Improving Personalized Dialogue Generation Models with Data-level Distillation and Diversification

Anonymous ACL submission

Abstract

Personalized dialogue generation is a challenging task in which a persona-consistent response needs to be generated conditioning both persona texts and dialogue utterances, being more complex than conventional dialogues. Multiple persona texts and utterances exist in one sample and some of them can be distractors for generating. Thus even strong models have difficulty posing attention to suitable personas so generating persona-irrelevant responses. Besides, the limited data scale and diversity further affect the performance. Thus, we start from data and propose to boost the model in data-level distillation and diversification ($\mathbf{D}^3$). We first distill the original training samples into simplified persona-consistent ones, lowering the difficulty by removing redundant information in personas and dialogue history. Next in the diversification, we increase both the amount and diversity of distilled data to ease its insufficiency. A model will be trained via curricula, first on easier augmented samples and then on harder original ones. Experiments on the PersonaChat benchmark dataset illustrate the superiority of our method when packed with two strong base dialogue models (Transformer and GPT2) on various automatic metrics and human evaluation.

1 Introduction

Deep neural dialogue models have shown to be effective when trained on large-scale data, such as Seq2Seq (Bahdanau et al., 2015), CVAE (Zhao et al., 2017) and Transformers (Vaswani et al., 2017). Pretrained language models, like OpenAI GPT (Radford et al., 2018) and GPT2 (Radford et al., 2019), also prove their capabilities on dialogue generation tasks (Budzianowski and Vulić, 2019; Ham et al., 2020). Recently, there is a growing interest in personalized dialogue generation (Song et al., 2019; Wolf et al., 2019). Figure 1 shows a clipped personalized dialogue from PersonaChat (Zhang et al., 2018a). An interlocutor is explicitly described using several persona texts, which makes it harder to generate desired responses as additional personalities need to be conditioned.

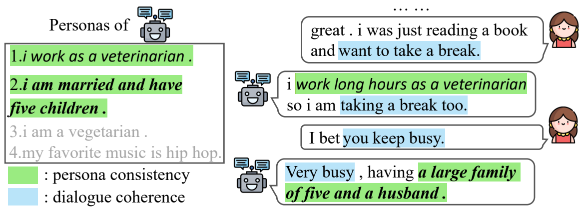


Figure 1: Responses in personalized dialogue generation are mostly correlated to one persona text and the latest utterance. (Grey persona texts are redundant and font styles indicate different persona consistency.)

The challenge of personalized dialogue generation also originates from the training data, where multiple personality sentences and history utterances are included in each sample. Even the powerful Transformer model encounters difficulty in learning on current data when concatenating all these texts as a single long sequence input. The uncertain consistency relationship between the response and each persona text, as well as the large input length, makes it hard for a model to pose enough attention to proper texts. This motivates us to build a better format for current training data to ease the model training. We can remove redundant information in both personal texts and dialogue history such that a response is highly correlated to the selected persona texts and history utterances.

Another reason for the unsatisfactory performance lies in the limited size and diversity of training data. Compared with conventional dialogue datasets, such as OpenSubtitles (Lison and Tiedemann, 2016) and Weibo (Gao et al., 2019) who reach millions of data scale, personalized dialogue datasets are small and dwarfed. E.g., PersonaChat SELF dataset only has 65.7k samples with 4.7k unique persona texts. Since it is costly to collect high-quality personalized dialogues, it is meaningful to seek techniques to make current data more diverse. Former dialogue data augmentation studies show remarkable promotion (Li et al., 2019; Cai et al., 2020a), but they only consider the relation between a query and a response to get new dialogue pairs. To augment personalized dialogues, we need to maintain both the coherence between the history and response and the consistency between the persona text and response simultaneously, which can not be accomplished by former methods.

Inspired by the above discussions, we propose **D**ata-level **D**istillation and **D**iversification (**D**^{3}), a data augmentation method to promote personalized dialogue generation without model modification. Original training samples are firstly distilled to contain only useful and less redundant persona texts and dialogue utterances for more efficient learning. Due to the easiness of distilled samples, we augment samples by imitating them instead of the difficult original ones. We design various methods to obtain edited new personas, and then new aligned consistent responses to promote the data diversity. With both augmented distilled and original data in hand, we arrange them into a data curriculum for model learning (Bengio et al., 2009), where the model is trained on the easier augmented distilled data and then the harder original data. To examine our method, we perform experiments on the PersonaChat benchmark dataset (Zhang et al., 2018a) with our method used on two popular models, Transformer encoder-decoder and GPT2. It is also easy to be extended to other models.

Our contributions can be summarized as follows:

- We distill original training data to get simplified persona-consistent samples as an easy data curriculum, helping the model training more effectively.
- We further diversify the distilled data via editing new personas and constructing corresponding aligned responses with quality filtering.
- Extensive experiments and analysis are conducted to demonstrate how **D**^{3} affects the model.

2 Related Work

Personalized dialogue generation It sees growing interest in recent years, thanks to the release of benchmark datasets such as PersonaChat/ ConvAI2 (Zhang et al., 2018a; Dinan et al., 2020). Previous works mostly focus on modifying dialogue models to condition the auxiliary personality information, including extra persona embedding(Li et al., 2016b), profile memory (Zhang et al., 2018a), copying from personas (Yavuz et al., 2019), CVAE with persona texts (Song et al., 2019), and so on (Song et al., 2020). Recent works try to adopt the more powerful Transformer (Vaswani et al., 2017) or large-scale pre-trained models on this task. Most of them can achieve a fairly good generation by simply concatenating persona texts and dialogue history together as a single input (Wolf et al., 2019; Roller et al., 2020). However, state-of-the-art results are still far from satisfactory.

Text data augmentation It has been widely used in many NLP tasks (Sennrich et al., 2016; Hou et al., 2018; Guo et al., 2019; Min et al., 2020). It is also effective to boost the performance of dialogue models. New generated dialogue utterances (Li et al., 2019; Niu and Bansal, 2019) and retrieval results (Zhang et al., 2020) can be used to augment the training data. However, all previous work only study the pairwise relationship between a query and a response to design the augmentation techniques, that are not applicable to involving the auxiliary information such as personas simultaneously.

Besides data augmentation, there are other ways to manipulate dialogue data to improve model learning. For example, a few approaches filter uninformative or noisy samples to enhance data quality (Csáky et al., 2019; Akama et al., 2020). Cai et al. (2020a) combine data augmentation and reweighting to make models learn more effectively.

Curriculum learning Bengio et al. (2009) examine the benefits of training models using various curricula successively from easy to hard. It has been applied to many NLP tasks such as machine translation (Platanios et al., 2019), reading comprehension (Tay et al., 2019) and language understanding (Xu et al., 2020). Cai et al. (2020b) adopt the idea in open-domain dialogue generation, where curriculum plausibility is determined by the response properties including coherence and diversity. Our work is different in that we introduce new data regarding personas as a curriculum.

3 Methodology

We first formally define the personalized dialogue generation task. Each sample consists of multiple L persona description texts $P = \{p_1, p_2, .., p_L\}$, M dialogue history utterances $H = \{h_1, h_2, .., h_M\}$, and a gold response $R = \{r\}$. The original training dataset is $\mathcal{D} = \{(P, H, R)\}$. For an input containing P and H from $\mathcal{D}$, a dialogue model needs to generate a response $\hat{r}$, which is coherent with the dialogue history H as well as reflecting part of the persona P. Taking PersonaChat (Zhang

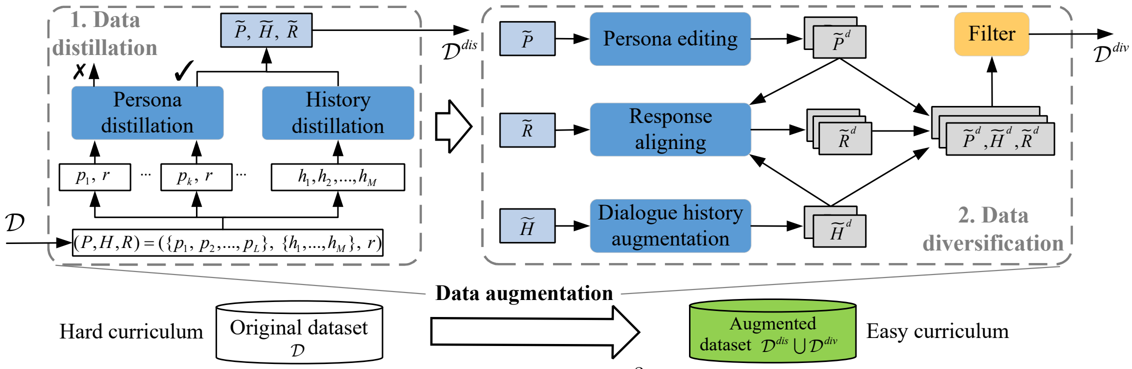


Figure 2: The framework of our data augmentation method **D**^{3}. It aims to obtain the augmented dataset $\mathcal{D}^a = \mathcal{D}^{dis} \cup \mathcal{D}^{div}$ from the original dataset $\mathcal{D}$. Curriculum strategy is used, where a model first learns on the augmented data $\mathcal{D}^a$ as an easy curriculum and then on the original training data $\mathcal{D}$ as the hard curriculum.

et al., 2018a) as an example, L ranges from 4 to 6, persona texts are simple statements, e.g., "I favorite music is country music" or "I work in sales".

