A Model-agnostic Data Manipulation Method for Persona-based Dialogue Generation

Anonymous ACL submission

Abstract

Towards building intelligent dialogue agents, there has been a growing interest in introducing explicit personas in generation models. However, with limited persona-based dialogue data at hand, it may be difficult to train a dialogue generation model well. We point out that the data challenges of this generation task lie in two aspects: first, it is expensive to scale up current persona-based dialogue datasets; second, each data sample in this task is more complex to learn with than conventional dialogue data. To alleviate the above data issues, we propose a data manipulation method, which is model-agnostic to be packed with any persona-based dialogue generation model to improve their performance. The original training samples will first be distilled and thus expected to be fitted more easily. Next, we show various effective ways that can diversify such easier distilled data. A given base model will then be trained via the constructed data curricula, i.e. first on augmented distilled samples and then on original ones. Experiments illustrate the superiority of our method with two strong base dialogue models (Transformer encoder-decoder and GPT2).

1 Introduction

The ability to generate responses with consistent personas is important towards building intelligent dialogue agents. In past years, there has been a growing interest in introducing explicit personas in dialogue generation models (Song et al., 2019; Wolf et al., 2019). A piece of persona text generally consists of profiles and background personal facts. A clipped persona-based dialogue from the PersonaChat (Zhang et al., 2018a) dataset is shown in Figure 1, which covers rich persona features. For a persona-based dialogue generation model, generated responses need to be relevant to the dialogue context as well as consistent with personas.

Most existing generation models for this task rely heavily on training with sufficient persona-based dialogues. However, available data are limited due to their expensive collection costs. Take the PersonaChat as an example, two crowd-sourced annotators are hired to play the part of a provided persona and converse naturally with each other. In total, about 162 thousand dialogue utterances are collected with less than 5 thousand unique persona profiles. Compared with conventional dialogue datasets such as OpenSubtitles (Lison and Tiedemann, 2016) and Weibo (Shang et al., 2015) with millions of data, persona-based dialogue datasets are relatively small.

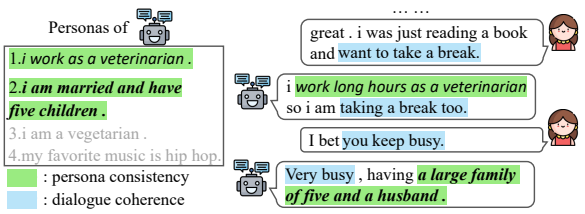


Figure 1: Each response in a persona-based dialogue is mostly related to one persona sentence and its latest dialogue history utterance. Persona sentences in grey are redundant for all responses.

Besides the limited data scale, another data issue we want to point out is that a persona-based dialogue is more complex to learn with, in comparison with conventional dialogues. Recall that a persona-based dialogue involves not only multiple dialogue utterances but also auxiliary persona sentences. Welleck et al. (2019) showed that not all responses in the PersonaChat dataset are consistent with the provided personas. This makes a model hard to capture a reliable mapping from training data. Supposing we apply a similar dialogue model as in conventional dialogue generation tasks with a comparable parameter size, we should expect more data to train a robust model on the more difficult data setting. Moreover, it may be difficult to use existing data augmentation methods (Li et al., 2019; Niu and Bansal, 2019) to automatically construct such complex persona-based dialogue data. For example, if we apply back translation (Sennrich et al.,

2016) to every sentence in persona-based samples, the augmented ones may not maintain the coherence between the dialogue history and the response as well as the consistency between the persona and the response simultaneously.

A few studies have been conducted to alleviate the above data issues by fintuning existing pretrained models such as GPT (Wolf et al., 2019; Golovanov et al., 2019) or BERT (Lin et al., 2020; Song et al., 2021). They often stick to a certain pretrained model. Sophisticated finetuning strategies, including proper network modifications and loss functions, are required to get satisfactory performance, making them not useful across different pretrained models. Moreover, they do not address the data difficulty issue explicitly. Most of them simply concatenate all persona and dialogue history sentences into a single input sequence for finetuning, and rely on the ability of the pretrained model to fast adapt to the target data domain. Hence, we want to design a model-agnostic method to address both the data scale and data difficulty issue, which can be packed with any base model, either trained from scratch or finetuned from a pretrained model.

In this work, we propose a novel data manipulation method for persona-based dialogue data, which is model-agnostic to be packed with any base generation model to improve their robustness and consistency. Our method includes three data manipulation operations, namely $\mathbf{D}^3$, in sequel: (i) **D**ata distillation: original training samples are simplified into contain only useful and less redundant persona sentences and dialogue utterances, which are expected to be fitted more easily; (ii) **D**ata diversification: with the easier distilled samples, we can also perform data augmentation more reliably. We design various methods to edit new personas, and then align them with new and consistent responses to improve data diversity; (iii) **D**ata curriculum: with both augmented distilled and original data at hand, we arrange them into a data curriculum for model learning (Bengio et al., 2009), where the base model is trained on the easier augmented distilled data and then the harder original data. To validate the effectiveness of our method, we perform experiments on two strong base dialogue models, Transformer-based encoder-decoder and GPT2.

2 Related Work

Persona-based dialogue generation It sees growing interest in recent years, thanks to the released benchmark datasets such as PersonaChat/ ConvAI2 (Zhang et al., 2018a; Dinan et al., 2020). Previous works mostly focus on modifying dialogue models to condition auxiliary persona information, including extra persona embedding(Li et al., 2016b), profile memory (Zhang et al., 2018a), copying from personas (Yavuz et al., 2019), and CVAE with persona information (Song et al., 2019).

Recent works try to adopt large-scale pretrained models on this task. GPT/GPT2 (Radford et al., 2018, 2019) are chosen the most often and shown to improve the generation quality with different finetuning strategies (Wolf et al., 2019; Golovanov et al., 2019; Cao et al., 2020). Some leverage BERT (Devlin et al., 2019) as backbones (Lin et al., 2020; Song et al., 2021). Other pretrained models also demonstrate their effectiveness (Roller et al., 2021; Lin et al., 2021). The aforementioned methods often need proper network modifications and finetuning loss functions in order to get satisfactory performance. It is hard to transfer them to be useful across different pretrained models. Moreover, most of them simply concatenate persona texts and dialogue history together as a single input sequence (Wolf et al., 2019; Roller et al., 2020), highly depending on the ability of the pretrained model to fast adapt on the target data domain.

