 10^{-5}$ yields best performance. For generation, we follow the setup of FUDGE, which use top-k sampling with $k = 10$.

C Experiment Setups for Semantic Label Control

We conduct all of our experiments on single GeForce RTX 2080Ti GPU server with 11019 MB memory.

C.1 Direct Fine-tuning

We directly update all parameters of the pre-trained language model on the DailyDialog training set without having any side network or control attributes loss. For the training of the pre-trained language model, we use AdamW (Loshchilov and Hutter, 2019) with learning rate 0.0001. We set the batch size to 2, the total training epoch to 10, and automatically evaluate the model on the validation set every 1000 iterations. We save the model checkpoint which achieves lowest validation loss as the final optimal model. For generation, we follow the setup of FUDGE, which use top-k sampling with $k = 10$.

C.2 PPLM

For the implementation of the attribute model, we follow the generic discriminator implementation in the original paper (Dathathri et al., 2019). We run the model on the DailyDialog dataset using the code provided by the original paper. We train a dialogue act classifier which takes single response as input and produces a prediction on one of the four dialogue acts. For the training of the classifier, we use Adam (Kingma and Ba, 2017) with learning rate 0.0001. We set the batch size to 64, the total training epoch to 10. For the generation of PPLM, we set the maximum generation length to 50, the number of gradient update steps to 10, the step size to 0.2, the number of generated sentences to 1, $\gamma_{gm} = 0.95$, $\lambda_{KL} = 0.01$.

C.3 FUDGE

For the implementation of the attribute model, we follow the attribute discriminator implementation in the original paper (Yang and Klein, 2021). We run the model on the DailyDialog dataset using the code provided by the original paper. We train a dialogue act discriminator which takes the dialogue context and the current response as input and produces a prediction on one of the four dialogue acts. For the training of the discriminator, we use Adam (Kingma and Ba, 2017) with learning rate 2×10^{-5}. We set the batch size to 16, the total training epoch to 10. For the generation of FUDGE, we set the maximum generation length to 60, the weight on conditioning model to 1.0, consider top 200 outputs from DialoGPT at each decoding timestep before conditioning, and sample from top 10 outputs from DialoGPT at each decoding timestep.

C.4 SideControl

For the implementation of the side network, we use a single-layer feed-forward neural network which shares the same hidden dimension with the final hidden states of the base network. Besides, we pre-trained a dialogue act classifier to compute the control loss in Equation 22. We emphasize that this dialogue act classifier is different from the evaluation classifier. It models the sentence representation

from the base network, i.e. DialoGPT, and adds a single-layer feed-forward neural network to predict the dialogue act of current response. We train this classifier using AdamW (Loshchilov and Hutter, 2019) with learning rate 0.0001 for 10 epochs. Then, we fix this classifier and begin to train the side network using AdamW (Loshchilov and Hutter, 2019) with learning rate 0.0001 for another 10 epochs. We evaluate the model on the validation set every 1000 iterations, and save the model checkpoint which has the lowest validation loss. For the hyper-parameter λ of the control loss in Equation 22, we use grid search on the validation set to obtain the optimal number. We search from the set $\lambda = \{1, 10, 100, 10^3, 10^4, 10^5, 10^6\}$ and find $\lambda = 10^5$ yields best performance on the full training set. For generation, we follow the setup of FUDGE, which use top-k sampling with $k = 10$.

D Full performances of Ablation Study

We provide performance details for ablation study in knowledge document control and semantic label control. The full performances of ablation study in knowledge document control is shown in Table 7. The full performances of ablation study in semantic label control is shown in Table 8.

E Full performances of Knowledge Document Control under Different Number of Training Samples

For all experiments across different number of training samples, we take the hyper-parameter $\lambda = 10^{-5}$ for $\mathcal{L}_{control}$. Full performance for all models are demonstrated in Table 9, Table 11, Table 13 and Table 15. We also provide some generated samples from the test set for reference, demonstrated in Table 10, Table 12, Table 14, Table 16, Table 17.

F Full performances of Semantic Label Control under Different Number of Training Samples

For the semantic label control task, we find the optimal hyper-parameter λ for $\mathcal{L}_{control}$ differs across different number of training samples. Full performance for all models are demonstrated in Table 18, Table 20, Table 22 and Table 24. We also provide some generated samples from the test set for reference, demonstrated in Table 19, Table 21, Table 23, Table 25, Table 26.