Given a dialogue model architecture, we aim to promote it by augmenting the original training dataset $\mathcal{D}$ to $\mathcal{D}^a$ in three steps, shown in Figure 2:
1. Data distillation. We construct simple persona-consistent data $\mathcal{D}^{dis} = \{(\widetilde{P}, \widetilde{H}, \widetilde{R})\}$ by removing redundant information in P and H;
2. Data diversification. Due to the limited amount of distilled samples, we desire various methods to increase the variety and scale of them and obtain the diversified data $\mathcal{D}^{div} = \{(\widetilde{P}^d, \widetilde{H}^d, \widetilde{R}^d)\}$;
3. Data curriculum. We use $\mathcal{D}^{dis}$ and $\mathcal{D}^{div}$ to compose the augmented dataset $\mathcal{D}^a$, extending $\mathcal{D}$. And a curriculum strategy is defined to train the model.

3.1 Data distillation

Before introducing our distillation method, we discuss the difficulty of training a model with the original training samples in detail. The dependency of a response on the given persona fluctuates between different parts of the persona texts. As shown in Figure 1, most responses only correspond to one persona text. The remaining long persona parts are mostly **redundant**, which are noises to confuse the model to learn suitable attention on personas. Similarly, the long dialogue history information is mostly **redundant** except the latest utterance, which may further deteriorate this model training.

Therefore, we distill an original sample into a new one such that all responses are consistently determined on the provided persona texts and dialogue history. This is also connected with previous work, in which models benefit from data with supervised attention (Liu et al., 2016; Hsu et al., 2018). Here, we also mimic to output "hard" attention alignment between the response and useful persona texts/dialogue history by simply removing the unaligned information. Unlike previous work that inject supervision by modifying the model, our method only manipulates data. In the following, we describe how to construct simplified samples from the perspective of reducing redundancy in persona texts and dialogue history respectively.

Distill persona texts We aim to determine which persona texts consistent with the current response. Given a sample (P, H, R), we associate each persona p_k with the target response r to form a series sentence pairs $\{(p_1, r), (p_2, r), ..., (p_L, r)\}$. Hence, we formulate the problem as determining the consistency between p_k and r. We cast it as a natural language inference (NLI) problem, in which a model needs to determine whether a sentence r entails the other sentence p_k. If r entails p_k with a probability $\rho_e \geq \tau$, where τ is a threshold, it is considered to be consistent with P_k, otherwise irrelevant to P_k. A RoBERTa (Liu et al., 2019) model is used here as the NLI model with an accuracy of 90.8% on the DialogueNLI (Welleck et al., 2019) dev set (details in Appendix A.1).

Distill dialogue history We notice that models tend to attend more on the rear utterances of $\mathcal{H}$ rather than the front ones (Appendix C.1). Similar observations were also drawn in previous work (Khandelwal et al., 2018; Sankar et al., 2019). Thus we only keep the latest utterance H_M in a new sample, which should ease the model learning while also guaranteeing the generation coherence.

A distilled sample $(\widetilde{P}, \widetilde{H}, \widetilde{R})$ is finally obtained. Here, $\widetilde{P} = \{p_k\}$ where p_k entails r, $\widetilde{H} = \{h_M\}$ which is the last utterance in the dialogue history, and $\widetilde{\mathcal{R}} = \{r\}$. Such samples form the distilled set $\mathcal{D}^{dis}$. Note that an original sample in $\mathcal{D}$ may result in none, one, or multiple distilled samples, as R may entail none, one, or multiple persona texts.

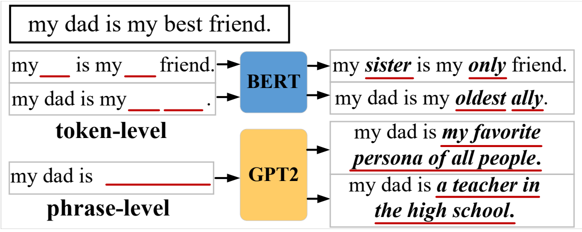


Figure 3: The illustration of persona editing.

3.2 Data diversification

Distilled samples should ease the model training as their responses strongly condition $\widetilde{P}$ and $\widetilde{H}$. However, samples obtained in $\mathcal{D}^{dis}$ are limited in terms of both **scale** (40% quantity of the original data) and **diversity** (about 4.5k unique persona sentences), which may affect the training efficiency of data-driven models so that personalized dialogue generation. Hence, it is necessary to diversify $\mathcal{D}^{dis}$.

Some studies validate the advantage of adding augmented samples on conventional dialogue tasks (Li et al., 2019; Cai et al., 2020a). But these methods only consider the query-response dialogue pairs and cannot handle the more complicated dependency between dialogue and the auxiliary information such as personas. Due to the higher persona-response certainty and less distraction of $\mathcal{D}^{dis}$, it is easier for us to diversify it with more semantically various samples especially in terms of persona texts to benefit models. Our data diversification containing three main parts as shown in Figure 2: persona editing, dialogue history augmentation, and response aligning along with quality filtering, starting from a distilled sample $(\widetilde{P}, \widetilde{H}, \widetilde{R})$.

Persona editing It is essential to involve more diverse persona texts in order to learn a robust persona-consistent model. We consider both token-level and phrase-level editing methods here. Given a persona text $\widetilde{p} \in \widetilde{P}$ in a distilled sample:

- Token-level editing: we randomly mask tokens with a pre-defined ratio, then use a pre-trained BERT (Devlin et al., 2019) model to make predictions on the masked positions, and the new tokens will take place of the old ones.
- Phrase-level editing: we mask the sentence tail with a random ratio and utilize a pre-trained single-direction language model GPT2 (Radford et al., 2019) to obtain a new rear part of $\widetilde{p}$.

Multiple edited persona texts can be obtained from a single $\widetilde{p}$ via sampling. We also finetune pre-trained models using original persona texts for few steps here, achieving a trade-off between semantic diversity and domain similarity. Figure 3 illustrate an editing case, showing that the new persona texts can effectively increase personality diversity.

To ensure a satisfactory fluency and novelty of a edited persona $\widetilde{p}^d$, we rate it via a scoring function:

$$s_p(\widetilde{p}^d) = \alpha(f_{PPL}(\widetilde{p}^d)) + (1-\alpha) f_{BS}(\widetilde{p}^d, \widetilde{p}). \quad (1)$$

Here $f_{PPL}(\cdot)$ calculates the normalized perplexity via a GPT2 model to measure the fluency, $f_{BS}(\cdot, \cdot)$ stands for calculating BERTScore (Zhang et al., 2019) to evaluate the similarity between two sentences. Lower values for both items are preferred, meaning higher fluency or novelty. α is a hyper-parameter. We rank all edited personas originated from the same $\widetilde{p}$ with the ascending order of their scores $s_p(\cdot)$, and select the top N_p ones to form the diversified persona set $\widetilde{P}^d = \{\widetilde{p}_i^d\}_{1 \le i \le N_p}$.

Response aligning The semantic of an edited persona text obtained above could change, thus the original response may not be suitable. Therefore, we need to get a new aligned response to recover the consistency. Two approaches are utilized to obtain response $\widetilde{r}^d$ for an edited persona text $\widetilde{p}^d$ and the corresponding distilled history utterance $\widetilde{h}$:

- Token-level editing: We observe that some overlapped tokens can be found between $\widetilde{p} \in \widetilde{P}$ and $\widetilde{r} \in \widetilde{R}$. If a token t_i of them have been edited to new token t_i' so as to form a new persona $\widetilde{p}^d$, we directly replace t_i in $\widetilde{r}$ with t_i' in the same positions, resulting a aligned response $\widetilde{r}^d$. An illustration figure can be found in Appendix A.2.
- Model predicting: If no overlapped tokens can be found, then token-level editing is not applicable. Here we employ a powerful GPT2-based encoder-decoder model (Cao et al., 2020) fine-tuned on the distilled data $\mathcal{D}^{dis}$ to predict responses with the given $\widetilde{p}^d$ and a dialogue history utterance $\widetilde{h}$.

