Learning to Predict Persona Information for Dialogue Personalization without Explicit Persona Description

Wangchunshu Zhou[*] [†] **Qifei Li**[*] **Chenle Li**
Beihang University, Beijing, China
zhouwangchunshu@buaa.edu.cn

Abstract

Personalizing dialogue agents is important for dialogue systems to generate more specific, consistent, and engaging responses. However, most current dialogue personalization approaches rely on explicit persona descriptions during inference, which severely restricts its application. In this paper, we propose a novel approach that learns to predict persona information based on the dialogue history to personalize the dialogue agent without relying on any explicit persona descriptions during inference. Experimental results on the PersonaChat dataset show that the proposed method can improve the consistency of generated responses when conditioning on the predicted profile of the dialogue agent (i.e. "self persona"), and improve the engagingness of the generated responses when conditioning on the predicted persona of the dialogue partner (i.e. "their persona"). We also find that a trained persona prediction model can be successfully transferred to other datasets and help generate more relevant responses.

1 Introduction

Recently, end-to-end dialogue response generation models (Sordoni et al., 2015; Serban et al., 2016; Bordes et al., 2017) based on recent advances of neural sequence-to-sequence learning models (Sutskever et al., 2014; Vaswani et al., 2017) have gained increasing popularity as they can generate fluent responses. However, as the dialogue agent is trained with datasets containing dialogues from many different speakers, it can not generate personalized responses for the current speaker, making the generated responses less relevant and engaging (Li et al., 2016b).

To address this problem, recent studies attempt to personalize dialogue systems by generating dialogue responses conditioning on given persona descriptions have been shown to help dialogue agents perform better (Zhang et al., 2018; Mazaré et al., 2018). However, a major drawback of the current dialogue agent personalization approaches is that they require explicit persona descriptions in both training and inference stages, which severely limits their application in real-world scenarios because detailed persona descriptions for current speakers are not available in most scenarios. Another problem is that current dialogue personalization approaches are not interpretable and the role of additional persona information is unclear.

In this paper, we propose a novel dialogue agent personalization approach that automatically infers the speaker's persona based on the dialogue history which implicitly contains persona information. Our model generates personalized dialogue responses based on the dialogue history and the inferred speaker persona, alleviating the necessity of the persona description during inference.

Specifically, we propose two different approaches to perform persona detection. The first approach learns a "persona approximator" which takes dialogue history as the input and is trained to approximate the output representation of a persona encoder that takes explicit persona description as the input. The second approach instead addresses the persona detection problem as a sequence-to-sequence learning problem and learns a "persona generator" which takes the dialogue history as the input and generates the persona description of the speaker. This approach provides a stronger supervision signal compared with the first approach and is more interpretable as the encoded persona information can be decoded to reconstruct the detected persona description.

Our proposed approach can be used to incorporate both "self-persona" which is the persona information of the dialogue agent, and "their-persona" which is the persona information of the dialogue partner. On one hand, generating dia-

[*] Equal contribution.
[†] Corresponding author

Findings of the Association for Computational Linguistics: ACL 2023, pages 2979–2991
July 9-14, 2023 ©2023 Association for Computational Linguistics