	Controllability	Text Quality			
	SIMILARITY ↑	PERPLEXITY ↓	BLEU-1 ↑	BLEU-2 ↑	METEOR ↑
$\lambda = 0$	0.7273	**14.24**	15.72	2.16	0.0858
$\lambda = 10^{-6}$	0.7284	14.30	16.08	2.29	0.0800
$\lambda = 10^{-5}$	**0.7526**	14.34	13.46	1.96	0.0988
$\lambda = 10^{-4}$	0.7306	14.65	15.87	2.32	0.0846
$\lambda = 10^{-3}$	0.7259	15.65	15.72	2.09	0.0802
$\lambda = 10^{-2}$	0.7217	30.29	15.30	2.05	0.0803
$\lambda = 10^{-1}$	0.7137	22481.68	15.50	2.01	0.0774

Table 7: Knowledge document control performances of DialoGPT-SideControl with different λ.

	Controllability	Text Quality			
	ACCURACY ↑	PERPLEXITY ↓	BLEU-1 ↑	BLEU-2 ↑	METEOR ↑
$\lambda = 0$	0.4950	**12.37**	16.19	1.95	0.0534
$\lambda = 10^{1}$	0.5229	12.48	15.06	1.76	0.0525
$\lambda = 10^{2}$	0.5366	12.51	15.59	1.76	0.0517
$\lambda = 10^{3}$	0.5232	12.59	15.59	1.75	0.0512
$\lambda = 10^{5}$	**0.5376**	12.79	16.37	1.90	0.0526
$\lambda = 10^{6}$	0.5357	13.10	15.29	1.67	0.0485

Table 8: Semantic label control performances of DialoGPT-SideControl with different λ.

	Controllability	Text Quality			
METHOD	SIMILARITY ↑	PERPLEXITY ↓	BLEU-1 ↑	BLEU-2 ↑	METEOR ↑
DialoGPT-FT	0.6736	180.65	13.04	1.32	0.0503
DialoGPT-FUDGE	0.6672	-	10.33	0.56	0.0530
DialoGPT-PPLM	0.6858	-	11.30	0.94	0.0646
DialoGPT-SideControl	0.6750	24.64	12.00	1.31	0.0553

Table 9: Knowledge document control performances under 100 training samples sampled from ConvAI2, where $\lambda = 10^{-5}$ for $\mathcal{L}_{control}$ in DialoGPT-SideControl.

Dialogue Context:	**Persona Profile**:
"hi how are you tonight ? i am relaxing in a bubble bath tonight ."	"my favorite season is winter."
"hi how are you doing that sounds great ."	"i won a gold medal in the 1992 olympics."
"it is , how old are you ? i'm 31 , feeling like i'm getting old lol."	"i love cats and have two cats."
"not old you are still young try 50 i'm getting old ."	"i've a hat collection of over 1000 hats."
"lol my bones suggest other wise . do you have any animals ?"	"i have a weakness for vanilla ice cream."

Ground Truth: "yes i like and love all animals i have two cats."
DialoGPT-Ori: "my dogs are my favorite, but i'm pretty fond of all dogs so i don't have many."
DialoGPT-FT: "i love my three dogs. "
DialoGPT-FUDGE: "My bones say I'm young too, so I think we're good there for now."
DialoGPT-PPLM: "I think i'm only 35, but i've been in a pretty good relationship so i've probably gotten better."
DialoGPT-SideControl: "!"

Table 10: Knowledge document control generation results under 100 training samples sampled from ConvAI2.

	Controllability	Text Quality			
METHOD	SIMILARITY ↑	PERPLEXITY ↓	BLEU-1 ↑	BLEU-2 ↑	METEOR ↑
DialoGPT-FT	0.6625	18.64	15.67	1.69	0.0628
DialoGPT-FUDGE	0.6672	-	10.33	0.56	0.0530
DialoGPT-PPLM	0.6858	-	11.30	0.94	0.0646
DialoGPT-SideControl	0.6857	19.32	15.88	1.98	0.0748

Table 11: Knowledge document control performances under 1000 training samples sampled from ConvAI2, where $\lambda = 10^{-5}$ for $\mathcal{L}_{control}$ in DialoGPT-SideControl.