Dialogue history augmentation To further extend the data scale, we would also manipulate the dialogue history $\widetilde{H}$. We can apply a similar method in persona editing to first edit the history utterances and then obtain the new coherent responses. But we find the diversity scarcity issue is not severe in $\widetilde{H}$. Hence, we use a simple sentence-level augmentation, back translation (BT) (Sennrich et al., 2016), to obtain variants of dialogue utterances, in which we consider their semantics are identical. $\widetilde{H}$ is translated into an intermediate language then back into the source language using a couple of translation models. The original dialogue history and N_h new ones obtained via BT compose the augmented dialogue history set $\widetilde{H}^d = \widetilde{H} \cup \{\widetilde{h}_j^d\}_{1 \le j \le N_h}$.

Combining the above three parts together, we can actually obtain new responses $\widetilde{R}^d = \{\widetilde{r}^d_{ij}\}$ by the permutations of each item from $\widetilde{P}^d$ and $\widetilde{H}^d$. To ensure the quality of each new sample $\widetilde{S}^d = (\{\widetilde{p}^d_i\}, \{\widetilde{h}^d_j\}, \{\widetilde{r}^d_{ij}\})$, we evaluate it with respect to fluency, persona consistency and history coherence:

$$s(\widetilde{S}^d) = \beta(-f_{PPL}(\widetilde{r}^d_{ij})) + \gamma f_e(\widetilde{p}^d_i, \widetilde{r}^d_{ij}) + (1-\beta-\gamma) f_c(\widetilde{h}^d_j, \widetilde{r}^d_{ij}), \quad (2)$$

where $f_{PPL}(\cdot)$ is the same as (1), $f_e(\cdot,\cdot)$ is the entailment probability of two items by the same NLI model in Sec 3.1 for consistency, and $f_c(\cdot,\cdot)$ indicates the coherence probability of two input using another NLI model (Dziri et al., 2019)(details in Appendix A.2). β and γ are hyper-parameters. We filter samples below a threshold T, and the remaining samples constitute diversified data $\mathcal{D}^{div}$. The whole augmented training dataset is the combination of two subsets, $\mathcal{D}^a = \mathcal{D}^{dis} \cup \mathcal{D}^{div}$. The quality of augmented samples is discussed in Appendix B.

3.3 Data curriculum

During inference, the model should be capable to handle testing data, which has the same format as the original data $\mathcal{D}$ with multiple persona texts and history utterances. Therefore, we should not train a model using $\mathcal{D}^a$ only, but using both $\mathcal{D}^a$ and $\mathcal{D}$ instead. Unlike previous studies that treat the original and augmented data equally and mix them, we design a curriculum strategy to utilize data. Considering the difficulty of learning on different data, we treat $\mathcal{D}^a$ as an easy curriculum while the original dataset $\mathcal{D}$ as a hard curriculum. Because we remove some distractors in $\mathcal{D}^a$. The model is trained on them successively to find better local minima.

4 Experiments

4.1 Experiment setup

Dataset We conduct experiments on **PersonaChat** dataset (Zhang et al., 2018a), following the former relevant work (Song et al., 2019, 2020; Wolf et al., 2019; Golovanov et al., 2019). Although other datasets may exist, it is the most commonly-accepted one and it is easy to make comparison. It contains 8,939/1,000/968 multi-turn dialogues in Train/Dev/Test set respectively, totally 164,356 utterances. We consider the SELF ORIGINAL set with fewer samples for a harder setting. Each sample has a dialogue history H with no more than 15 utterances ($M \leq 15$) and personas P between 4 to 6 sentences ($4 \leq L \leq 6$). For **D**^{3}, we set the distillation threshold $\tau = 0.99$, the edited persona number $N_p = 5$. A suitable filtering threshold T extends distilled data into about 200% of its original size in diversification[1]. Sample and token quantities in each stage during training are listed in Table 1.

	$\mathcal{D}$	$\mathcal{D}^{dis}$	$\mathcal{D}^{div}$	$\mathcal{D}^a$	$\mathcal{D}+\mathcal{D}^a$
#sample	65,719	26,693	26,700	53,393	119,112
#persona	4,710	4,522	9,788	14,310	14,498
#token	20,467	13,420	12,794	17,835	23,269

Table 1: Statistics of samples obtained in each stage.

Base models Two dialogue model architectures are considered: 1) Transformer (Vaswani et al., 2017): a Seq2seq model architecture using Transformer as the backbone with pointer generator (See et al., 2017) integrated; 2) GPT2: following Wolf et al. (2019), we involve pre-trained model but use GPT2 (Radford et al., 2019) as the backbone rather than GPT, which is one of the most powerful models on this task. Both models concatenate P and H as a single input sequence in which special symbols and token type embeddings are involved to distinguish between them. The negative log-likelihood loss is used to train models using Adam optimizer (Kingma and Ba, 2014).

Compared methods We pack base models with our **D**^{3} and other approaches that also try to boost a dialogue generation model at the data level: 1) **Back translation (BT)** (Sennrich et al., 2016): We perform BT on all sentences in a training sample, including the persona texts and utterances; 2) **CVAE** (Li et al., 2019): a CVAE-based method in which a model is trained on the original data, and then used to extend the corpus with generated texts via sampling different latent code. Since it can only handle query-response pairs, we concatenate all input as a single query to obtain new samples. 3) **Entropy Filter (filter)** (Csáky et al., 2019): it remove dull and general responses according to the entropy. We calculate entropy using the dialogue history and response without personas. As our base models can achieve competing performance among existing works, we do not focus on comparing with other network architectures. The details and statistics of each method are given in Appendix B.

4.2 Evaluation metrics

Automatic metrics We adopt multiple widely used metrics to measure the performance. Perplex-

[1]Our code will be available after publication.

Model	PPL	BLEU	NIST-4	BS$_f$	Ent-1	Ent-2	Ent-3	Dis-1	Dis-2	Dis-3	C	Flu.	Coh.	Pcon.
Human	-	-	-	-	5.680	8.913	10.27	5.259	34.90	66.37	0.472	2.625	2.451	0.531
Trans	38.28	3.140	1.148	0.1486	4.046	5.484	6.262	1.609	6.298	11.71	0.235	2.303	2.038	0.304
Trans-BT	37.92	3.315	1.082	0.1527	4.274	5.905	6.752	1.760	7.108	13.39	0.289	2.337	2.142	0.350
Trans-CVAE	37.61	3.312	1.191	0.1533	3.974	5.451	6.267	1.459	5.795	11.16	0.260	2.333	2.111	0.335
Trans-filter	38.99	2.946	1.101	0.1563	**4.283**	6.033	7.088	1.796	7.696	14.06	0.446	2.318	2.088	0.492
Trans-D^3	**37.30**	**3.358**	**1.206**	**0.1574**	4.223	**6.165**	**7.298**	**1.826**	**7.923**	**14.42**	**0.485**	**2.397**	**2.172**	**0.513**
Trans-D^3*	37.67	3.259	1.185	0.1554	4.197	6.095	7.232	1.794	7.835	14.27	0.439	2.378	2.164	0.481
GPT2	17.63	3.761	1.278	0.1693	4.485	6.187	7.029	2.011	8.260	15.03	0.518	2.508	2.243	0.508
GPT2-BT	16.96	3.943	1.348	0.1663	4.547	6.248	7.089	1.947	8.113	14.94	0.509	2.488	**2.259**	0.454
GPT2-CVAE	17.16	3.339	1.360	0.1592	4.245	5.691	6.490	1.748	6.799	12.19	0.484	2.358	2.150	0.426
GPT2-filter	16.90	3.734	1.337	0.1788	4.570	6.352	7.263	2.148	9.031	16.52	**0.571**	2.527	2.233	0.537
GPT2-D^3	**15.69**	**4.184**	**1.429**	**0.1835**	**4.614**	**6.426**	**7.321**	**2.267**	**9.803**	**18.20**	0.557	**2.532**	2.255	**0.548**
GPT2-D^3*	15.77	4.082	1.388	0.1809	4.611	6.408	7.312	2.209	9.657	17.91	0.536	2.525	2.249	0.527

Table 2: The main results of various methods on PersonaChat dataset using two base models. (Trans: Transformer, BLEU, Dist-n are %, * means using an NLI model trained on 200 labeled samples in data distillation.)

ity (**PPL**) indicates how well a model fits the test data. **BLEU** (Papineni et al., 2002) and **NIST-4** (Doddington, 2002) reflect the generation n-gram accuracy compared with references. BERTScore (Zhang et al., 2019) is also included to indicate the semantic similarity between the references and candidates. We use its F1 value here and rescale it to magnify the discrepancy (**BS**$_f$). To illustrate the diversity of responses, we use Distinct-n (Li et al., 2016a) (**Dist**, n=1,2,3) which is the ratio of unique n-grams among the corpus, and Entropy-n (Zhang et al., 2018b) (**Ent**, n=1,2,3) that is the entropy obtained via the n-gram distribution in a sentence. Moreover, **C-score** (Madotto et al., 2019) (**C**) is involved that uses the output of a trained NLI model to indicate the consistency between a response and provided personalities.