Text data manipulation Various data augmentation methods have been widely used in many NLP tasks (Sennrich et al., 2016; Hou et al., 2018; Guo et al., 2019; Min et al., 2020), which are 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 studies the pairwise relationship between a query and a response to design the augmentation techniques, which are not applicable to involving 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 ma-

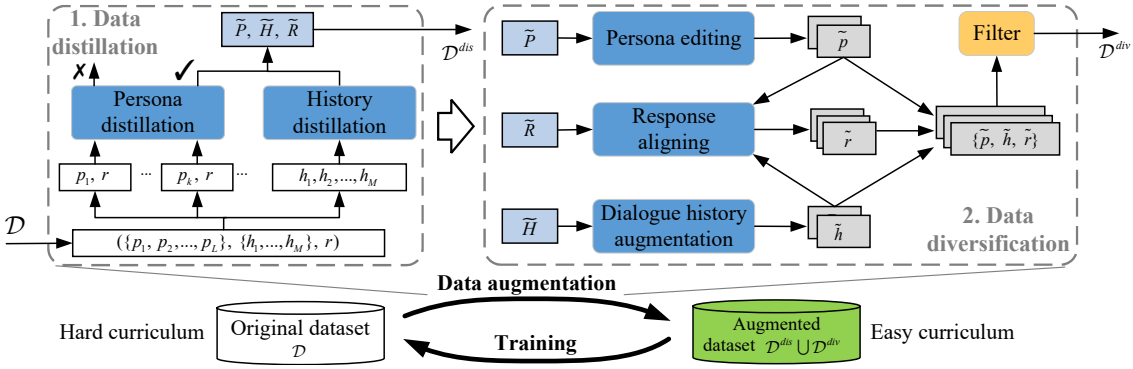


Figure 2: The framework of our data manipulation method $\mathbf{D}^3$. It obtains the augmented dataset $\mathcal{D}^a = \mathcal{D}^{dis} \cup \mathcal{D}^{div}$ from the original dataset $\mathcal{D}$ through data distillation and data diversification. Curriculum strategy is used to train a model by first learning on the easy augmented data $\mathcal{D}^a$and then on the hard original training data $\mathcal{D}$.

chine 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 distilled data regarding as a curriculum.

3 Our Data Manipulation Method

We first formally define a persona-based training sample. It consists of L persona description sentences $P = \{p_1, p_2, .., p_L\}$, M dialogue history utterances $H = \{h_1, h_2, .., h_M\}$, and a gold response R. The given training dataset is denoted as $\mathcal{D} = \{(P, H, R)\}$. Note that L and M in different training samples can be different. A dialogue model needs to generate a response $\hat{R}$, which is coherent with the dialogue history H and consistent with persona information in P.

Our proposed data manipulation method $\mathbf{D}^3$ is model-agnostic. For any dialogue model, we will not change the model itself but only manipulate its training data. We develop three data manipulation operations in sequel, former two for augmentation and the last one eases training, shown in Figure 2:
1. **D**ata 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. **D**ata diversification. Due to the limited amount of distilled samples, we design various methods to increase the data variety and scale, and obtain the diversified data $\mathcal{D}^{div} = \{(\widetilde{p}, \widetilde{h}, \widetilde{r})\}$;
3. **D**ata curriculum. We combine $\mathcal{D}^{dis}$ and $\mathcal{D}^{div}$ as the augmented dataset $\mathcal{D}^a$. A curriculum strategy is defined to train the model with the easier distilled samples in $\mathcal{D}^a$ first and then the original ones in $\mathcal{D}$.

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 sentences. As shown in Figure 1, most responses only correspond to one persona sentence. The remaining persona information is mostly redundant, and may confuse the model to attend on useful persona information. Similarly, we notice that models tend to attend more on the last few utterances of H rather than the historical ones. We find that by using a Transformer encoder-decoder model, the attention weights of the last Transformer layer on the last utterance is 45% higher than the average on the other utterances. See Appendix C.1 for the experiment and results. This observation is also consistent with previous studies on multi-turn context understanding (Khandelwal et al., 2018; Sankar et al., 2019).

A few previous works have demonstrated that attention-based models will be distracted by noisy attended information and accurate attention supervisions can be very beneficial (Liu et al., 2016; Hsu et al., 2018). Inspired by them, we mimic a "hard" attention supervision between the response and useful persona/dialogue history by directly removing redundant tokens in the attended sequences. Therefore, different from previous work that modify the model to inject attention supervisions, our method only manipulates data.

Persona distillation We aim to determine which persona sentence the current response is consistent with, and thus remove the remaining non-consistent ones. To do so, we associate each persona sentence p_k with the target response R , and determine the consistency between each p_k and R. Following

previous work (Welleck et al., 2019), we cast it as a natural language inference (NLI) problem. If R entails p_k , it is considered to be consistent with p_k, otherwise irrelevant to p_k. A trained RoBERTa (Liu et al., 2019) model is used here as the NLI model, with an accuracy of 90.8% on the DialogueNLI dev set provided in Welleck et al. (2019). Details are provided in Appendix A.1.

Dialogue history distillation We can adopt a trained attention-based model to determine useful context sentences. For simplicity, we could also keep only the most useful last utterance H_M in a distilled sample (as suggested by our preliminary experiments discussed in the beginning of this section). In our experiments in Sec. 4, we find that using the last utterance is enough for our method to work well.

A distilled sample $(\widetilde{P}, \widetilde{H}, \widetilde{R})$ is ready to be constructed now. Here, $\widetilde{P}$ and $\widetilde{H}$ both contain only one sentence. $\widetilde{P}$ is any p_k that entails R, and $\widetilde{H}$ is the last utterance in the dialogue history, and $\widetilde{R} = R$. Such samples form the distilled dataset $\mathcal{D}^{dis}$. 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 sentences.

3.2 Data Diversification

Distilled samples should ease model training as their responses are highly dependent on their $\widetilde{P}$ and $\widetilde{H}$. However, samples in $\mathcal{D}^{dis}$ are limited in terms of both **scale** (around 40% of the original data) and **diversity** (about 4.5k unique persona sentences). Hence, it is necessary to augment $\mathcal{D}^{dis}$. Thanks to the assured relationship between $\widetilde{P}$/$\widetilde{H}$ and R, we can devise possible methods to diversify distilled samples with more semantically various samples. Our data diversification operation contains the following three parts along with quality filtering, as shown in Figure 2.

Persona editing We aim to obtain new persona sentences to improve the data scale, and more importantly the persona diversity. Hence, we here consider both token-level and phrase-level editing methods given a persona sentence $\widetilde{P}$:

• Token-level editing: we randomly mask a predefined ratio of tokens in $\widetilde{P}$, then use a pretrained BERT (Devlin et al., 2019) model to make predictions on the masked positions one by one.

• Phrase-level editing: we remove the last few tokens in $\widetilde{P}$ with the removal length determined by a random ratio, and utilize a pretrained GPT2 (Radford et al., 2019) to rewrite the removal part.

Multiple edited persona sentences can be obtained from one certain $\widetilde{P}$. Here, we finetune pretrained models using all persona sentences for a trade-off between semantic diversity and domain similarity. To ensure a satisfactory fluency and novelty of an edited persona $\widetilde{p}$, we rate it via a scoring function:

$$f = \alpha \cdot \text{PPL}(\widetilde{p}) + (1 - \alpha) \cdot \text{BS}_f(\widetilde{p}, \widetilde{P}). \quad (1)$$

Here, PPL calculates the normalized perplexity via a GPT2 model to measure its fluency, and the rescaled F1 value of BERTScore (BS_f) (Zhang et al., 2019) is employed to evaluate the semantic similarity between two sentences. Lower values for both functions are preferred, indicating 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 in Eq. 1, and select the top N_p ones.