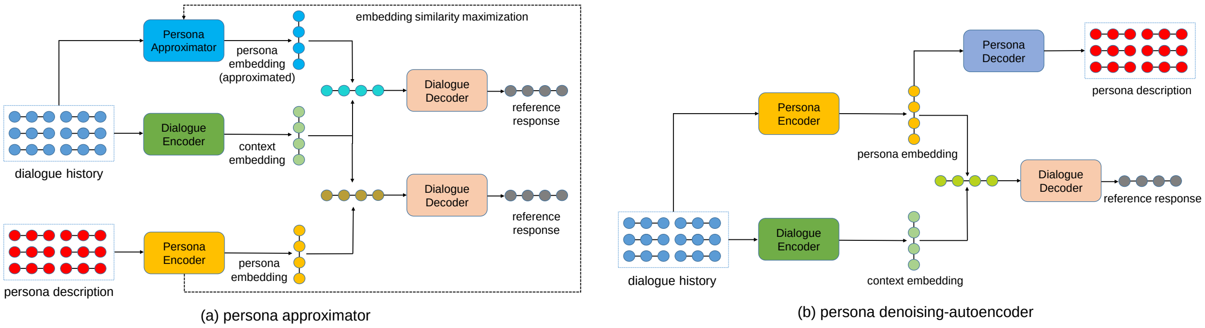

Figure 1: Illustration of the proposed persona detection models. The persona approximator is on the left. It is trained to maximize the embedding similarity between persona embedding approximated by the persona approximator and the persona encoder, which is obtained by taking dialogue history and persona description respectively. The persona generator is on the right, which is trained to recover persona description from the dialogue history, thus can also be viewed as a "persona denosing-autoencoder".

logue responses conditioning on the inferred "self-persona" can help the dialogue agent maintain a consistent persona during the conversation, thus enhancing the consistency of generated responses without the need of a pre-defined persona description for every dialogue agent. On the other hand, generating dialogue responses conditioning on the predicted persona of the dialogue partner helps the dialogue model generate more engaging responses that are relevant to its dialogue partner. The ability to automatically infer the persona information of the dialogue partner is particularly attractive because in many real-world application scenarios, the persona information of the user is hardly available before the dialogue starts. In addition, to facilitate training and tackle the problem of lacking training data, we propose to train the persona detection model with multi-task learning by sharing layers and training jointly with the dialogue context encoder in both approaches.

Our experiments on dialogue datasets with and without the persona description demonstrate the effectiveness of the proposed approach and show that a trained persona detection model can be successfully transferred to datasets without persona description.

2 Related Work

Preliminary study on dialogue personalization (Li et al., 2016b) attempts to use a persona-based neural conversation model to capture individual characteristics such as background information and speaking style. However, it requires the current speaker during inference to have sufficient dialogue utterances included in the training set, which is quite restricted by the cold-start problem.

More recently, Zhang et al. (2018) released the PersonaChat dataset which incorporates *persona* of two speakers represented as multiple sentences of profile description to personalize dialogue agents. They propose a profile memory network by considering the dialogue history as input and then performing attention over the persona to be combined with the dialogue history. Mazaré et al. (2018) proposed to train a persona encoder and combine the encoded persona embedding with context representation by concatenation. The combined representation is then fed into the dialogue decoder to generate personalized responses. (Yavuz et al., 2019) designed the DeepCopy model, which leverages copy mechanism to incorporate persona texts and Madotto et al. (2019) propose to use meta-learning to adapt to the current speaker quickly, their approach also requires several dialogues of the speaker to perform dialogue personalization, which is different from our approach. Welleck et al. (2019) propose a dialogue natural language inference dataset and use it to measure and improve the consistency of the dialogue system. More recently, Zheng et al. (2019) propose personalized dialogue generation with diversified traits. Song et al. (2020) introduce a multi-stage response generation stage to improve the personalization of generated responses. Wu et al. (2020) propose a variational response generator to better exploit persona information. Different from the aforementioned works, our approach does not require persona information during test time, which makes it more generally applicable.

Concurrently, Ma et al. (2021) propose to infer implict persona base on dialogue histories.

3 Methodology

The motivation behind the proposed approach is that we can learn to detect the profile (i.e., persona) of dialogue speakers based on the dialogue history, which is demonstrated by experimental results in Zhang et al. (2018) that we can train a model to effectively distinguish the corresponding persona from randomly sampled negative persona based on the dialogue history.

The key idea is to jointly train a persona detection model with a conventional dialogue response generation model. The persona detection model is trained with persona description to infer the persona information based on the dialogue history, which provides persona information for the dialogue model, thus alleviating the necessity of provided persona information during test time. We propose two different persona detection models. The first model is a "persona approximator" and the second is a "persona generator". An overview of the proposed models is illustrated in Figure 1. We describe them in detail in this section, together with a multi-task learning objective which facilitates the training stage of the model.