Human evaluation We randomly sampled 200 samples, which is a common quantity in former work, from the test set. Five professional annotators from a third-party company were asked to rate these responses in three dimensions: 1) **Fluency (Flu.)**; 2) **Coherence (Coh.)** with the dialogue history, 3) **Persona consistency (Pcon.)**. The scores for the former 2 dimensions are three-scale in which 1, 2, and 3 indicate unacceptable, moderate, and satisfactory respectively. The last one is binary where 1 means the response is consistent with at least one persona in the sample and 0 means irrelevant to anyone (We did not consider the contradict condition as it is very rare). The agreement rate from raters is 97.5%, 89.5%, 100% @3 for each dimension, proving the validity of scores.

4.3 Main results

We report the main results in Table 2. Compared to the base model or other data augmentation approaches, our **D**^{3} obtains the best persona consistency, e.g., 70% higher than the base Transformer. Our method shows less improvement on GPT2 than Transformer, but many former data-level methods even fail on GPT2. The reason is that Transformer is an end-to-end model while GPT2 is pre-trained on a huge corpus and data issues may have a less significant impact. Besides, **D**^{3} can improve the generation diversity, benefited by the diversification process. We notice that Entropy Filter also enhances persona consistency, yet it does not have consistent improvements on the metrics reflecting fluency and coherence. The reason is that fewer training samples are adopted by excluding the uninformative ones, which may still be useful to learn a general language model and a generic responding scheme. Moreover, we test the performance of **D**^{3} when using an NLI model under few-shot training (200 samples) in data distillation. It is still superior to most baselines, despite is a bit worse than **D**^{3} with sufficient NLI training data. And the response diversity nearly remains unchanged. Therefore, **D**^{3} also shows its value in more general applications where limited in-domain NLI labels are available.

4.4 More analysis

In this section, we further validate the contributions made by different components in our method **D**^{3} by analyzing the following **three questions**:

1. whether there is a need to construct simple persona-consistent data $\mathcal{D}^{dis}$ as in data distillation;
2. whether data diversification can effectively promote the diversity of distilled data;
3. whether the curriculum strategy better involves augmented data and benefits the model learning.

We use the results on Transformer here for discussion in the following part, and results of GPT2

	PPL	BLEU	NIST-4	BS_f	Ent-1	Ent-2	Ent-3	Dis-1	Dis-2	Dis-3	C
Trans	38.28	3.140	1.148	0.1486	4.046	5.484	6.262	1.609	6.298	11.71	0.235
Trans-$\mathbf{D}^3$	37.30	3.358	1.206	0.1574	4.223	6.165	7.298	1.826	7.923	14.42	0.485
w/o diversification	37.90	3.159	1.105	0.1511	4.051	5.664	6.533	1.570	6.992	13.42	0.454
w/o distilled format	38.25	3.105	1.126	0.1499	4.026	5.459	6.290	1.495	6.131	11.76	0.352
only distillation	104.8	1.509	0.939	0.1059	4.002	5.398	6.265	1.279	4.630	8.505	0.637
w/o persona editing	37.96	3.284	1.136	0.1535	4.171	5.686	6.517	1.608	6.599	12.62	0.422
w/o history augmentation	38.10	3.291	1.222	0.1550	4.150	5.759	6.560	1.608	6.493	12.52	0.461
w/o response filter	38.21	3.106	1.087	0.1503	4.207	5.841	7.080	1.592	6.991	12.98	0.399

Table 3: The results of automatic metrics when using $\mathbf{D}^3$ distillation variants (middle), and data diversification ablations (lower), compared with the original $\mathbf{D}^3$ (top) on Transformer. (BLEU, Dist-n are %.)

will be discussed in Appendix C.2. We use automatic metrics here. Despite they are not so reliable among different model architectures, they can basically reflect the performance gaps under the same architecture based on our observation in Table 2.

Analysis of data distillation In order to examine the effectiveness of data distillation, we need to neutralize the influence of data diversification as it is only applicable to distilled data. Following variants of $\mathbf{D}^3$ are considered: 1) *w/o diversification*, in which only distilled data $\mathcal{D}^{dis}$ is used to form the easy curriculum without diversified data $\mathcal{D}^{div}$. 2) *w/o distilled format*, based on 1), we recover samples in $\mathcal{D}^{dis}$ into their original formats which means multiple persona texts and history utterances are included. 3) *only distillation*, only $\mathcal{D}^{dis}$ is used in training while the original data $\mathcal{D}$ is not used.

Results of these variants are shown in the middle part of Table 3. Obviously, removing data diversification will hurt the performance in all aspects as the scale of training data decreases. If we further remove the simplified format in data distillation and use them in the original forms, the model will perform even worse especially on **C** score. Although $\mathcal{D}^{dis}$ only contains responses that are consistent with at least one persona which should be easier for model learning than the original data, totally relying on it is not enough. The reason is that the distilled samples without the original training data encourage the model to focus more on the personas while ignoring other aspects in dialogue. Therefore, despite only using distilled data in training can promote **C** score, it significantly degenerates the model in other aspects. That is why we utilize curricula that cover the original data format.

Analysis of data diversification From Table 1, we see that the diversified data contains many new persona texts as well as tokens. Besides, we compute the Novelty metrics (Wang and Wan, 2018) of diversified samples taking the original distilled samples

text type	Novelty-1, 2, 3, 4			
persona	40.26	62.17	70.47	77.81
utterance	26.20	39.52	45.48	50.56
all	30.89	47.07	53.81	59.64

Table 4: Novelty metrics of each part in diversified data $\mathcal{D}^{div}$ compared to the original distilled data $\mathcal{D}^{dis}$.

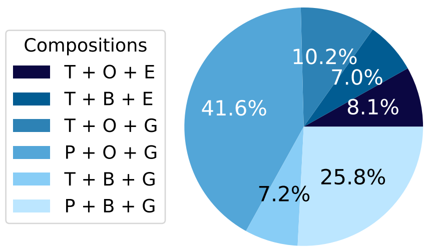


Figure 4: The compositions of diversified data. (T/P: token/ phrase-level editing to get edited personas, O/B: original/ BT-augmented dialogue history, E/M: token editing/ model predicting to get responses.)

as references, indicating the frequency of newly-appeared n-grams. Results in Table 4 again demonstrate that new language patterns are involved.

To further validate the effectiveness of each part of data diversification, we conduct ablation studies and considering the following conditions: 1) *w/o persona editing*: no new persona will be generated during data diversifying; 2) *w/o history augmentation*: only original dialogue history is used to obtain the diversified data $\mathcal{D}^{dis}$; 3) *w/o response filtering*: all new responses are directly used as diversified samples without filtering. Results of these ablations are shown in the lower part of Table 3. All these parts contribute to the performance of the whole method in various aspects. Response filter is the most important one as it ensures the quality of new samples so it affects both the n-gram accuracy and persona-consistency. Introducing new personas and paraphrased history are both beneficial for generation diversity. The former one has a significant effect on **C** score as novel persona texts benefit model robustness on persona consistency.

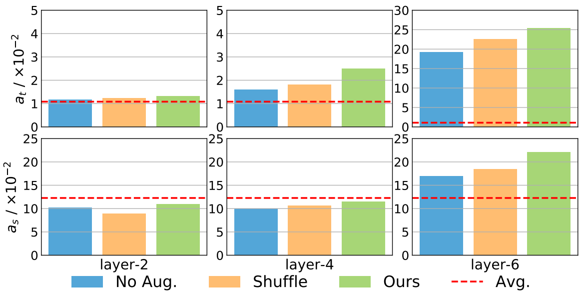


Figure 5: Average consistent attention weight in different decoder layers of Transformer trained without augmentation (No Aug.), using shuffling data of $\mathcal{D}$ and $\mathcal{D}^a$ (Shuffle) or our curriculum. Upper: token-level a_t, lower: sentence-level a_s, Avg.: average values if the attention distributes on all positions evenly as a baseline.