Response aligning Since the semantic meaning of an edited persona sentence obtained above could change, the original response may not be consistent with it. Therefore, we need to get a new aligned response to maintain the persona consistency. Two approaches are utilized to obtain an aligned response $\widetilde{r}$ given an edited persona sentence $\widetilde{p}$ and the corresponding distilled history utterance $\widetilde{H}$:

• Token-level editing: We observe that some overlapped tokens can be found between $\widetilde{P}$ and $\widetilde{R}$. If an overlapped token w has been changed to a new token w' in the edited persona $\widetilde{p}$, we directly replace w in $\widetilde{R}$ with w' in the same positions, resulting in an aligned response $\widetilde{r}$. An illustration figure can be found in Appendix A.2.

• Model predicting: If no overlapped token can be found, token-level editing will not be applicable. Then we employ a GPT2-based encoder-decoder model (Cao et al., 2020) finetuned on the distilled data $\mathcal{D}^{dis}$ to predict responses with the given $\widetilde{p}$ and a dialogue history utterance $\widetilde{H}$.

Dialogue history augmentation To further scale up the size of distilled samples, we also manipulate the dialogue history $\widetilde{H}$. Since the diversity scarcity issue is not severe in $\widetilde{H}$, we use a popular sentence-level data augmentation method, back translation (BT) (Sennrich et al., 2016), to obtain variants of dialogue utterances. We could consider the semantics of the variants are identical. Distilled history utterance $\widetilde{H}$ is translated into an intermediate language, then back into the source language using a couple of existing translation models. The original

dialogue history and its N_h variants compose the augmented dialogue history set $\{\widetilde{h}\}$.

Combining the above three parts together, we now obtain new samples $\{(\widetilde{p}, \widetilde{h}, \widetilde{r})\}$. We evaluate them with respect to fluency, persona consistency and history coherence:

$$s = \beta \cdot \text{PPL}(\widetilde{r}) + \gamma \cdot \text{NLI}(\widetilde{p}, \widetilde{r}) + (1 - \beta - \gamma)\text{NLI}_c(\widetilde{h}, \widetilde{r}), \quad (2)$$

where NLI measures the entailment between a persona sentence and the response by the same NLI model in Sec 3.1, and NLI_c evaluates the entailment between a dialogue history utterance and the response 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 the diversified data set $\mathcal{D}^{div}$. The whole augmented training dataset is the union of $\mathcal{D}^{dis}$ and $\mathcal{D}^{div}$. 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 with multiple persona sentences and dialogue history utterances as the original data. Therefore, a model trained using $\mathcal{D}^a$ only is not proper. We should use both $\mathcal{D}^a$ and $\mathcal{D}$. Unlike previous studies that treat the original and augmented data equally and mix them directly, we design a curriculum strategy. Considering the different training difficulty of data in $\mathcal{D}^a$ and $\mathcal{D}$, we treat $\mathcal{D}^a$ as an easy curriculum while the original dataset $\mathcal{D}$ as a hard curriculum. The model is trained on such data curriculum successively until convergence.

4 Experiments

To validate the effectiveness of our proposed model-agnostic data manipulation method, we first experiment on two strong persona-based dialogue generation models (Transformer encoder-decoder and GPT2) on the benchmark PersonaChat (Zhang et al., 2018a) dataset. Next we conduct a series of analysis to examine the usefulness of different data manipulation operations in our method. [1]

4.1 Experimental Setup

Dataset The PersonaChat (Zhang et al., 2018a) data is widely used in this field (Song et al., 2019, 2020; Wolf et al., 2019; Golovanov et al., 2019).

[1]Our code will be available after publication.

	$\mathcal{D}$	$\mathcal{D}^{dis}$	$\mathcal{D}^{div}$	$\mathcal{D}^a$	$\mathcal{D}+\mathcal{D}^a$
#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.

Each sample has a dialogue history H with no more than 15 utterances ($M \leq 15$) and a persona P with between 4 and 6 sentences ($4 \leq L \leq 6$). Numbers of samples, unique persona sentences, and tokens in each stage of our method are listed in Table 1.

Base models Two dialogue model architectures are considered:

- TRANSFORMER (Vaswani et al., 2017): an encoder-decoder architecture using Transformer as the backbone with pointer generator (See et al., 2017) integrated;
- GPT2: one of the most powerful pretrained models on this task (Wolf et al., 2019; Golovanov et al., 2019; Cao et al., 2020).

TRANSFORMER is trained from scratch, and GPT2 is finetuned. For both models, we construct training data by concatenating persona and dialogue history 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 two base models with our method $\mathbf{D}^3$ and other data manipulation approaches for comparison:

- BACK TRANSLATION (BT) (Sennrich et al., 2016): we perform BT on all sentences in a training sample, including the persona sentences and dialogue utterances, and train the model with the augmented and original data jointly;
- CVAE (Li et al., 2019): a CVAE-based generation model is trained on the original data and then used to generate new responses via sampling with different latent codes. Since it can only handle pair-wise data, we concatenate all input sentences as a single input sequence in this method;
- ENTROPY FILTER (FILTER) (Csáky et al., 2019): it removes generic responses according to the entropy, which is calculated using the dialogue history and the response without using the persona.

The detailed configurations of each method are given in Appendix B.

Automatic metrics We adopt multiple widely used metrics to measure the response quality, including Perplexity (PPL), BLEU (Papineni

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**
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**

Table 2: Results of all compared data manipulation methods on two base models. BLEU and Dist-n are in %. Best results are in bold, and second best are underlined. Shaded numbers indicate our D^3 is significantly better than this method on human evaluation, C-score and BS_f, where $p < 0.05$ in T-test.

et al., 2002), NIST-4 (Doddington, 2002) and BERTScore (Zhang et al., 2019). We use the same BS_f in Eq. 1 for BERTScore. To evaluate the response diversity, 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, where we follow the default setting and use the output of an NLI model trained on the DialogueNLI dataset (Welleck et al., 2019) to indicate the consistency between a response and persona sentences.

Human evaluation We randomly selected 200 samples from the test set for human evaluations. Five professional annotators from a third-party company were asked to rate the responses from three aspects: 1) Fluency (Flu.); 2) Coherence (Coh.) with the dialogue history, 3) Persona consistency (Pcon.). The scores for the first two aspects have three scales, in which 1/2/3 indicates unacceptable/moderate/satisfactory respectively. The last one is binary, where 1 means the response is consistent with at least one persona sentence in the sample and 0 otherwise. The agreement rate from raters is 97.5%, 89.5%, 100% @3 (at least 3 of them reach an agreement) in the these aspects, indicating the validity of scores. The instruction of human evaluation is given in Appendix B.

4.2 Results

Table 2 reports the results on two based models trained with the use of various compared data manipulation methods. T-test is conducted between our D^3 and other compared methods on each base model for metrics including BS_f, C-score and three human evaluation metrics. Other automatic metrics have similar results or are not applicable such as Distinct-n. Details of the significant tests are given in Appendix C.2.

On TRANSFORMER, all methods achieve improvements on most metrics compared with training with the original dataset. Our method yields the best performance except for Ent-1. On GPT2, many methods fail to improve the various metrics consistently. For example, on the persona consistency (Pcon.), only ENTROPY FILTER and our method can get higher scores than training with the original dataset. The reason is that the data scarcity issue is less severe with a pretrained model, and it is more important to address the data diversity issue. In our method, the augmented distilled samples are encouraged to have different semantics with the original ones and improve the data diversity, and thus continue to get improvements on the strong pretrained GPT2.