3.1 Task Definition

Given a dialogue dataset $\mathcal{D}$ with personas, an example of the dataset can be represented as a triplet (h, p, r). Specifically, $h = \{u_1, u_2, ..., u_{nh}\}$, which represents the dialogue history with nh utterances. $p = \{p_1, p_2, ..., p_{np}\}$, which represents a persona with np profile sentences. r represents the ground-truth response. Existing personalized dialogue models learn a dialogue response generation model G which takes h and p as input during inference and generates a personalized response $G(h, p)$. Our goal is to learn a persona detection model D which enables the dialogue model to generate personalized response $G(h, D(h))$ without relying on given persona description p during test time. In this way, the persona description in the dataset is used to train the personalized dialogue agent and after training, our model should be able to generate personalized dialogue responses without relying on persona description.

3.2 Persona Approximator

The idea of persona approximator is that given a trained personalized dialogue model with persona encoder which takes the persona description as input and outputs the persona embedding, we can train a persona approximator which takes the dialogue history as input and learns to output a persona embedding which is similar with that encoded by the trained persona encoder. Persona embedding approximation is possible as dialogue history is shown to be sufficient for discriminating the corresponding persona (Zhang et al., 2018).

Formally, given dialogue history h and persona description p, the persona encoder E takes p as input and outputs persona embedding $emb(p) = E(p)$. The proposed persona approximator A takes h as input and outputs the approximated persona embedding $a = A(h)$. The training objective of A is to optimize the embedding similarity (e.g. cosine similarity) between a and $emb(p)$. At the same time, we minimize the cosine similarity between a and the embedding of a randomly sampled persona embedding of another user, which serves as a negative example.

We discuss several pros and cons of the proposed persona approximator here. The advantage of this approach is that it alleviates the requirement of persona description during training and can incorporate several off-the-shelf personalized dialogue models with persona encoder seamlessly. However, as the persona encoder itself is far from perfect and non-interpretable, a persona approximator which is trained to approximate the persona encoder may also be sub-optimal and even less interpretable. Another issue is that the persona approximator can only be trained after training the dialogue model and persona encoder. To alleviate this problem and train an interpretable persona detection model more effectively, we propose another persona detection model which is named "persona generator".

3.3 Persona Generator

As dialogue history can be used to predict the corresponding persona, which is demonstrated by Zhang et al. (2018), we hypothesize that dialogue history implicitly contains the persona of dialogue partners. Therefore, we argue that a good persona detection model should be able to reconstruct the dialogue partners' persona descriptions based on the dialogue history. Based on this insight, we propose a "persona generator" model which formulates the persona detection problem as a sequence-to-sequence learning problem and train the persona generator to recover the textual persona description of dialogue partners from the dialogue history.

Formally, the persona generator receives the dialogue history h as input and is trained to generate the persona description p, which is a sequence of tokens p_i of length n. The persona generator is trained by maximizing the likelihood of the ground-truth persona descriptions:

$$\mathrm{L}_{pg} = -\sum_{i=1}^{n} \log P(p_i | p_{<i}, h) \tag{1}$$

As illustrated in Figure 1(b), the persona generator consists of a persona encoder and a persona decoder. During training, the persona encoder takes the dialogue history as input and outputs a persona embedding that represents the persona information of either the dialogue model or its dialogue partner. The persona embedding is then concatenated with the context embedding generated by the dialogue encoder and fed into the dialogue decoder to generate the response. In addition, the persona embedding is also fed into the persona decoder to generate the textual persona description of the dialogue partner. During inference, only the encoder of the trained persona generator will be used to provide persona information for the response generation model.

While previous dialogue personalization approaches, as well as the aforementioned persona approximator, generally train the persona encoder to maximize the likelihood of gold responses with MLE and can not ensure that the persona encoder actually captures useful persona information, the persona generator is directly trained to generate persona information from dialogue history, which enforces the persona information to be successfully captured. This approach also enhances the interpretability of the dialogue personalization procedure as the persona embedding encoded from dialogue history can be decoded into persona description with the decoder of trained persona generator.

3.4 Multi-Task Learning

Training the proposed persona detection models can be difficult because the available persona description is limited. To alleviate this problem, we propose to adopt multi-task learning (Argyriou et al., 2006) by training the dialogue encoder jointly with the persona detection model. This is possible because both the dialogue encoder and the persona detection model take dialogue history as input and outputs a latent vector. The difference is that the dialogue context encoder is trained to provide direct information for response generation while the persona detection model is trained to predict persona description. These two tasks both require dialogue understanding and commonsense reasoning ability, which can be shared and help each other generalize better. We thus propose to adopt the multi-task learning paradigm to facilitate training. Specifically, we share the parameter of the first layer, which can be viewed as a general-purpose dialogue information encoder, between the dialogue context encoder and the persona detection model.