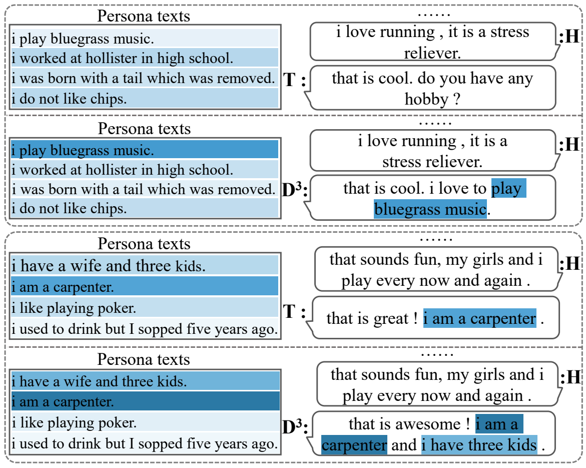


Figure 6: Response cases and visualized attention on personas. **T**:Transformer, **D**^{3}:Transformer-**D**^{3}.

The proportions of diversified samples coming from various source combinations are shown in Figure 4. As can be seen, more than 80% diversified samples have their responses obtained via model predicting as token editing sets a strict condition that overlapped tokens must exist. And phrase-level editing contributes to more high-quality personas with good fluency and semantic novelty.

Analysis of data curriculum To demonstrate the effect of training using the designed data curriculum, we try other variants by shuffling two kinds of data together (original data $\mathcal{D}$ and augmented data $\mathcal{D}^a$), or using a reverse curriculum order. Our method obtains consistently the best performance among them on all metrics without a doubt. And detailed results can be found in Appendix C.3.

We want to further quantify the effect of curriculum training on models using the attention from the response on the persona texts. We define two metrics, token-level / sentence-level consistent attention weight (a_t and a_s), to measure how it contributes to reflecting the proper personas. Recall that the concatenation of multiple persona texts P and history utterances H as the model input. Personas are firstly distilled like Sec 3.1 for each sample. We record the token positions of the entailed persona texts in the input sequence, forming a set $\mathcal{S}$ for a_t. Then for each index s in $\mathcal{S}$, if its corresponding token is the same as one token in the response, we put their index pair into a set $\mathcal{T} = \{(s, l)\}$, where s and l denote the token position index in the input sequence and the response respectively. Then we have two measurements for each sample

$$a_t = \frac{1}{|\mathcal{T}|} \sum_{(i,j) \in \mathcal{T}} a_{ij}, \quad a_s = \frac{1}{Y} \sum_{i=1}^{Y} \sum_{j \in \mathcal{S}} a_{ij}, \quad (3)$$

where $a_{ij} \in [0, 1]$ is the normalized scalar attention weight at the i-th decoding step on the j-th input token, i.e. $\sum_j a_{ij} = 1$, and Y is the length of generated response.

A higher a_t or a_s indicates that the model poses more attention on tokens or sentences of proper personas. Attention comparison on the average of all applicable samples from the dev set is shown in Figure 5. Our method shows the highest a_t and a_s on all layers. The superiority is more significant in higher layers, while the attentions of all models tend to distribute uniformly in lower layers.

Some case studies are shown in Figure 6 to demonstrate promotion brought by **D**^{3} on Transformer. Here **H** indicates dialogue history, a darker color on a persona text denotes that a higher attention weight is posed by the model. Obviously, **D**^{3} offers a model with the capability to pose more accurate and rich attention on the persona texts. More cases can be found in Appendix C.4.

5 Conclusion

We target the challenging personalized dialogue generation task. Unlike previous work that designs a powerful network to improve performance, we carefully analyze the difficulty of using current training data to get a good model. Based on the understanding, we propose a data-level augmentation method **D**^{3} to promote the existed model without model-level modification. It first distills the original data and then augment both the amount and diversity of the distilled data. A curriculum training is then applied to utilize both augmented and original data. Automatic metrics and human evaluation show that **D**^{3} effectively improve the performance of two powerful base model structures.

References

Reina Akama, Sho Yokoi, Jun Suzuki, and Kentaro Inui. 2020. Filtering noisy dialogue corpora by connectivity and content relatedness. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 941–958.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations*.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48.

Paweł Budzianowski and Ivan Vulić. 2019. Hello, it's gpt-2-how can i help you? towards the use of pretrained language models for task-oriented dialogue systems. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 15–22.

Hengyi Cai, Hongshen Chen, Yonghao Song, Cheng Zhang, Xiaofang Zhao, and Dawei Yin. 2020a. Data manipulation: Towards effective instance learning for neural dialogue generation via learning to augment and reweight. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6334–6343.

Hengyi Cai, Hongshen Chen, Cheng Zhang, Yonghao Song, Xiaofang Zhao, Yangxi Li, Dongsheng Duan, and Dawei Yin. 2020b. Learning from easy to complex: Adaptive multi-curricula learning for neural dialogue generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7472–7479.

Yu Cao, Wei Bi, Meng Fang, and Dacheng Tao. 2020. Pretrained language models for dialogue generation with multiple input sources. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 909–917.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *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.

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.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram cooccurrence statistics. In *Proceedings of the second international conference on Human Language Technology Research*, pages 138–145.

Nouha Dziri, Ehsan Kamalloo, Kory W Mathewson, and Osmar Zaiane. 2019. Evaluating coherence in dialogue systems using entailment. *arXiv preprint arXiv:1904.03371*.

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

Sergey Golovanov, Rauf Kurbanov, Sergey Nikolenko, Kyryl Truskovskyi, Alexander Tselousov, and Thomas Wolf. 2019. Large-scale transfer learning for natural language generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6053–6058.

Hongyu Guo, Yongyi Mao, and Richong Zhang. 2019. Augmenting data with mixup for sentence classification: An empirical study. *arXiv preprint arXiv:1905.08941*.

Donghoon Ham, Jeong-Gwan Lee, Youngsoo Jang, and Kee-Eung Kim. 2020. End-to-end neural pipeline for goal-oriented dialogue systems using gpt-2. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 583–592.

Yutai Hou, Yijia Liu, Wanxiang Che, and Ting Liu. 2018. Sequence-to-sequence data augmentation for dialogue language understanding. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1234–1245.

Wan-Ting Hsu, Chieh-Kai Lin, Ming-Ying Lee, Kerui Min, Jing Tang, and Min Sun. 2018. A unified model for extractive and abstractive summarization using inconsistency loss. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 132–141.

Urvashi Khandelwal, He He, Peng Qi, and Dan Jurafsky. 2018. Sharp nearby, fuzzy far away: How neural language models use context. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 284–294.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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.

Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and William B 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.

Juntao Li, Lisong Qiu, Bo Tang, Dongmin Chen, Dongyan Zhao, and Rui Yan. 2019. Insufficient data can also rock! learning to converse using smaller data with augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6698–6705.

Pierre Lison and Jörg Tiedemann. 2016. Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 923–929.

Lemao Liu, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2016. Neural machine translation with supervised attention. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3093–3102.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

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.

Junghyun Min, R Thomas McCoy, Dipanjan Das, Emily Pitler, and Tal Linzen. 2020. Syntactic data augmentation increases robustness to inference heuristics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2339–2352.

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

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 on association for computational linguistics*, pages 311–318.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing*, pages 1532–1543.

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom Mitchell. 2019. Competence-based curriculum learning for neural machine translation. 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 1162–1172.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

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*.

Chinnadhurai Sankar, Sandeep Subramanian, Christopher Pal, Sarath Chandar, and Yoshua Bengio. 2019. Do neural dialog systems use the conversation history effectively? an empirical study. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 32–37.

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.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96.

Haoyu Song, Yan Wang, Weinan 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.

Haoyu Song, Wei-Nan Zhang, Yiming Cui, Dong Wang, and Ting Liu. 2019. Exploiting persona information for diverse generation of conversational responses. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 5190–5196.

Yi Tay, Shuohang Wang, Anh Tuan Luu, Jie Fu, Minh C Phan, Xingdi Yuan, Jinfeng Rao, Siu Cheung Hui, and Aston Zhang. 2019. Simple and effective curriculum pointer-generator networks for reading comprehension over long narratives. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4922–4931.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Ke Wang and Xiaojun Wan. 2018. Sentigan: generating sentimental texts via mixture adversarial networks. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 4446–4452.

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.

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

Benfeng Xu, Licheng Zhang, Zhendong Mao, Quan Wang, Hongtao Xie, and Yongdong Zhang. 2020. Curriculum learning for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6095–6104.

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.

Rongsheng Zhang, Yinhe Zheng, Jianzhi Shao, Xiaoxi Mao, Yadong Xi, and Minlie Huang. 2020. Dialogue distillation: Open-domain dialogue augmentation using unpaired data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 3449–3460.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018a. 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.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Yizhe Zhang, Michel Galley, Jianfeng Gao, Zhe Gan, Xiujun Li, Chris Brockett, and Bill Dolan. 2018b. Generating informative and diverse conversational responses via adversarial information maximization. pages 1810–1820.

Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. 2017. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 654–664.

A Implementation Details of $\mathbf{D}^3$

A.1 Details of distillation

In order to obtain the NLI model to determine the persona consistency, the RoBERTa-Large-MNLI[2] model having 24 layers and 1024 hidden size is utilized. To make the model can better fit the domain of PersonaChat, we fine-tune the model on the DialogueNLI dataset (Welleck et al., 2019) who has the same corpus as PersonaChat. We set batch size as 32 and finetune it for 5 epochs using learning rate 1e-5. We use the whole training set for the default $\mathbf{D}^3$ and obtain a model RoBERTa$_{nli}$ achieving 90.8% accuracy on the dev set. This model will also be responsible for calculating entailment probability $_{e}$ in response filtering and **C** score in the experiments. $\tau = 0.99$ is used here to filter low-confident samples. For the few-shot setting $\mathbf{D}^3$*, we randomly sample 200 samples from the training set to train the model using learning rate 2e-5, and obtain an NLI model achieving 79.3% on the dev set.

A.2 Details of diversification

BERT-based-uncased model[3] and base GPT2[4] are involved as the pre-trained model in this stage for persona editing and quality evaluation. To ensure that the pre-trained model can make predictions that better fit the current domain while also has enough uncertainty for generation diversity, we 1) fine-tune BERT and GPT2 on the persona sentences for 100 steps with batch size 32 and learning rate 1e-4, obtaining BERT$_{per}$ and GPT2$_{per}$; 2) fine-tune GPT2 on the responses for 200 steps with batch size 32 and learning rate 1e-4 and obtain GPT2$_{res}$. **Persona editing** BERT$_{per}$ and GPT2$_{per}$ will be used for token-level and phrase-level respectively, each will generate 10 unique new personas for each original persona text via sampling according to multinomial distribution. At token level, we only mask the most informative tokens which can be decided by the POS tags given by SpaCy[5] as it is

[2]https://huggingface.co/roberta-large-mnli

[3]https://huggingface.co/bert-base-uncased

[4]https://huggingface.co/gpt2

[5]https://spacy.io/

POS tags	VERB, NOUN, PROPN, NUM, ADV, ADP, ADJ

Table 5: The target POS tags for token-level masking.

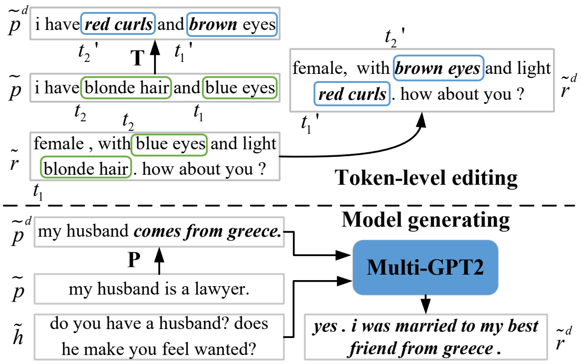


Figure 7: Aligning responses for new personas via token-level editing or model generating. **T/P**: edit persona in token/phrase level.

meaningless to mask some words, e.g. prepositions “to", “in" or articles “a", “the". The target POS tags are listed in Table 5. We set the token-level mask ratio ρ^t as 0.8 in our implementation. At phrase level, the mask ratio ρ^p is randomly sampled between $[0.3, 0.6]$. We also restrict that at least 2 tokens are masked and the maximum length of generated ones from GPT2$_{per}$ are not exceed 30% of the original length to ensure a similar style.

In filtering, we use $\alpha = 0.4$ to calculate the score $s_p(\widetilde{p}^d)$, where f_{PPL} is given by GPT2$_{per}$ and then normalized by a constant $C_p = 50$. When comes to BERTScore, the F1 value is used as f_{BS} while other configurations follow the recommendation for English. And N_p is set as 5 which means 5 new personas with the lowest s_p originated from the same original persona are remained in $\widetilde{P}^d$. Note that we obtain edited personas for each unique distilled persona text rather than each distilled sample.

Response aligning Given the permutations of pseudo personas and dialog history utterances from different sources, we only apply token-level editing on persona-history pairs whose source distilled sample contains consistent tokens exist between $\widetilde{P}$ and $\widetilde{H}$. The POS tags of these tokens are also restricted according to Table 5 to avoid the influence of common words such as "i" or "is". Then editing will be processed on the corresponding positions in the original responses, replacing old tokens with new ones to get aligned responses. For model-based generating, we train the Multi-GPT2 model on the distilled data $\mathcal{D}^{dis}$. Its performance on the dev set distilled from the original dev set of PersonaChat is shown in Table 6. We can found that this model shows high n-gram accuracy and persona consistency which should be effective. Figure 7 demonstrates the two approaches to obtain new responses.

Dialogue history augmentation we use the *transformer_wmt_en_de* Transformer model in Fairseq[6] as the translation model, who has 6 layers in both encoder and decoder. It is trained on the WMT14 EN-FR dataset with 40.5M samples. All configurations follow the default ones and the training step number is 10000. During inference, we use beam search with size 5 for both en-fr and fr-en translation, resulting in 25 new utterances for each original one. For a larger divergence, we selected $N_p = 1$ new utterance with the lowest BLEU score when taking the original one as the reference.

Filtering We use GPT2$_{res}$ to get the PPL of responses, regarded as f_{PPL}.A constant $C_r = 50$ is used to normalize it. Based on the previous study that a NLI model can also be used to determine the coherence between utterances (Dziri et al., 2019), we fine-tune another RoBERTa-Large-MNLI model on InferConvAI dataset[7] which achieves 88.7% accuracy on the dev set. The entailment probability given by this model is regarded as f_c. We set $\beta = 0.2$, $\gamma = 0.6$ as the persona consistency is our first priority.

Quality of diversified samples To prove the quality of generated responses in diversification, we employ GPT2-based PPL and NLI model-based score (similar as Filtering) to measure its fluency and coherence to query respectively. We compare the results with original responses from the training set, which are shown in Table 7.

In addition, we also evaluate the GPT2-PPLs for edited and original persona texts, which are 6.427 vs. 10.426. The edited ones has lower PPL due to filtering.

B Details of Experiment

Base model For Transformer model, we use 300-dim GloVe (Pennington et al., 2014) trained on 6B corpus as the word embeddings. There are 6 layers in both the encoder and decoder, whose hidden size is also 300 and the head number is 4. During training, a cross-entropy loss is used along with Label Smoothing whose ratio is 0.1. For GPT2 model, we use the base pre-trained model with 12 layers and 768-dim hidden state. It will be trained using

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

[7]https://github.com/nouhadziri/DialogEntailment

	PPL	BLEU	NIST-4	BS$_f$	Ent-1	Ent-2	Ent-3	Dis-1	Dis-2	Dis-3	C
Multi-GPT2	17.70	6.186	1.4773	0.3216	4.665	6.809	7.704	4.111	15.693	27.115	0.850

Table 6: The performance of trained Multi-GPT2 on the distilled dev set. (Dist-n and BLEU are in %.)

	GPT2-PPL	Coherence score
Original	13.119	0.361
Diversified	18.847	0.525

Table 7: The average GPT2-based PPL and NLI model-based coherence score of the original responses and responses generated in diversification.

method	Train sample number
Original	65,719
BT	131,436
CVAE	131,436
Entropy-Filter	59,892
	53,393 (easy)
D^3(Ours)	65,719 (hard)
	119,112 (all)

Table 8: The training sample number used for each method.

the average of a cross-entropy loss on generating and a classification loss between true response and one randomly sampled negative response. Beam search whose size 3 along with length penalty is used during inference for both models.

The formats of input or response for both models are shown in Figure 8. Here <bos>, <eos>, <talker1>, and <talker2> are special symbols for distinguishing different part of input or response. And for an augmented sample (P^a, H^a, R^a), P^a, H^a and R^a only contain a single persona text p_a, a single history utterance h_a and a single response r_a respectively.

Model training We use learning rate 2e-4 for Transformer and 6.25e-5 for GPT2, while the batch size is 256 for both models. Training will be stopped until the loss on the dev set does not decrease for N epochs. Here N is 15 for Transformer and N is 5 for GPT2. In curriculum learning, the learning rate is the same for different curricula. The dev set of the easy curriculum is obtained by applying the same augmentation to the original dev set. The best model obtained on the easy curriculum is used as the initial model in the hard curriculum. All experiments are implemented via PyTorch on one 32GB NVIDIA V100 GPU. Each epoch takes about 10 min for Transformer and 25min for GPT2. All **hyper-parameters are determined empirically using a coarse-grained grid search** to ensure satisfactory performance.