4.3 More Analysis

We further analyze the contributions made by different data manipulation operations in our method by answering the following three questions:

1. whether there is a need to construct simple data $\mathcal{D}^{dis}$ as in data distillation;
2. whether data diversification can effectively obtain diverse distilled data;
3. whether the curriculum strategy better exploits the augmented data and helps model training.

We use results on TRANSFORMER here for discussion in the following part. Results on GPT2 are provided in Appendix C.3.

Analysis of data distillation 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

	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
TRANS-$\mathbf{D}^3$*	37.67	3.259	1.185	0.1554	4.197	6.095	7.232	1.794	7.835	14.27	0.439
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 distillation	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: Automatic evaluation results with variant in data distillation (middle), and diversification (bottom), compared with our full method (top) on TRANSFORMER. $\mathbf{D}^3$* means using an NLI model trained under a few-show setting (200 labelled samples) in the data distillation.

our $\mathbf{D}^3$ are considered: 1) *w/o diversification*: only using distilled data $\mathcal{D}^{dis}$ in the easy curriculum; 2) *w/o distillation*: based on 1), we recover samples in $\mathcal{D}^{dis}$ into their original format, which means all their persona sentences and history utterances are included; 3) *only distillation*: only $\mathcal{D}^{dis}$ is used in training without using the original data in $\mathcal{D}$.

Results of these variants are shown in the middle of Table 3. Obviously, removing data diversification decreases the performance in all aspects as the model has less training data. If we further remove data distillation and use the same amount of data in their original formats, the model performs even worse, especially on the C-score. This validates the effectiveness of data distillation in our method. However, it is not proper to completely rely on distilled data. From the results of only using distilled data in training, our method improves the C-score, yet significantly degenerates in other aspects. The reason is that the relationship between persona/dialogue history and the response has changed from the original data to their distilled ones. Thus a model trained with distilled data should serve as a warm start to learn the original data, but not to replace the original data.

We also test the robustness of our data distillation method by using an NLI model trained in a few-shot setting (200 samples). Results are included in Table 3 as $\mathbf{D}^3$*. It is slightly worse than our method with sufficient NLI training data, but still superior to most compared methods. Note that the response diversity metrics nearly remain unchanged. This means that our data diversification methods are still effective when starting from noisy distilled samples. It also shows that our method can be useful when only limited in-domain NLI labeled data are available for data distillation.

Analysis of data diversification Table 1 shows

	Novelty-1, 2, 3, 4			
sample	30.89	47.07	53.81	59.64
persona	40.26	62.17	70.47	77.81

Table 4: Novelty metrics of the diversified data compared to distilled data in sample and persona level.

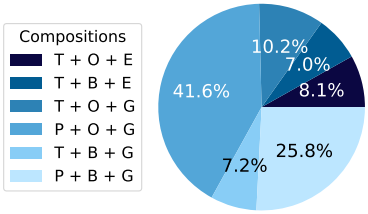


Figure 3: The compositions of diversified data. T/P: token/phrase-level editing to get edited personas, O/B: original/BT-augmented dialogue history, E/M: token editing/model predicting to get aligned responses.

that the diversified data contain many new persona sentences as well as tokens. Besides, we compute the Novelty metrics (Wang and Wan, 2018; Zhang et al., 2020) of diversified samples in $\mathcal{D}^{div}$. It takes the original distilled samples in $\mathcal{D}^{dis}$ as references, and uses the Jaccard similarity function to measure the proportion of n-grams ($n = 1, 2, 3, 4$) in $\mathcal{D}^{div}$ but not in $\mathcal{D}^{dis}$. A higher value means more "novel" content. Note that we particularly prefer more novel personas, while not encouraging more novel dialogue histories. Thus, the Novelty scores on the overall samples which include dialogue histories, personas and responses, are lower than those on the personas.

To further examine how each part of data diversification works, we conduct the following ablation studies: 1) *w/o persona editing*: no persona sentence will be edited; 2) *w/o history augmentation*: only original dialogue history is used; 3) *w/o response filtering*: all constructed samples are directly used without using Eq. 2. Results in the bottom of Table 3 show that all these designs con-

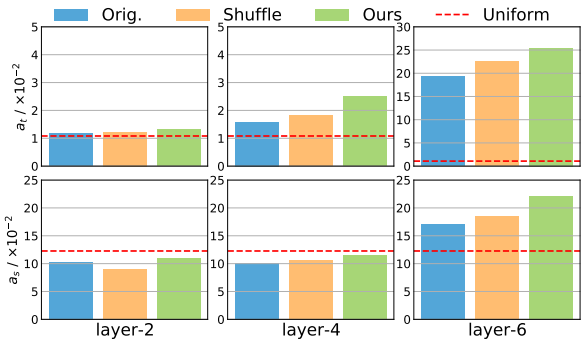


Figure 4: Average consistent attention weights in different decoder layers of TRANSFORMER trained with (i) original dataset (Orig.), (ii) shuffled data in $\mathcal{D}$ and $\mathcal{D}^a$ (Shuffle), and (3) our data curriculum. Uniform: uniform attention values on all positions. Top: token-level a_t; bottom: sentence-level a_s.

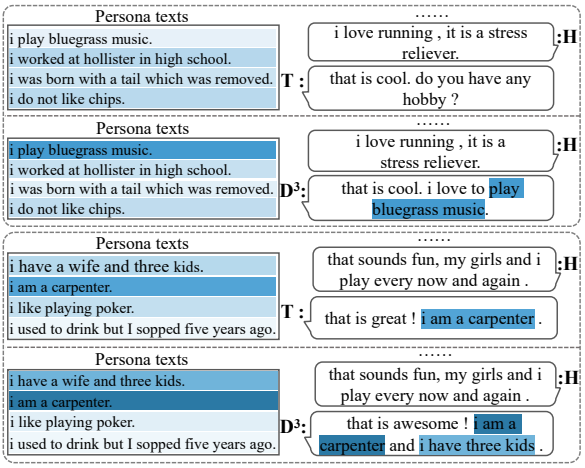


Figure 5: Sample responses and visualized model attentions on personas. **T**:TRANS, **D**^{3}:TRANS-**D**^{3}.

tribute to the performance of the whole method. Among them, response filtering is the most important as it ensures the quality of augmented samples.

We also investigate the proportions of diversified samples coming from various source combinations. Results are shown in Figure 3, which shows that more than 80% diversified samples have their responses obtained via model predicting, as token editing sets a strict condition that overlapped tokens must exist. Phrase-level editing also contributes to more high-quality personas with satisfactory fluency and semantic novelty.