Dialogue Context:	**Persona Profile**:
"hi how are you tonight ? i am relaxing in a bubble bath tonight ."	"my favorite season is winter."
"hi how are you doing that sounds great ."	"i won a gold medal in the 1992 olympics."
"it is , how old are you ? i'm 31 , feeling like i'm getting old lol."	"i love cats and have two cats."
"not old you are still young try 50 i'm getting old ."	"i've a hat collection of over 1000 hats."
"lol my bones suggest other wise . do you have any animals ?"	"i have a weakness for vanilla ice cream."

Ground Truth: "yes i like and love all animals i have two cats."
DialoGPT-Ori: "my dogs are my favorite, but i'm pretty fond of all dogs so i don't have many."
DialoGPT-FT: "cats and dogs. you?"
DialoGPT-FUDGE: "My bones say I'm young too, so I think we're good there for now."
DialoGPT-PPLM: "I think i'm only 35, but i've been in a pretty good relationship so i've probably gotten better."
DialoGPT-SideControl: "i don't, they don't get to play."

Table 12: Knowledge document control generation results under 1000 training samples sampled from ConvAI2.

	Controllability	**Text Quality**			
METHOD	SIMILARITY ↑	PERPLEXITY ↓	BLEU-1 ↑	BLEU-2 ↑	METEOR ↑
DialoGPT-FT	0.6741	24.46	15.65	1.17	0.0634
DialoGPT-FUDGE	0.6676	-	10.20	0.54	0.0528
DialoGPT-PPLM	0.6858	-	11.30	0.94	0.0646
DialoGPT-SideControl	0.7016	17.30	15.76	1.99	0.0764

Table 13: Knowledge document control performances under 5000 training samples sampled from ConvAI2, where $\lambda = 10^{-5}$ for $\mathcal{L}_{control}$ in DialoGPT-SideControl.

Dialogue Context:	**Persona Profile**:
"hi how are you tonight ? i am relaxing in a bubble bath tonight ."	"my favorite season is winter."
"hi how are you doing that sounds great ."	"i won a gold medal in the 1992 olympics."
"it is , how old are you ? i'm 31 , feeling like i'm getting old lol."	"i love cats and have two cats."
"not old you are still young try 50 i'm getting old ."	"i've a hat collection of over 1000 hats."
"lol my bones suggest other wise . do you have any animals ?"	"i have a weakness for vanilla ice cream."

Ground Truth: "yes i like and love all animals i have two cats."
DialoGPT-Ori: "my dogs are my favorite, but i'm pretty fond of all dogs so i don't have many."
DialoGPT-FT: "!!! i've 3 cats but they are all scared of me haha."
DialoGPT-FUDGE: "My bones say I'm young too, so I think we're good there for now."
DialoGPT-PPLM: "I think i'm only 35, but i've been in a pretty good relationship so i've probably gotten better."
DialoGPT-SideControl: "i do. i love them. what do you do for fun? goldfish?"

Table 14: Knowledge document control generation results under 5000 training samples sampled from ConvAI2.

	Controllability	**Text Quality**			
METHOD	SIMILARITY ↑	PERPLEXITY ↓	BLEU-1 ↑	BLEU-2 ↑	METEOR ↑
DialoGPT-FT	0.6666	17.78	14.86	1.56	0.0657
DialoGPT-FUDGE	0.6684	-	10.21	0.58	0.0529
DialoGPT-PPLM	0.6858	-	11.30	0.94	0.0646
DialoGPT-SideControl	0.7096	16.98	15.49	1.91	0.0774

Table 15: Knowledge document control performances under 10000 training samples sampled from ConvAI2, where $\lambda = 10^{-5}$ for $\mathcal{L}_{control}$ in DialoGPT-SideControl.

Dialogue Context:	**Persona Profile**:
"hi how are you tonight ? i am relaxing in a bubble bath tonight ."	"my favorite season is winter."
"hi how are you doing that sounds great ."	"i won a gold medal in the 1992 olympics."
"it is , how old are you ? i'm 31 , feeling like i'm getting old lol."	"i love cats and have two cats."
"not old you are still young try 50 i'm getting old ."	"i've a hat collection of over 1000 hats."
"lol my bones suggest other wise . do you have any animals ?"	"i have a weakness for vanilla ice cream."