Baselines We apply the same translation models as the ones used in A.2 for the **BT** (Sennrich et al., 2016) baseline, extending each sample with a new sample originated from it which is consisted of texts that have the lowest BLEU scores to the original one. For CVAE (Li et al., 2019) method, we use the same default settings to train the model on PersonaChat dataset without using the persona texts. New samples are sampled having the same quantity as the original dataset. In Entropy-filter (Csáky et al., 2019), we set the threshold as 1.1 and using both source and target directions for filtering. Only the samples that survived after filtering is used in training. The whole training sample numbers of all methods are listed in Table 8. Note that **all models are trained until the loss does not decrease for** N **epoch patience for a fair comparison**.

Metrics BERTScores presented in our experiments are F1 values implemented using the default setting and official script with rescale[8]. RoBERTa$_{nli}$ obtained before is used here to calculate **C score**.

C Additional Experimental Results

C.1 Attention on dialogue history

To confirm how models pose attention on each part of dialogue history especially the last utterance, we calculate the attention weight from different decoder layers on the last utterance or other utterances except the last one of dialogue history. Transformer model is used here, which is trained with the original training data without any augmentation. The sentence-level attention is the summation of all attention weight within the goal sentences, while the token-level value is the average of weights among all tokens. Results are shown in Figure 9, obtained on the dev set of PersonaChat. Obviously, the last utterance in history obtains more attention, while other parts obtain less than the average value, especially at the token level. It proves the meaning of our dialogue history distillation.

C.2 Analysis of ablations on GPT2

We also provide the extensive results of ablation experiments on GPT2 which is similar to the ones given in Section 4.4 on Transformer. Table 9 illus-

[8]https://github.com/Tiiiger/bert$_s$$core$

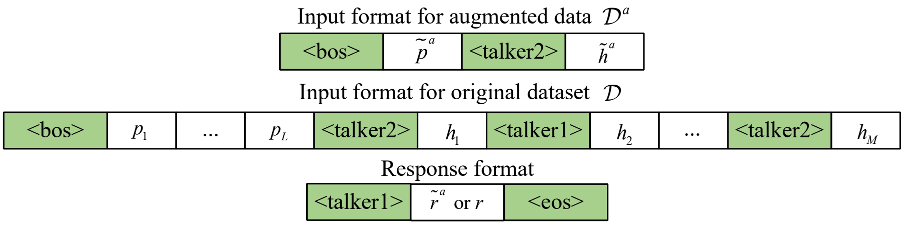


Figure 8: The sequence format of input and expected target for both Transformer and GPT2 models.

	PPL	BLEU	NIST-4	BS_f	Ent-1	Ent-2	Ent-3	Dis-1	Dis-2	Dis-3	**C**
GPT2	17.63	3.761	1.278	0.1693	4.485	6.187	7.029	2.011	8.260	15.03	0.518
GPT2-$\mathbf{D}^3$	15.69	4.184	1.429	0.1835	4.614	6.426	7.321	2.179	9.458	17.72	0.557
w/o diversification	15.89	4.119	1.441	0.1817	4.526	6.281	7.148	2.131	9.243	17.11	0.528
w/o distilled format	16.04	4.026	1.379	0.1788	4.462	6.151	7.097	2.017	9.022	16.86	0.518
only distillation	29.73	2.912	1.325	0.1509	4.558	6.392	7.250	1.252	4.807	9.048	1.131
w/o persona editing	15.81	*4.190*	1.427	0.1801	4.503	6.204	7.062	2.065	8.867	16.83	0.524
w/o history augmentation	15.75	*4.213*	*1.503*	0.1812	4.562	6.333	7.244	2.057	9.131	17.34	0.533
w/o response filter	15.83	4.119	1.395	0.1790	4.604	6.387	7.265	2.158	9.414	*17.74*	0.518

Table 9: The results of automatic metrics when usingD3distillation variants (middle), and data diversification ablations (lower), compared with the original $\mathbf{D}^3$(top) on Transformer. (BLEU, Dist-n are %.)

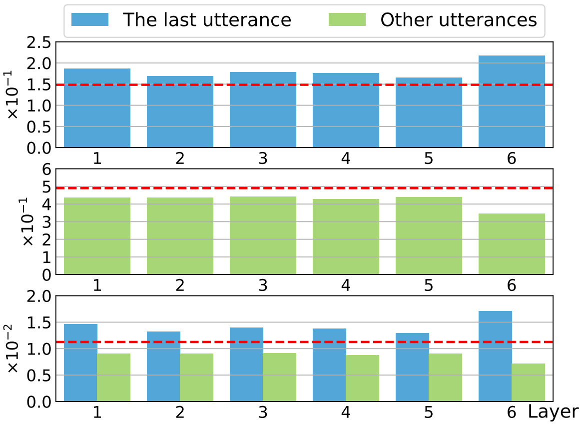


Figure 9: The sentence-level or token-level attention weights from different decoder layers in Transformer on different parts of dialogue history. Upper: sentence-level attention on the last utterance, mid: sentence-level attention on other utterances: token-level attention on each part. Red lines: the baseline values when all attention distributes evenly among all tokens.

trates the results when applying different variants related to distillation. We can found the influence of data diversification, as well as our distillation. But both of them have a similar but less effect on GPT2 compared to Transformer. The reason is that GPT2 is a very powerful pre-trained model that is less dependent on the training data.

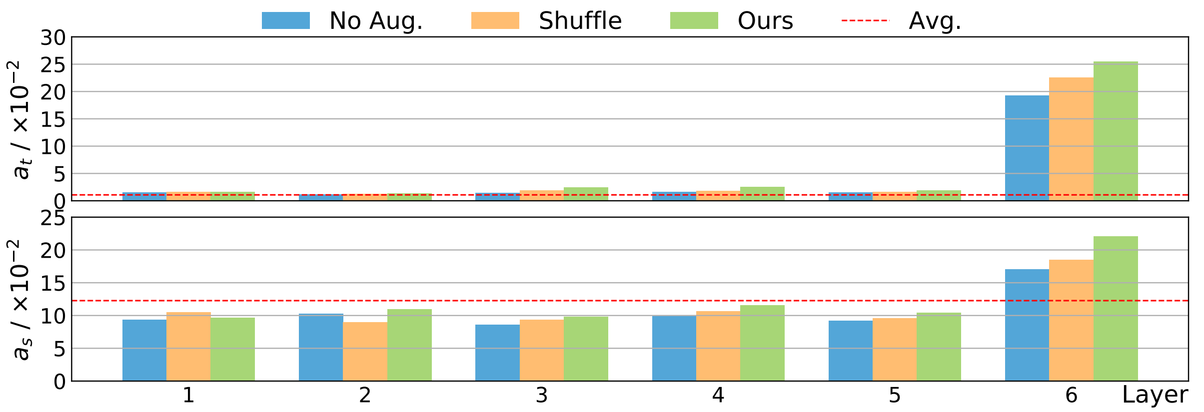
The results of ablation studies in the data diversification module on GPT2 are shown in Table 9. The performance gaps between them are also narrowed compared to the results when using Transformer as the base model. But the similar conclusions can still be drawn that response filter has a relatively more important contribution, while persona editing affects the generation diversity as well as persona consistency. History augmentation has the least significant influence.

C.3 Detailed results of curriculum analysis

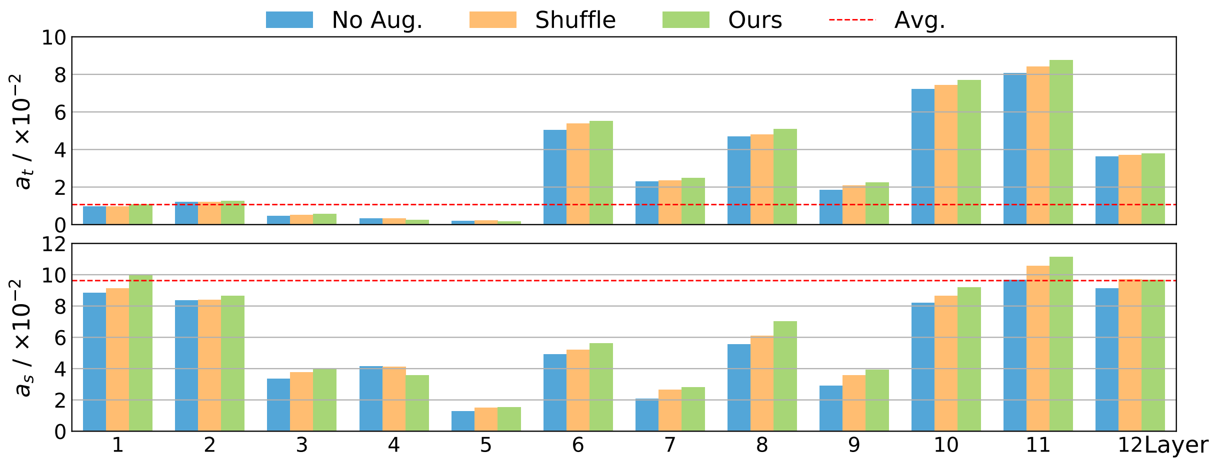
We test several variants of our data curriculum: 1) No augment.: only the original dataset $\mathcal{D}$ (the hard curriculum) is used, it is equal to the original model; 2) Only augment.: only the augmented dataset $\mathcal{D}^a$ (the easy curriculum) is used; 3) Shuffle: shuffling of the original dataset $\mathcal{D}$ and the augmented dataset $\mathcal{D}^a$ together to train the model; 4) Reverse: using the curricula in a reverse order, which means the hard curriculum is used first to train the model.