Analysis of data curriculum We first compare other data curriculum variants to show the usefulness of training with the designed data curriculum. Details can be found in Appendix C.4. We further quantify the effect of curriculum training on models using the attention from the response on the persona sentences. We define two metrics, token-level/sentence-level consistent attention weight (a_t and a_s), to measure how the attention contributes to reflecting the proper personas. Recall that we concatenate the persona sentences and history utterances as a single model input. We record the token positions of the entailed persona sentences in the input sequence, denoted as $\mathcal{S}$. Then for each index $s \in \mathcal{S}$, if its corresponding token occurs in the response, we put this index pair into a set $\mathcal{T} = \{(s, l)\}$, where l is the token position in the response sequence 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 the generated response. A higher a_t/a_s indicates that the model poses more attention on proper persona tokens. Averaged results on all samples from the dev set are shown in Figure 4. 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 5. Here **H** indicates dialogue history, a persona sentence shaded in a darker color denotes that it has a higher attention weight posed by the model. Obviously, our method offers a model with the capability to pose more attention on the accurate persona information. More cases can be found in Appendix C.6.

5 Conclusion

Our work targets the challenging personal-based dialogue generation task. Unlike previous work that designs a new dialogue model to improve the generation performance, we analyze the data issues to train current models. On one hand, the data scale and diversity are expensive to increase by data collection. On the other hand, current data are difficult to learn with. Based on these understanding, we propose a model-agnostic data manipulation method for this task packed with any model. 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. Experimental results showed that our method effectively improves the performance of two strong dialogue models, i.e. Transformer encoder-decoder and GPT2.

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.

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.

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*, 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*, 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 co-occurrence 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*.

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*.

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*, 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*, 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*, 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*, pages 6698–6705.

Zhaojiang Lin, Zihan Liu, Genta Indra Winata, Samuel Cahyawijaya, Andrea Madotto, Yejin Bang, Etsuko Ishii, and Pascale Fung. 2020. Xpersona: Evaluating multilingual personalized chatbot. *arXiv preprint arXiv:2003.07568*.

Zhaojiang Lin, Andrea Madotto, Yejin Bang, and Pascale Fung. 2021. The adapter-bot: All-in-one controllable conversational model. In *Proceedings*

of the AAAI Conference on Artificial Intelligence, pages 16081–16083.

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*, 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*, (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*.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, et al. 2021. Recipes for building an open-domain chatbot. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*, pages 300–325.

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*, 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*, pages 86–96.

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 1577–1586.

Haoyu Song, Yan Wang, Kaiyan Zhang, Wei-Nan Zhang, and Ting Liu. 2021. BoB: BERT over BERT for training persona-based dialogue models from limited personalized data. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 167–177.

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*, 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. In *Advances in Neural Information Processing Systems*, pages 1810–1820.

A Implementation Details of 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 is utilized. To make the model better fit the domain of PersonaChat, we finetune the model on the DialogueNLI dataset (Welleck et al., 2019) which is a part of the original PersonaChat. We set the batch size as 32 and finetune the model for 5 epochs using a learning rate 1e-5. We obtain a model RoBERTa_{nli} achieving 90.8% accuracy on the dev set. This model will also be responsible for calculating the entailment probability NLI in response filtering and **C**-score in the experiments. A threshold $\tau = 0.99$ is used in this model for predicting the NLI labels. For the few-shot setting $\mathbf{D}^3*$ in Sec 4.3, we randomly sample 200 samples from the training set to train the above NLI model using learning a rate 2e-5, and obtain a model achieving 79.3% on the dev set.

A.2 Details of diversification

The BERT-based-uncased model[3] and GPT2-base[4] are involved as the pretrained models in this stage. To ensure that the pretrained models can make predictions that better fit current data domain while also have enough capabilities of generation diversity, we perform the following finetuning: 1) finetune BERT and GPT2 on the persona sentences for 100 steps with a batch size 32 and a learning rate 1e-4, obtaining BERT_{per} and GPT2_{per}; 2) finetune GPT2 on responses for 200 steps with a batch size 32 and a learning rate 1e-4, and obtain GPT2_{res}.

Persona editing BERT_{per} and GPT2_{per} will be used for token-level editing and phrase-level editing respectively. Each will generate 10 unique new persona sentences from one original persona sentence via sampling according to the multinomial distribution. At the token level, we only mask the most informative tokens which can be decided by the POS tags given by SpaCy[5] as it is meaningless to mask some words such as prepositions "to" and "in". The target POS tags are listed in Table 5. We set the token-level mask ratio as 0.8. At phrase level, the mask ratio is randomly sampled between $[0.3, 0.6]$. We also restrict that at least 2 tokens are masked and the maximum length of generated text pieces from GPT2_{per} does not exceed 30% of the original length to preserve the sentence similarity.

[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.

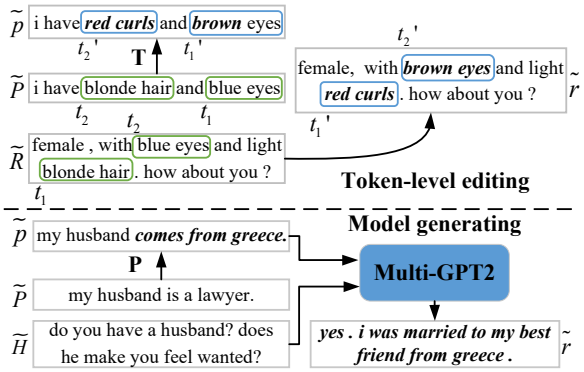


Figure 6: Aligning responses for new personas via token-level editing or model generating. **T/P**: edit persona in token/phrase level.(t_1 and t_2 are overlapped tokens, t_1' and t_2' are corresponding new edited and aligned tokens.)

We use $\alpha = 0.4$ in Eq. 1, where PPL is given by GPT2_{per} normalized by a constant 50 (which is about the highest PPL value given by the GPT2 model on current corpus). For BERTScore, the F1 value is used as BS_f while other configurations follow the recommendation for English in Zhang et al. (2019)[6]. N_p is set as 5.

Response aligning For token-level editing, we also restrict the POS tags of overlapped tokens according to Table 5. For model predicting, 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 see that this model shows high n-gram accuracy and persona consistency, thus should be effective. Figure 6 demonstrates the two approaches.

Dialogue history augmentation We use the *transformer_wmt_en_de* Transformer model in Fairseq[7] as the translation model. It is trained on the WMT14 EN-FR dataset with 40.5M samples and default configurations. During inference, we use beam search with its size 5 for both en-fr and fr-en translation, resulting in 25 new utterances for each original one. For a large divergence, we select $N_p = 1$ new utterance with the lowest BLEU score when taking the original one as the reference.

Quality filtering We use GPT2_{res} normalized by a constant 50 to get the PPL of responses. Here,

[6]https://github.com/Tiiiger/bert_score
[7]https://github.com/pytorch/fairseq

	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.

	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.

Param	Candidate values
τ	0.9, 0.95, 0.99
α	0.4, 0.5, 0.6
β	0.2, 0.3
γ	0.4, 0.5, 0.6

Table 8: The candidate values for hyper-parameters during grid searching.

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

Table 9: The training sample number used in each method.

we finetune another RoBERTa-Large-MNLI model on the InferConvAI dataset[8] which achieves 88.7% accuracy on its dev set. The entailment probability given by this model is regarded as NLI_c. We set $\beta = 0.2$, $\gamma = 0.6$ in Eq. 2.

We compare the fluency and coherence of responses with the GPT2-based PPL and NLI model-based score from the training set, which are shown in Table 7. In addition, we also evaluate the GPT2-PPL's for edited and original persona sentences, which are 6.427 vs. 10.426.