In addition, we also train the persona detection model to maximize the likelihood of ground-truth responses together with the dialogue model, which ensures that the persona detection model not only encodes persona information but also helps generate more fluent dialogue responses. We control the relative importance between the original MLE objective and the training objectives of the proposed persona detection models by weighting the loss of persona detection objective with a hyperparameter α which is empirically set to 0.1 in our experiments.

4 Experiments

4.1 Dataset

We conduct our experiments on PersonaChat dataset (Zhang et al., 2018) which is a multi-turn chit-chat conversation dataset containing conversations between human annotators who are randomly assigned a "persona". We experiment with two settings where the models are trained either with the persona description of themselves (i.e., self persona) or with the persona description of their dialogue partner (i.e., their persona). We present an example of the dataset in the Appendix.

In addition, we also expect our approach to be able to perform personalized dialogue response generation on other datasets (application scenarios) where persona description is not available even in the training set. Therefore, we also conduct experiments on the Dailydialog dataset (Li et al., 2017), which is a multi-turn dialogue dataset in a similar domain with PersonaChat but without persona description, to explore the transferability of our approach.

4.2 Evaluation Metrics

For automated evaluation, we employ the following metrics following previous work:

- **Perplexity** Following Zhang et al. (2018), we use perplexity (ppl) to measure the fluency of responses. Lower perplexity means better fluency.
- **Distinct** Following (Li et al., 2016a), we calculate the token ratios of distinct bigrams (Distinct-2, abbreviated as Dst for convenience). We use this metric to measure the diversity of the responses.
- **Hits@1** Following Zhang et al. (2018), Hit@1 measures the percentage of correct identification of a gold answer from a set of 19 distractors.
- **Consistency** We also include the Consistency score proposed by Welleck et al. (2019). It is calculated by subtracting the percentage of generated response entails or contradicts (predicted with a pretrained dialogue NLI model) the persona information.
- **P-Cover** We also include the P-Cover metric proposed by Song et al. (2019), which evaluates how well the generated responses covers the persona information.

As automated metrics generally fail to correlates well with human evaluation (Liu et al., 2016; Zhou and Xu, 2020). We also systematically conduct human evaluation to further evaluate the proposed method. Specifically, we invite 20 human annotators that are all graduate students with good English proficiency to evaluate the quality of the model. Following Zhang et al. (2018), we ask human annotators to interact with compared models and evaluate the fluency, engagingness, and consistency of the model (scored between 1- 5). In addition, the degree of personalization of the model is measured by the ability of human annotators to detect the model's profile after the conversation, which is measured by displaying the real persona description together with a randomly sampled persona description and asking the human annotator to select which is more likely to be the profile of the model. The persona detection metric is only available in PersonaChat where test persona is available.

4.3 Compared Models

To explore to what extent our proposed approach is able to personalize dialogue agents, we compare two variants of our model which incorporate the persona approximator method and the persona generator method with the following baseline models:

- **DialogGPT** A Transformer-based dialogue response generation based on the GPT-2 architecture and pre-trained on 147M conversation-like exchanges extracted from Reddit comment chains. It has 345M parameters and fine-tuned on Personachat by prepending all persona descriptions at the begining of the dialogue context.
- **DialogGPT w/o persona** The same DialogGPT model fine-tuned on Personachat dataset without using persona information during training or inference.
- **DialogGPT+PE** A transformer-based dialogue model based on pre-trained DialogGPT model and fine-tuned by training a transformer-based persona encoder to provide persona embedding information.
- **PersonaCVAE** Our re-implementation of the PersonaCVAE model (Song et al., 2019) with the pre-trained DialogGPT as the base model.
- **GPMN** Generative Profile Memory Network (Zhang et al., 2018) is an RNN-based model that encodes persona as memory representations in a memory network.