The results of these variants along with our $\mathbf{D}^3$ on both Transformers and GPT2 are shown in Table 10. There is no doubt that our curriculum is the best when comprehensively considering all aspects. Although Aug. and Reverse show high **C** scores, their responses are much worse in n-gram accuracy as they involve more personas while focusing less on the dialogue coherence during generating. Shuffle shows an intermediate performance between our $\mathbf{D}^3$ and No Aug. as it includes more simplified persona-consistent training data which may benefit the training. But the mixing strategy is not so efficient as the data curriculum.

We also provide the token-level/ sentence-level consistent attention weights a_t and a_s in all layers



(a) Consistent attention weights from different decoder layers in Transformer. Upper: token-level a_{tc}, lower: sentence-level a_{sc}.



(b) Consistent attention weights from different decoder layers in GPT2. Upper: token-level a_{tc}, lower: sentence-level a_{sc}.

Figure 10: Consistent attention weights on Transformer and GPT2. No Aug.:training the model without augmented data; Shuffle: training the model using the shuffling data of $\mathcal{D}$ and $\mathcal{D}^a$; Ours: training the model using our curriculum strategy; Avg.: average values if the attention distributes on all positions evenly as a baseline.

	PPL	BLEU	NIST-4	BS$_f$	Ent-1	Ent-2	Ent-3	Dis-1	Dis-2	Dis-3	C
Trans-**D**^{3}	37.30	3.358	1.206	0.1554	4.223	6.165	7.298	1.826	7.923	14.42	0.485
No augment	38.28	3.140	1.148	0.1486	4.046	5.484	6.262	1.609	6.298	11.71	0.235
Only augment	126.3	1.603	0.956	0.0852	4.315	6.309	7.426	1.747	7.530	12.66	0.942
Shuffle	37.66	3.203	1.175	0.1521	4.128	6.096	6.979	1.659	6.889	13.79	0.404
Reverse	48.17	2.137	1.019	0.1508	3.947	5.291	6.039	1.368	5.503	9.211	0.912
GPT2-**D**^{3}	15.69	4.184	1.429	0.1835	4.614	6.426	7.321	2.179	9.458	17.72	0.557
No augment	17.63	3.761	1.278	0.1693	4.485	6.187	7.029	2.011	8.260	15.03	0.518
Only augment	33.01	2.540	1.078	0.1035	4.574	6.255	7.232	1.916	7.340	11.77	1.148
Shuffle	16.58	3.801	1.321	0.1799	4.588	6.261	7.216	2.128	9.391	17.55	0.525
Reverse	30.46	2.615	1.069	0.1189	4.298	6.074	6.960	1.646	6.709	9.529	1.111

Table 10: Performance comparison between different curriculum strategies on two base models. (Top: Transformer, bottom: GPT2, BLEU, Dist-n are %.)

of Transformer and GPT2 trained via No Aug., Shuffle data or our **D**^{3} method, which are shown in Figure 10. Our method has the most accurate attention on personas at both levels. Compared to Transformer, the divergence between different layers in GPT2 is more significant.

C.4 More case studies

Except for the cases provided in Section4.4, we provide additional cases including the responses

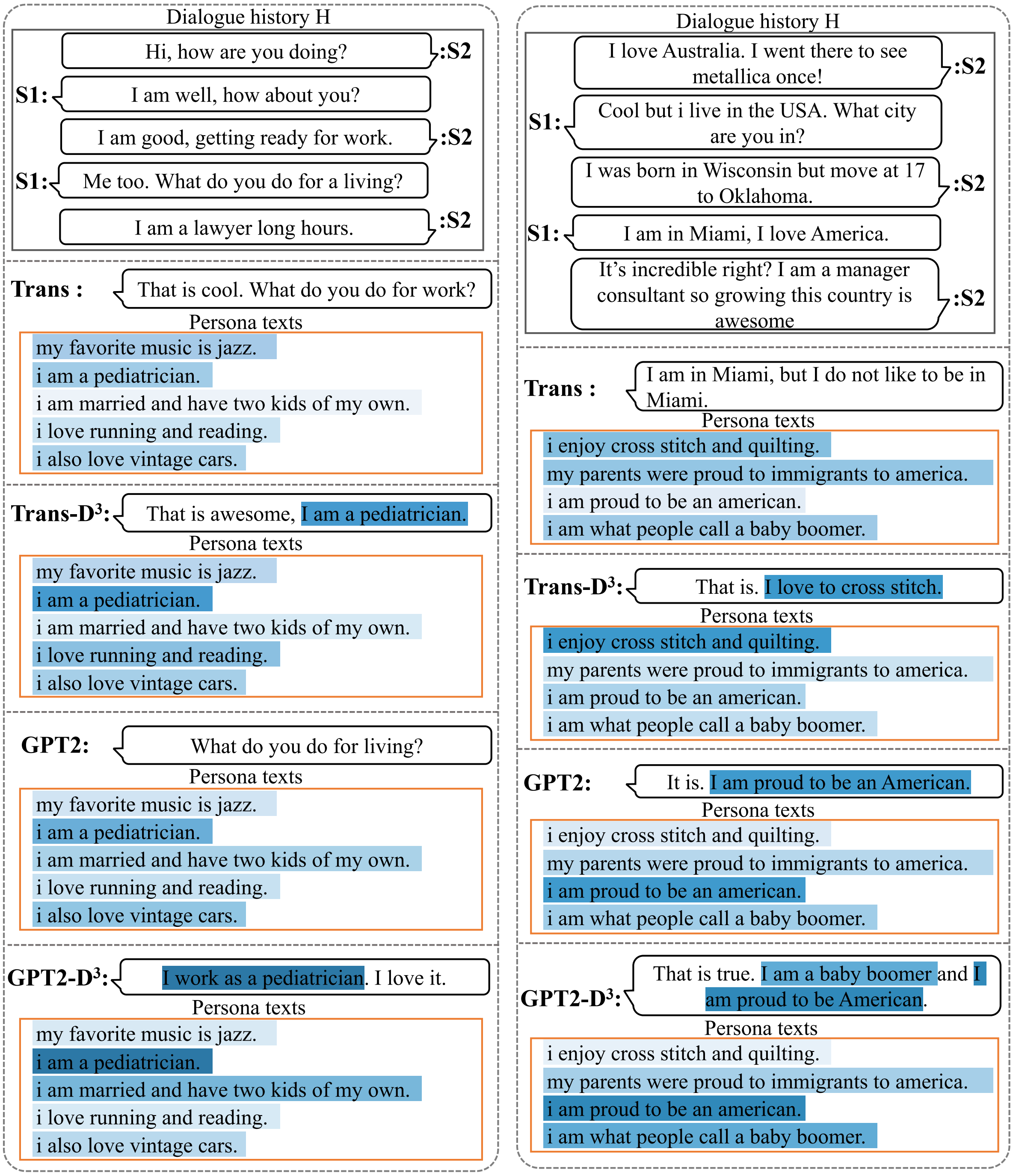


Figure 11: Additional responses cases and visualization by Transformers(Trans) and GPT2 without or with our $\mathbf{D}^3$ data augmentation method. Colors in each persona text indicate the attention weight paid by different models. A darker color means a higher attention weight is posed by the current model. Colored texts in the response denote the persona consistency.

given by GPT2. They are shown in Figure 11, including visualized attention weights posed by different models on their persona texts. Note that the attention weights are normalized along the whole input sequence including dialogue history. It can be found that our method can help the model to pay more attention to suitable persona texts, thus the generated responses are better in persona-consistency.