B Details of Experiment

Base model For TRANSFORMER, 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, with the hidden size 300 and 4 heads. During training, a cross-entropy loss is used along with Label Smoothing with the ratio 0.1. For GPT2, we use the base pretrained model with 12 layers and 768-dim hidden state. It will be trained using the average of a cross-entropy loss on generating and a classification loss between true response and one randomly sampled negative response. Beam search with the beam 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 7. Here <bos>, <eos>, <talker1>, and <talker2> are special symbols to distinguish different parts of input or response.

Model training We use a learning rate 2e-4 for TRANSFORMER and 6.25e-5 for GPT2, which is a common setting in former similar works. And the training 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 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. Models with the minimum loss at each curriculum are remained as the best. 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 32GB NVIDIA V100 GPUs. Each epoch takes about 10 min for Transformer and 25min for GPT2.

Hyper-parameters All hyper-parameters are determined using a coarse grid search to ensure satisfactory performance, including τ in data distillation, α in Eq. 1, β, γ in Eq. 2. The candidate values of these hyper-parameters are given in Table 8, which are determined empirically to reduce the searching cost. The search target we want to maximize is the normalized average of all automatic metrics listed in Table 2 when inferencing on the test set, except PPL. Note that we only take TRANSFORMER as the base model for search, each time of search takes about 0.7 GPU day. GPT2 model follows the same setting as TRANSFORMER. We found that τ plays a more important role in our method who determine the quality of distilled samples, while other parameters have fewer impacts on our method.

Baselines We apply the same translation models as the ones used in A.2 for the BT (Sennrich et al., 2016) baseline and augment each sample with a

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

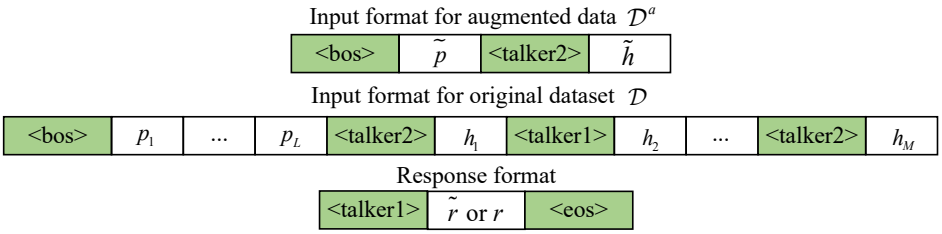


Figure 7: The sequence format of an input and an output for both TRANSFORMER and GPT2 models.

new sample from it. For CVAE (Li et al., 2019) method, we use its default setting to train the model on PersonaChat dataset without using the personas. A new sample is generated for each input in the original dataset. In Entropy-filter (Csáky et al., 2019), we set the threshold as 1.1 and using both source and target sequences for filtering. Only samples that survived after filtering are used in training. The total numbers of training samples of all methods are listed in Table 9. Note that 0all models are trained until the loss does not decrease for the same N epochs for a fair comparison.

Metrics We use the same BS_f and RoBERTa_{nli} obtained before to calculate the BERTScore and C-score metrics. The annotation instruction for human annotators is provided in Table 13 and Table 14.

C Additional Experimental Results

C.1 Attention on dialogue history

To investigate how models pose attention on each part of dialogue history, especially the last utterance, we calculate the attention weights from different decoder layers on the last utterance or the other dialogue history utterances. TRANSFORMER model is used here, which is trained with the original training data without any augmentation. When testing on the dev set of PersonaChat dataset, the average token-level attention weight on the last utterance in the dialogue history is significantly higher than that on all other utterances, as shown in Figure 8. Thus, our history distillation can ease model learning for such knowledge by removing former utterances.

C.2 Statistical results of Table 2

We conduct T-test between the experimental results of our method $\mathbf{D}^3$) and other baselines for each base model to verify the significant results between these methods. Here, all human evaluation results (Fluency, Coherence, Persona-consistency), and some applicable automatic metrics (C-score, BS_f) are included. We can find that nearly all results

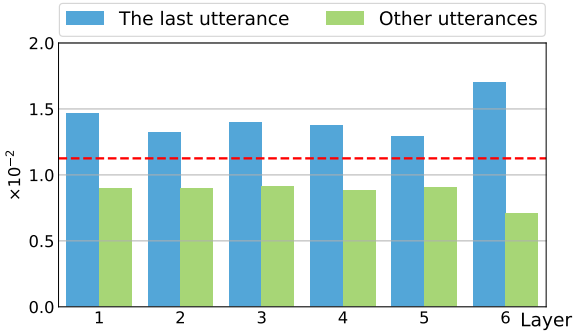


Figure 8: The average token-level attention weights from different decoder layers in TRANSFORMER on the last utterance or other part of dialogue history. Red line: the baseline values when all attention distributes uniformly among all tokens.

from baselines satisfy the null hypothesis (results are significantly different from $\mathbf{D}^3$) given $p < 0.05$ or even a smaller value using TRANSFORMER as the base model. Such a significance of difference from baselines decreases on GPT2 model except CVAE, which again shows that data manipulation methods have fewer impacts when packed with a pretrained model.

C.3 Analysis of ablations on GPT2

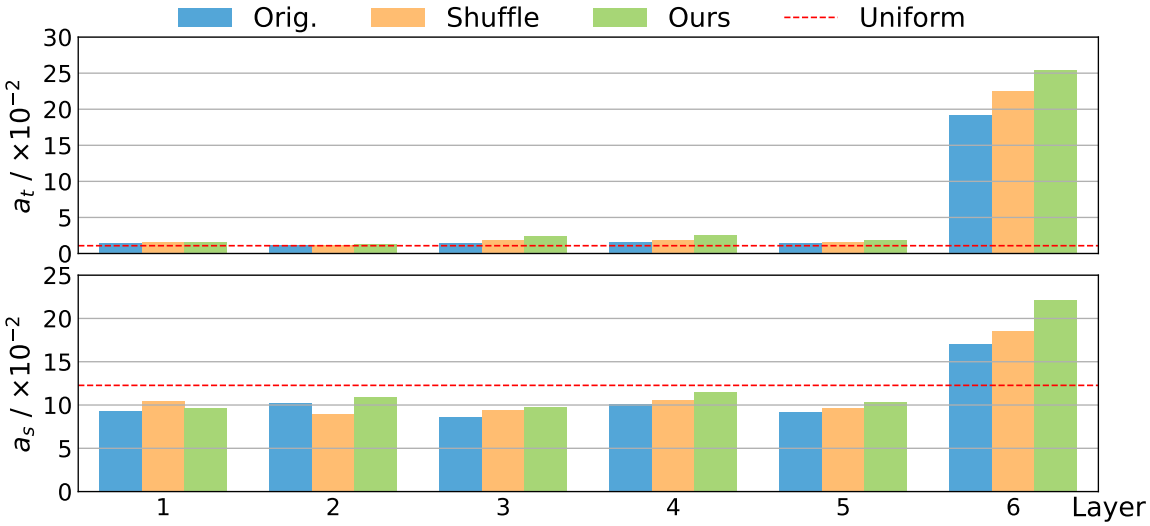
We also provide the extensive results of ablation experiments on GPT2 which is similar to the ones given in Section 4.4 on TRANSFORMER. Table 10 shows the results. We can find the influence of data diversification, as well as our distillation have less impacts on GPT2 compared to TRANSFORMER. The reason is that GPT2 is a very powerful pretrained model and thus is less vulnerable to the different numbers of data samples. Using an NLI model trained under few-shot setting is still comparable to the one using an NLI model with sufficient samples.