Ground Truth: "yes i like and love all animals i have two cats."
DialoGPT-Ori: "my dogs are my favorite, but i'm pretty fond of all dogs so i don't have many."
DialoGPT-FT: "i have a dog! what do you do for work?"
DialoGPT-FUDGE: "My bones say I'm young too, so I think we're good there for now."
DialoGPT-PPLM: "I think i'm only 35, but i've been in a pretty good relationship so i've probably gotten better."
DialoGPT-SideControl: "i've a pet cat named monkey."

Table 16: Knowledge document control generation results under 10000 training samples sampled from ConvAI2.

Dialogue Context:	**Persona Profile**:
"hi how are you tonight ? i am relaxing in a bubble bath tonight ."	"my favorite season is winter."
"hi how are you doing that sounds great ."	"i won a gold medal in the 1992 olympics."
"it is , how old are you ? i'm 31 , feeling like i'm getting old lol."	"i love cats and have two cats."
"not old you are still young try 50 i'm getting old ."	"i've a hat collection of over 1000 hats."
"lol my bones suggest other wise . do you have any animals ?"	"i have a weakness for vanilla ice cream."

Ground Truth: "yes i like and love all animals i have two cats."
DialoGPT-Ori: "my dogs are my favorite, but i'm pretty fond of all dogs so i don't have many."
DialoGPT-FT: "my dogs and cats are old as well."
DialoGPT-FUDGE: "."
DialoGPT-PPLM: "I think i'm only 35, but i've been in a pretty good relationship so i've probably gotten better."
DialoGPT-SideControl: "! i do not, i love animals. i don't know how to have pets, i'm too busy. i have two cats. they are my best friend."

Table 17: Knowledge document control generation results under full training samples from ConvAI2.

	Controllability	**Text Quality**			
METHOD	ACCURACY ↑	PERPLEXITY ↓	BLEU-1 ↑	BLEU-2 ↑	METEOR ↑
DialoGPT-FT	0.4009	70.74	9.31	0.76	0.0364
DialoGPT-FUDGE	0.4723	-	14.59	1.59	0.0424
DialoGPT-PPLM	0.4470	-	11.11	0.57	0.0382
DialoGPT-SideControl	0.4736	30.33	11.17	0.77	0.0379

Table 18: Semantic label control performances under 100 training samples sampled from DailyDialog, where $\lambda = 1$ for $\mathcal{L}_{control}$ in DialoGPT-SideControl.

Dialogue Context:	**Dialogue Act**:
"Well , have you ever heard of Oulong tea and compressed tea ?"	"directive"
"Oh , yeah . Oulong tea is good for one's health . isn't it ?"	
"You surely know a lot about Chinese tea."	
"Sure , I like drinking tea at teahouses."	
"Oh , so do I."	

Ground Truth: "Why don't we go for one now ?"
DialoGPT-Ori: "I'm sure."
DialoGPT-FT: "yes. I have a large appetite. I like eating. I like cooking. I like shopping. I like to go to the markets. "
DialoGPT-FUDGE: "I'm not sure if you're being serious or not, but I'm pretty sure that's not what he's saying."
DialoGPT-PPLM: "I know that tea from somewhere, so it must be Chinese."
DialoGPT-SideControl: "You know your way, I can't be sure."

Table 19: Semantic label control generation results under 100 training samples sampled from DailyDialog.

	Controllability	Text Quality			
METHOD	ACCURACY ↑	PERPLEXITY ↓	BLEU-1 ↑	BLEU-2 ↑	METEOR ↑
DialoGPT-FT	0.3980	17.74	11.47	0.99	0.0428
DialoGPT-FUDGE	0.4745	-	14.58	1.61	0.0424
DialoGPT-PPLM	0.4947	-	11.35	0.75	0.0401
DialoGPT-SideControl	0.5418	18.74	14.38	1.38	0.0472

Table 20: Semantic label control performances under 1000 training samples sampled from DailyDialog, where $\lambda = 10^3$ for $\mathcal{L}_{control}$ in DialoGPT-SideControl.

Dialogue Context:	**Dialogue Act**:
"Well , have you ever heard of Oulong tea and compressed tea ?" "Oh , yeah . Oulong tea is good for one's health . isn't it ?" "You surely know a lot about Chinese tea." "Sure , I like drinking tea at teahouses." "Oh , so do I."	"directive"
Ground Truth: "Why don't we go for one now ?"	
DialoGPT-Ori: "I'm sure."	
DialoGPT-FT: "But do you like Chinese tea better than American tea?"	
DialoGPT-FUDGE: "I'm not sure if you're being serious or not, but I'm pretty sure that's not what he's saying."	
DialoGPT-PPLM: "I think it's the Chinese version that's for me"	
DialoGPT-SideControl: "You are the second person to make my point!"	