Both of our models (Persona Approximator and Persona Generator) are based on pre-trained DialogGPT (Zhang et al., 2020) and fine-tuned on Personachat. The model has the same architecture with GPT-2 and has 345M parameters. Fine-tuning hyperparameters are kept the same with Zhang et al. (2020). To make the model compatible with the encoder-decoder architecture described in the method section, we consider the hidden state of the last token in the transformer model as the context embedding. For the persona encoder, we share all layers except the last layer in the multi-task setting. The RNN-based baselines are trained from scratch and we used their original

Method	Self Persona					Their Persona				
	ppl	Dst	Hits@1	Cons	P-Cover	ppl	Dst	Hits@1	Cons	P-Cover
GPMN	36.11	13.5	54.9	0.15	.018	36.45	14.8	51.4	0.10	.021
DialogGPT	13.62	23.1	83.2	0.35	.052	14.03	23.9	78.9	0.27	.059
DialogGPT+PE	13.57	24.8	**84.5**	**0.38**	.055	13.90	25.1	**79.3**	0.28	**.063**
PersonaCVAE	14.83	**25.7***	84.0	0.37	.061	14.88	25.6	78.3	0.24	.066
DialogGPT w/o persona	15.49	19.6	72.9	0.13	.012	-	-	-	-	
Persona Approximator	14.42	24.2	83.3	0.33	.038	14.63	24.9	78.4	0.24	.040
Persona Generator	**13.39***	25.2	84.2	**0.38**	.049	**13.82**	**25.8**	79.1	**0.29**	.057

Table 1: Performance of dialogue models on automated evaluation metrics in the PersonaChat testset. "Self persona" means that the model is conditioned on the persona description of itself while "their persona" means the model is conditioned on the persona of its dialogue partner. We report the median as 5 random runs as the result. * denote statistically significant with p-value < 0.05.

architecture and training methods in the original paper.

4.4 Experimental Results

Results on PersonaChat We first present the experimental results on the PersonaChat dataset where persona description is available during training. In this scenario, the persona detection model is trained in the same domain as the response generation model.

The results of automated evaluation metrics are shown in Table 1. First, we can see that models explicitly incorporate textual persona descriptions, including the dialogue model that incorporate a persona encoder (i.e., **DialogGPT+PE**) or prepend persona descriptions (i.e., **DialogGPT**), outperform the baseline model that does not exploit persona information by a relatively large margin in all automated metrics. Also, dialogue models with a pre-trained Transformer model (i.e., DialogGPT) substantially outperform RNN-based models.

As for our proposed approaches, we find that both persona detection models substantially improve the performance upon the baseline with the pre-trained DialogGPT model without using persona information. When comparing the proposed two persona detection models, it is clear that the persona generator method performs much better than the persona approximator. Moreover, we find that it outperforms the competitive **DialogGPT** and **DialogGPT+PE** model on several automated metrics despite not using any persona information at test time. We hypothesis that it is because the persona generator is trained with the reconstruction loss, which is a useful supervision signal that is complementary to the MLE objective. In contrast, the persona encoder is trained jointly with the dialogue model by simply maximizing the likelihood of gold responses and may not actually capture the persona information. Our approach performs slightly worse than the best model using persona information in some metrics. However, the difference is very marginal even though our model does not take the persona information as input.

When comparing the performance of our proposed approaches trained with either "self persona" and "their persona", we can see that training the persona detection to predict the persona information of the dialogue system itself helps the model to maintain a consistent persona, thus improving the consistency of generated responses. In contrast, training the persona detection model to predict the persona of its dialogue partner helps the model to generate more diverse responses.

Human evaluation results are shown in Table 2. We can see that dialogue models which explicitly incorporate textual persona descriptions significantly improves all human evaluation metrics.

As for our proposed approaches, we find that both proposed persona detection models can improve the consistency, engagingness, and persona detection accuracy upon the baseline seq2seq model without sacrificing the fluency of generated responses. The persona generator performs better than the persona approximator, which is consistent with the results in the automated evaluation. In addition, the persona generator model performs comparably and even better when compared with the competitive **DialogGPT** baseline. This demonstrates that our proposed method can effectively personalize dialogue agents without relying on pre-defined persona descriptions at test time.