C.4 Detailed results of curriculum analysis

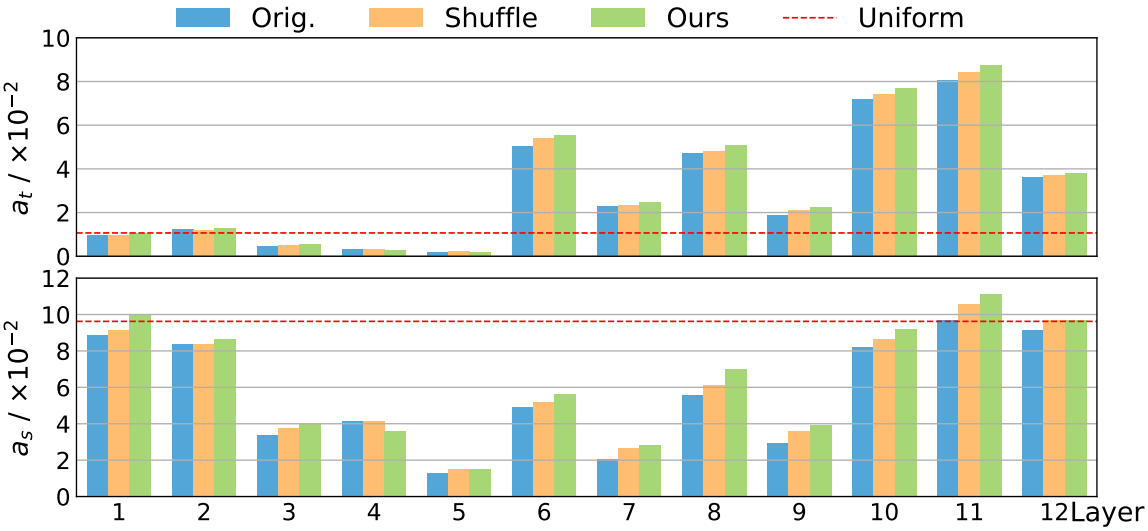
We test several variants of our data curriculum: 1) Original: only the original dataset $\mathcal{D}$ (the hard curriculum) is used, it is equal to the base model;

	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
GPT2-$\mathbf{D}^3*$	15.77	4.082	1.388	0.1809	4.611	6.408	7.312	2.209	9.657	17.91	0.536
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 10: Automatic evaluation results with variant settings in distillation variants (middle), and data diversification ablations (lower), compared with the original $\mathbf{D}^3$(top) on GPT2. $\mathbf{D}^3*$ means using an NLI model trained under a few-show setting (200 labelled samples) in the data distillation.



(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 9: Consistent attention weights on TRANSFORMER and GPT2. Orig.:training the model using the original training data $\mathcal{D}$; Shuffle: training the model using the shuffling data of $\mathcal{D}$ and $\mathcal{D}^a$; Ours: training the model using our curriculum strategy; Uniform.: the attention value distributed on all positions uniformly, which is a baseline.

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

	PPL	BLEU	NIST-4	BS_f	Ent-1	Ent-2	Ent-3	Dis-1	Dis-2	Dis-3	**C**
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
Original	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-$\mathbf{D}^3$	15.69	4.184	1.429	0.1835	4.614	6.426	7.321	2.179	9.458	17.72	0.557
Orignal	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 11: Performance comparison between different curriculum strategies on two base models. Top: Transformer, bottom: GPT2.

	PPL	BLEU	NIST-4	BS_f	Ent-1	Ent-2	Ent-3	Dis-1	Dis-2	Dis-3	**C**
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
TRANS-$\mathbf{D}^3$(200%)	37.49	3.367	1.199	0.1570	4.271	6.235	7.343	1.821	7.997	14.51	0.493
TRANS-$\mathbf{D}^3$(50%)	37.75	3.269	1.167	0.1551	4.132	6.085	7.003	1.743	7.658	14.10	0.468

Table 12: Performance comparison between original $\mathbf{D}^3$ and variants when using diversified dataset $\mathcal{D}^{div}$ with about 200% or 50% size of distilled dataset $\mathcal{D}^{dis}$.

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 11. 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 Orig. as it includes more simplified distilled training data, which may benefit the training. However, this shuffling 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 of Transformer and GPT2 trained via Orig., Shuffle or our $\mathbf{D}^3$ method, which are shown in Figure 9. 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.5 The influence of diversified sample numbers

Since we can simply control the threshold for s in Eq. 2 to determine how many diversified samples are generated for $\mathcal{D}^{div}$. How this quantity affect the performance of $\mathbf{D}^3$? We carry out experiments to use different $\mathcal{D}^{div}$ whose size is about 50% of $\mathcal{D}^{dis}$ or 200% of $\mathcal{D}^{dis}$ on TRANSFORMER, compared to the original method where $\mathcal{D}^{div}$ is nearly the same size as $\mathcal{D}^{dis}$. The results in terms of automatic metrics are shown in Table 12. It can be found that further extending the data scale will result in a very slight promotion but a longer training time, while squeeze the diversified dataset size has a more obvious effect on the performance. Nevertheless, using $\mathcal{D}^{div}$ with a similar size as $\mathcal{D}^{dis}$ is a good trade-off between resource cost and performance, while ensure a fair comparison between former methods.

C.6 More case studies

Except for the cases provided in Sec 4.3, we provide additional cases including the responses given by GPT2. They are shown in Figure 10, including visualized attention weights posed by different models on their persona sentences. 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 parts, thus the generated responses have better persona consistency.