Table 21: Semantic label control generation results under 1000 training samples sampled from DailyDialog.

	Controllability	Text Quality			
METHOD	ACCURACY ↑	PERPLEXITY ↓	BLEU-1 ↑	BLEU-2 ↑	METEOR ↑
DialoGPT-FT	0.4284	15.41	14.51	1.4 9	0.0517
DialoGPT-FUDGE	0.4726	-	14.63	1.62	0.0420
DialoGPT-PPLM	0.5597	-	11.03	0.62	0.0360
DialoGPT-SideControl	0.5443	15.38	14.17	1.55	0.0448

Table 22: Semantic label control performances under 5000 training samples sampled from DailyDialog, where $\lambda = 10^4$ for $\mathcal{L}_{control}$ in DialoGPT-SideControl.

Dialogue Context:	**Dialogue Act**:
"Well , have you ever heard of Oulong tea and compressed tea ?" "Oh , yeah . Oulong tea is good for one's health . isn't it ?" "You surely know a lot about Chinese tea." "Sure , I like drinking tea at teahouses." "Oh , so do I."	"directive"
Ground Truth: "Why don't we go for one now ?"	
DialoGPT-Ori: "I'm sure."	
DialoGPT-FT: "you like the Chinese traditional silk?"	
DialoGPT-FUDGE: "I'm not sure if you're being serious or not, but I'm pretty sure that's not what he's saying."	
DialoGPT-PPLM: "It's not tea if you mix tea and tea together"	
DialoGPT-SideControl: "That's right. We should go to a tea party."	

Table 23: Semantic label control generation results under 5000 training samples sampled from DailyDialog.

	Controllability	Text Quality			
METHOD	ACCURACY ↑	PERPLEXITY ↓	BLEU-1 ↑	BLEU-2 ↑	METEOR ↑
DialoGPT-FT	0.3702	14.59	8.18	0.98	0.0480
DialoGPT-FUDGE	0.4726	-	14.52	1.59	0.0421
DialoGPT-PPLM	0.5853	-	11.27	0.61	0.0385
DialoGPT-SideControl	0.5309	14.36	15.33	1.61	0.0478

Table 24: Semantic label control performances under 10000 training samples sampled from DailyDialog, where $\lambda = 10^4$ for $\mathcal{L}_{control}$ in DialoGPT-SideControl.

Dialogue Context:	**Dialogue Act**:
"Well , have you ever heard of Oulong tea and compressed tea ?" "Oh , yeah . Oulong tea is good for one's health . isn't it ?" "You surely know a lot about Chinese tea." "Sure , I like drinking tea at teahouses." "Oh , so do I."	"directive"

Ground Truth: "Why don't we go for one now ?"
DialoGPT-Ori: "I'm sure."
DialoGPT-FT: "I'm looking forward to tea together. Have you heard the news about that Chinese new food?"
DialoGPT-FUDGE: "I'm not sure if you're being serious or not, but I'm pretty sure that's not what he's saying."
DialoGPT-PPLM: "It's an excellent way to clean yourself."
DialoGPT-SideControl: "You should."

Table 25: Semantic label control generation results under 10000 training samples sampled from DailyDialog.

Dialogue Context:	**Dialogue Act**:
"Well , have you ever heard of Oulong tea and compressed tea ?" "Oh , yeah . Oulong tea is good for one's health . isn't it ?" "You surely know a lot about Chinese tea." "Sure , I like drinking tea at teahouses." "Oh , so do I."	"directive"

Ground Truth: "Why don't we go for one now ?"
DialoGPT-Ori: "I'm sure."
DialoGPT-FT: "well, what kind of tea do you like?"
DialoGPT-FUDGE: "I'm not sure if you're being serious or not, but I'm pretty sure that's not what he's saying."
DialoGPT-PPLM: "I know, but it's just tea."
DialoGPT-SideControl: "I'm not in the mood to go."

Table 26: Semantic label control generation results under full training samples from DailyDialog.