Similarly, we find that when conditioning on "self persona" as incorporating the persona description helps dialogue agents maintain a consis-

Model	Persona	Fluency	Engagingness	Consistency	Persona Detection
DialogGPT	self	3.56	3.57	3.63	0.88
DialogGPT	their	3.49	3.59	3.47	0.80
DialogGPT	both	3.63	3.69	3.60	0.88
DialogGPT+PE	self	3.62	3.49	3.61	0.87
DialogGPT+PE	their	3.57	3.51	3.52	0.82
DialogGPT+PE	both	3.69	3.65	**3.68***	**0.90**
PersonaCVAE	self	3.51	3.55	3.53	0.85
PersonaCVAE	their	3.50	3.52	3.42	0.77
PersonaCVAE	both	3.57	3.59	3.51	0.83
DialogGPT w/o persona	–	3.39	3.28	3.30	0.69
Persona Approximator	self	3.45	3.40	3.35	0.78
Persona Approximator	their	3.36	3.43	3.27	0.73
Persona Generator	self	3.67	3.61	3.58	0.89
Persona Generator	their	3.61	3.69	3.52	0.84
Persona Generator	both	**3.72***	**3.74***	3.63	**0.90**

Table 2: Human evaluation of dialogue models with different personalization approaches on the PersonaChat dataset. * denote statistically significant with p-value < 0.05. The Fleiss's Kappa value is 0.67, which indicates relatively strong inter-annotator agreement.

Model	Per	Fluen	Engag	Consis
DialogGPT w/o persona	–	3.42	3.41	3.48
w/ Persona Generator	self	**3.53**	3.52	**3.58**
w/ Persona Generator	their	3.48	**3.57**	3.56

Table 3: Performance of dialogue models with different personalization approaches on the Dailydialog dataset. The Fleiss's Kappa value is 0.61, indicating relatively strong inter-annotator agreement.

tent profile throughout the conversation. Again, when conditioned on "their persona", the dialogue agent learns to predict the profile of its dialogue partner, which helps generate more engaging and personalized responses. Based on this motivation, we also conduct experiment with both "their" and "self" persona at the same time. We find this make significant future improvement and enabling dialogue agent to generate dialogue responses that are both engaging and consistent.

On the transferability of persona detection models As persona descriptions are not available in most scenarios and datasets, we aim to enable dialogue agent personalization for dialogue models trained in datasets where no persona description is available with a persona detection model pretrained on PersonaChat. To test the transferability of trained persona detection models, we combine persona detection models pretrained on the PersonaChat dataset with dialogue systems trained on the Dailydialog dataset. The pretrained persona detection models are fine-tuned jointly with the pretrained dialogue model by maximizing the likelihood of ground-truth responses. The results are shown in Table 3. We can see that transferring pre-trained persona detection models in the target dialogue domain is able to improve the performance of dialogue models. Specifically, predicting self-persona improves the consistency of the dialogue agent while detecting the persona of the dialogue partner improves the engagingness of generated responses. The experimental result also confirms the effectiveness of the proposed persona generator model and the persona reconstruction loss.

4.5 Ablation Study

To further understand the proposed models, we conduct an ablation study that focuses on: 1) the effectiveness of the multi-task learning architecture and the multi-task objective of persona detection models, and 2) the effect of available dialogue history length on the performance of persona detection models. We employ the dialogue response generation model with persona generator with self persona as the full model and compare it with the following ablated variants: (1) **first half:** The variant where only the first half of conversations are used as the test set, which makes the input dialogue history for persona generator shorter. (2) **second half:** The counterpart of **first half** where the available dialogue histories for persona generator are longer. (3) **w/o shared layers:** The vari-

Model	perplexity	Dst	Hits@1	Cons
DialogGPT w/o Persona	15.49	19.6	72.9	0.13
- first half	18.31	15.7	66.5	0.05
- second half	13.24	23.8	79.3	0.19
w/ Persona Generator	13.39	25.2	84.2	0.38
- first half	16.24	23.5	79.8	0.32
- second half	12.01	26.9	88.6	0.44
- w/o shared layers	13.92	24.9	83.5	0.35
- w/o joint training	14.05	24.7	83.8	0.36

Table 4: Results of the ablation study

ant where the persona generator does not share its first layer with the encoder of the dialogue model. (4) **w/o joint training:** The variant where the persona generator is exclusively trained with the reconstruction loss without jointly training with the MLE objective.