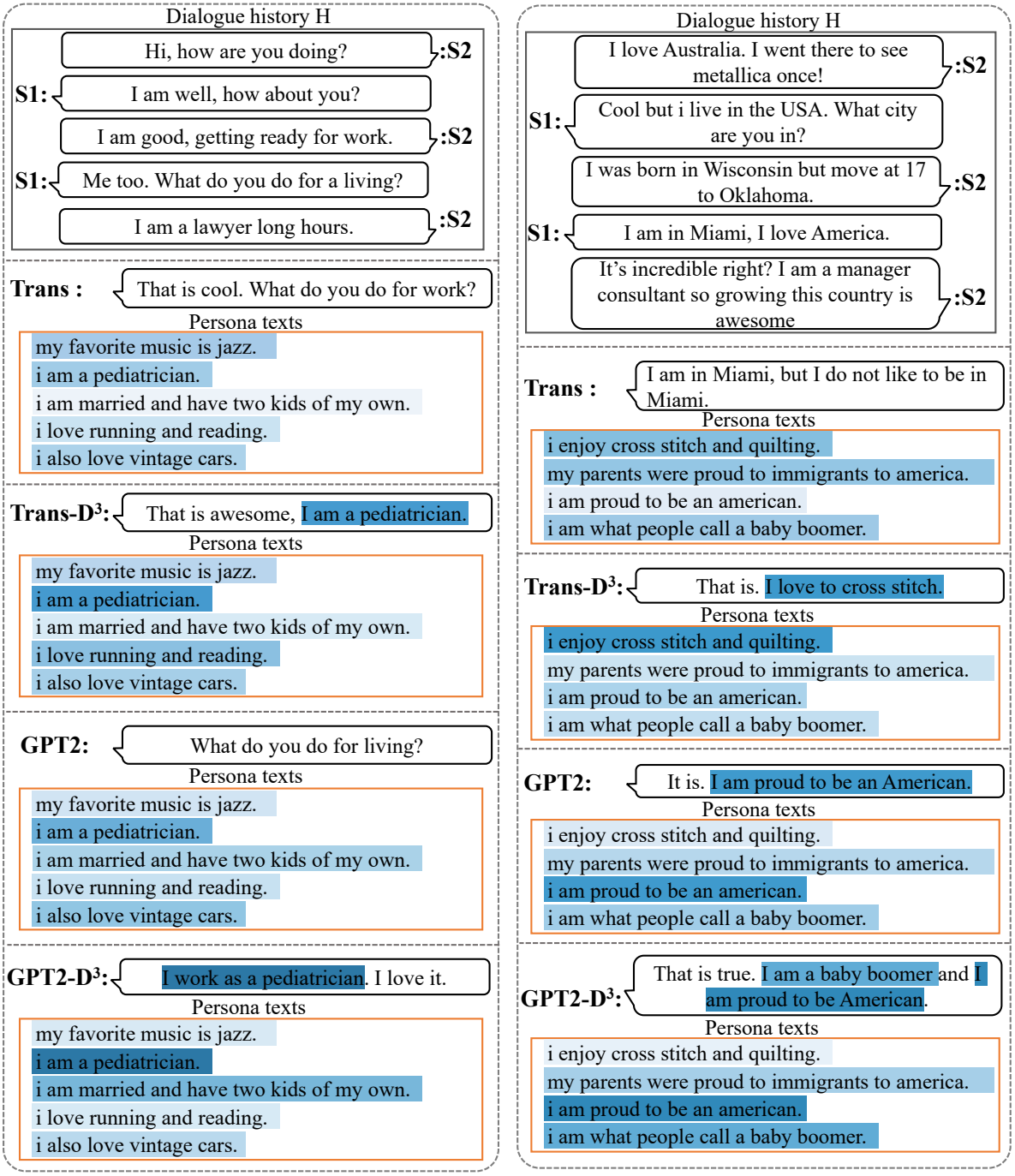


Figure 10: Additional responses cases and visualization by Transformers(Trans) and GPT2 without or with our $\mathbf{D}^3$ data augmentation method. Colors in each persona text indicate the attention weight paid by different models. A darker color means a higher attention weight is posed by the current model. Colored texts in the response denote the persona consistency.

Data description	You are supposed to be **Speaker S2**, you are required to evaluate the **quality of dialogue responses from S2** in the following 3 aspects, based on 1) **the persona information of S2**; and 2) **the dialogue history with Speaker S2**. Here, **the persona information of S2** mean the personality/characteristics of the speaker for the response need to be evaluated. The responses are expected to reflect the given persona for the speaker as possible, meanwhile, they should also be proper and coherent for the previous messages from Speaker S1. **Each serial number indicates one sample.** It contains persona information and corresponding dialogue history. The dialogue history contains several different responses (by different methods). Your need to **rating for every response** considering the persona information and dialogue history. **Rating contains the following 3 aspects.**

1. Fluency (1 $\sim$ 3. Your need not to consider the persona information and dialogue history, just the response itself.

Score	Description	Examples
1 (unsatisfied)	1) The text is totally broken, or contains severe grammar errors. 2) The text is very hard to understand	S1:i do not have any but charlie my puppy enjoys it S2:i am triplets triplets triplets triplets triplets (Cannot understand) S1:i am a college student . art major . S2:i love my spanish . is studying it has been studying ? (Totally not fluent)
2 (fair)	1) The text is basically fluent, contains grammar errors but do not affect understanding. 2) The response is short but fluent, without grammar error. 3) The text contains some repeated context. 4) The text is bassically fluent, but contains perverse content.	S1good , you have any hobbies ? S2i travel a lot (Fluent but too simple) S1what kind of dog is he ? S2he is a german shepard . he is a german shepard . (Fluent but contains repetitions)
3 (satisfied)	1) The text is long and informative, few grammar errors are acceptable. There may exist some non-fluent parts, but do not affect understanding. 2) The text is in medium length, fluent without grammar error.	S1:hello what are doing today ? S2:hello , i just got back from the gym . how are you ? S1:good , you have any hobbies ? S2:i used to be a painter , i still like to do that a lot . how about you ?

2. Dialogue coherence (1 $\sim$ 3. You need not to consider the fluency if there is no difficulty in understanding. Your need to consider both the repsonse and dialogue history.)

Score	Description	Examples
1 (unsatisfied)	The response is irrelevant to the dialogue history. E.g., it does not share the same topic or it is an irrelevant answer.	S1: how old are you ? i turned four on my birthday ! S2: awesome ! i love the insane clown posse love (Irrelevant answer)
2 (fair)	Very limit relevance exists between the response and history, or meets the following conditions: 1) The response is the same as the query. 2) The response is a kind of paraphrase of the query. 3) It is a general response that do not answer the query or contains very limited information,e.g., "i am sorry" 4) The response is a question without new information.	S1: yes i bet you can get hurt . my wife works and i stay at home S2: i wish i could do that (very limited relevance) S1: hi ! do you like turtles ? S2: yes i do , do you have any hobbies ? (a question without new information) S1:i would love to travel to italy . i love baking cookies . S2:i would love to visit italy sometime . (Praphrasing the query)
3 (satisfied)	1) The text is long and informative, few grammar errors are acceptable. There may exist some non-fluent parts, but do not affect understanding. 2) The text is in medium length, fluent without grammar error.	S1:hello what are doing today ? S2:hello , i just got back from the gym . how are you ? S1:good , you have any hobbies ? S2:i used to be a painter , i still like to do that a lot . how about you ?

Table 13: The instruction for annotators to make human evaluation for the generated responses (Part 1).

3. The consistency with given persona (0 or 1. Your need to consider both the persona sentences and the response.)		
Score	Description	Examples
0	The response totally does not reflect any given persona information.	Persona sentences: 1) i was born in south carolina. 2) hey there i am a professional singer. 3) i graduated from usc. 4) my name is joanna and i love watching horror films. S2: what is your favorite movie ? (totally irrelevant to persona) S2: I was born in Texas. So where is your home twon ? ("born in Texas" contradict the persona sentence "i was born in south carolina". And there is no other text can reflect the correct persona.)
1	The response can reflect one or several persona sentences directly or indirectly.	Persona sentences: 1) i read twenty books a year. 2) i'm a stunt double as my second job. 3) i only eat kosher. 4) i was raised in a single parent household. S2: nice . i love to read . (directly reflect the persona "i read twenty books a year.") S2: nice ! i am currently reading a horror novel . (Indirectly reflect the persona "i read twenty books a year.")

Table 14: The instruction for annotators to make human evaluation for the generated responses (Part 2).