The results of the ablation study are shown in Table 4. We can see that both sharing layers and joint training improve the performance of the persona detection model, which demonstrates the effectiveness of multi-task learning in our task. As for the influence of the length of the dialogue history, we find that the proposed persona generator model performs better when giving longer dialogue history (i.e., the second half of the conversation), which is demonstrated by a larger relative improvement compared with the sequence-to-sequence baseline given the same dialogue history. This is reasonable as longer dialogue history may provide richer information and help detect persona better. It also suggests that our approaches may be more effective for dialogue agents that aim to conduct relatively long dialogues with humans. This problem is similar to the well-known cold-start problem in the field of recommend systems. However, this does not suggest that our proposed approach is not useful for most application scenarios where the dialogue agent must start the dialogue from scratch. In contrast, our model will continually track the persona information of both the dialogue agent itself and the dialogue partner, thus maintaining a consistent persona throughout the progress of the dialogue and gradually improve the engagingness of generated responses with the dialogue going on. In addition, the ability to automatically infer the persona information of the dialogue partner is also beneficial for real-world applications, where although we can pre-define a persona for the dialogue agent, the users' persona is not always available.

No persona	I don't know what you could not do ?
PE w/ self	I am going to the club now.
PE w/ their	Do you want to play frisbee or something?
PG w/ self	okay I am going to make a cake.
- Generated Persona:	... I craving eating cake...
PG w/ their	I prefer that let's watch tv together.
- Generated Persona:	... I like TV show...

Table 5: Case study of the continuation of the conversation shown in Table 1 in the Appendix.

4.6 Qualitative Analysis

To better understand the proposed method intuitively, we conduct a case study by feeding different variants of the dialogue model with the dialogue history presented in the Appendix and generate different continuations of the conversation. The next utterances generated by different model variants are shown in Table 5. We can see that the dialogue model without persona information generates an irrelevant response that is not engaging. In contrast, both the persona encoder which takes the predefined persona description and the persona generator which infers the persona from dialogue history enables the dialogue agent to generate consistent and relevant responses, which are likely to be more engaging for the dialogue partner. In addition, we present the outputs of the decoder in the persona generator, which demonstrates that the proposed approach is more interpretable.

5 Conclusion

In this paper, we propose a dialogue personalization approach that automatically infers the current speakers' persona based on the dialogue history, which enables neural dialogue systems to generate personalized dialogue responses without using persona description at test time. Our experiments on the PersonaChat dataset show that the proposed models can improve the model's consistency and engagingness when conditioning on the inferred persona information of the dialogue agent itself or the dialogue partner. We also conduct experiments on the Dailydialog dataset where persona description is not available and find that pre-trained persona detection models can be successfully transferred to other datasets without annotated persona descriptions. This confirms the potential of our approach for dialogue personalization in domains where persona descriptions are not available or expensive to collect. Nevertheless, our method still requires annotated persona information during training, which can be hard to get for specific domains. We leave this for future work.

Limitations

One limitation of this work is that while our approach alleviates the requirement of persona description during inference, it still requires persona description for the training corpus. A viable solution is to transfer the pre-trained persona detection models to other datasets without persona description in train set. However, the success of this approach may depend on the degree of similarity between the target dataset and the PersonaChat dataset.

Ethics Considerations

Our proposed method can generate personalized dialogue responses to users and improve the engagingness of the dialogue systems. It faces several common ethics concerns that a neural dialogue system may generate unexpected responses that make human users uncomfortable. However, it is common for most neural dialogue systems. Another potential risk is that the persona generator may generate unexpected persona information that makes user uncomfortable. This issue could be addressed by adding constraints on the generated persona information.

References

Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. 2006. Multi-task feature learning. In *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006*, pages 41–48. MIT Press.

Antoine Bordes, Y-Lan Boureau, and Jason Weston. 2017. Learning end-to-end goal-oriented dialog. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

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

Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and Bill Dolan. 2016b. A persona-based neural conversation model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 994–1003, Berlin, Germany. Association for Computational Linguistics.

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.

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Mike Noseworthy, Laurent Charlin, and Joelle Pineau. 2016. How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2122–2132, Austin, Texas. Association for Computational Linguistics.

Zhengyi Ma, Zhicheng Dou, Yutao Zhu, Hanxun Zhong, and Ji-Rong Wen. 2021. One chatbot per person: Creating personalized chatbots based on implicit user profiles. In *SIGIR*, pages 555–564. ACM.

Andrea Madotto, Zhaojiang Lin, Chien-Sheng Wu, and Pascale Fung. 2019. Personalizing dialogue agents via meta-learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5454–5459, Florence, Italy. Association for Computational Linguistics.

Pierre-Emmanuel Mazaré, Samuel Humeau, Martin Raison, and Antoine Bordes. 2018. Training millions of personalized dialogue agents. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2775–2779, Brussels, Belgium. Association for Computational Linguistics.

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 *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 3776–3784. AAAI Press.

Haoyu Song, Yan Wang, Wei-Nan Zhang, Xiaojiang Liu, and Ting Liu. 2020. Generate, delete and rewrite: A three-stage framework for improving persona consistency of dialogue generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5821–5831, Online. Association for Computational Linguistics.

Haoyu Song, Weinan Zhang, Yiming Cui, Dong Wang, and Ting Liu. 2019. Exploiting persona information for diverse generation of conversational responses. In *IJCAI*, pages 5190–5196. ijcai.org.

Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. A neural network approach to context-sensitive generation of conversational responses. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 196–205, Denver, Colorado. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Sean Welleck, Jason Weston, Arthur Szlam, and Kyunghyun Cho. 2019. Dialogue natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3731–3741, Florence, Italy. Association for Computational Linguistics.

Bowen Wu, MengYuan Li, Zongsheng Wang, Yifu Chen, Derek F. Wong, Qihang Feng, Junhong Huang, and Baoxun Wang. 2020. Guiding variational response generator to exploit persona. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 53–65, Online. Association for Computational Linguistics.

Semih Yavuz, Abhinav Rastogi, Guan-Lin Chao, and Dilek Hakkani-Tur. 2019. DeepCopy: Grounded response generation with hierarchical pointer networks. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 122–132, Stockholm, Sweden. Association for Computational Linguistics.

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? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213, Melbourne, Australia. 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 *ACL, system demonstration*.

Yinhe Zheng, Guanyi Chen, Minlie Huang, Song Liu, and Xuan Zhu. 2019. Personalized dialogue generation with diversified traits. *arXiv preprint arXiv:1901.09672*.

Wangchunshu Zhou and Ke Xu. 2020. Learning to compare for better training and evaluation of open domain natural language generation models. In *AAAI*, pages 9717–9724.

Limitations

One limitation of this work is that while our approach alleviates the requirement of persona description during inference, it still requires persona description for the training corpus. A viable solution is to transfer the pre-trained persona detection models to other datasets without persona description in train set. However, the success of this approach may depend on the degree of similarity between the target dataset and the PersonaChat dataset. Our transfer experiments on the DialyDialog dataset and the additional Reddit dataset confirms the effectiveness of transferring a pre-trained persona detection model.

Another limitation of this work is that adding the persona detection module will increases the model size and slow down the inference. The size issue can be reduced by sharing parameters between the persona detection module and the dialogue model. The inference speed issue only results in approximately $1.03\times$ inference latency compared to the original model because the majority inference time is on decoding which is less affected by the persona detection module.

ACL 2023 Responsible NLP Checklist

A For every submission:

□ A1. Did you describe the limitations of your work?
Left blank.

□ A2. Did you discuss any potential risks of your work?
Left blank.

□ A3. Do the abstract and introduction summarize the paper's main claims?
Left blank.

□ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B □ Did you use or create scientific artifacts?

Left blank.

□ B1. Did you cite the creators of artifacts you used?
Left blank.

□ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Left blank.

□ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Left blank.

□ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Left blank.

□ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Left blank.

□ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Left blank.

C □ Did you run computational experiments?

Left blank.

□ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Left blank.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

□ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Left blank.

□ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Left blank.

□ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Left blank.

D □ Did you use human annotators (e.g., crowdworkers) or research with human participants?
Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Left blank.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Left blank.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Left blank.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Left blank.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Left blank.