Pretrained Language Models for Dialogue Generation with Multiple Input Sources

Yu Cao [1]*, **Wei Bi** [2], **Meng Fang** [3], **Dacheng Tao** [1]

[1] UBTECH Sydney AI Center, School of Computer Science, Faculty of Engineering,
The University of Sydney, Australia

[2] Tencent AI LAB, China, [3] Tencent Robotics X, China

ycao8647@uni.sydney.edu.au, victoriabi@tencent.com,
mfang@tencent.com, dacheng.tao@sydney.edu.au

Abstract

Large-scale pretrained language models have achieved outstanding performance on natural language understanding tasks. However, it is still under investigating how to apply them to dialogue generation tasks, especially those with responses conditioned on multiple sources. Previous work simply concatenates all input sources or averages information from different input sources. In this work, we study dialogue models with multiple input sources adapted from the pretrained language model GPT2. We explore various methods to fuse multiple separate attention information corresponding to different sources. Our experimental results show that proper fusion methods deliver higher relevance with dialogue history than simple fusion baselines.

1 Introduction

Large-scale pretrained language models (Devlin et al., 2019; Radford et al., 2018, 2019) have achieved outstanding performance on various natural language understanding tasks (Young et al., 2018; Liu et al., 2019). Researchers have then utilized them in dialogue generation tasks (Budzianowski and Vulić, 2019; Edunov et al., 2019; Zhang et al., 2019). Many of them simply concatenate the input dialogue history and the output response in finetuning, since the pretrained language model only accepts a single sequence as input. However, dialogue generation tasks may involve multiple input sources simultaneously. For example, in personalized or knowledge-grounded dialogue generation (Li et al., 2016; Zhang et al., 2018; Dinan et al., 2018), a response is generated conditioned on both dialogue history and an auxiliary user profile or knowledge article. Despite simple concatenation of all input sources, an important question arises on how we can better adapt a single-input pretrained language model to a multi-input dialogue generation task.

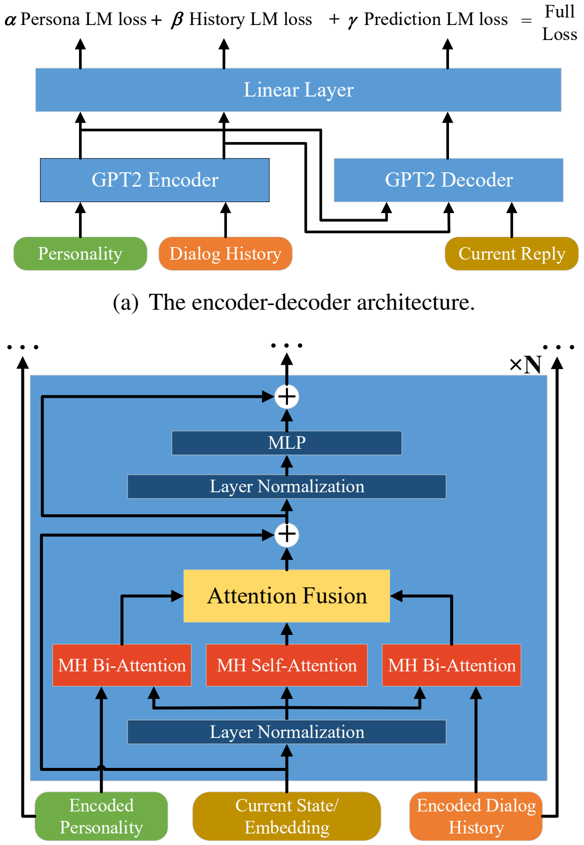
Some previous work forms an encoder-decoder architecture with both encoder and decoder duplicated from a pretrained language model (Golovanov et al., 2019; Zheng et al., 2019). Recently, BART (Lewis et al., 2019) even obtain a complete pretrained model under this architecture directly. Taking personalized dialogue generation (Zhang et al., 2018) as an example, we can treat persona information, dialogue history and previous generated tokens as three different input sources. The former two will be encoded firstly and then combined with the last one in the decoder. In Golovanov et al. 2019, the multi-head attention layer in the decoder is copied three times for each input source and mean pooling is used to average results from multiple attentions. This encoder-decoder adaptation is shown to outperform simple concatenation.

However, when dialogue history gets longer, this model tends to use less information of each dialogue history utterance to predict the next token. Zheng et al. 2019 add an extra weight predictor to combine multiple attention information, but they do not perform experiments using publicly released pretrained models, nor on public datasets, making their results not directly comparable to other work.

In this work, we build our dialogue model on the encoder-decoder architecture adapted from the pretrained language model GPT2 (Radford et al., 2019). Our main contribution is to empirically study the attention fusion methods for multiple information sources in each decoder layer. Three kinds of methods are explored in total. Our experimental results show performance improvements on both automatic and human evaluations by using proper attention fusion methods, compared to baselines using concatenation or mean pooling.

[*] This work was done during Yu Cao's internship in Tencent AI LAB, Shenzhen.

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 909–917
November 16 - 20, 2020.



(a) The encoder-decoder architecture.

(b) Details of each transformer block in decoder.

Figure 1: Architecture of our proposed model.

2 Model

2.1 The Encoder-Decoder Architecture

Following the former work (Golovanov et al., 2019), we use the personalized dialogue generation task on PersonaChat (Zhang et al., 2018) as an example in our study. The pretrained language model GPT2 and its parameters are duplicated to form an encoder-decoder architecture shown in Figure 1(a). We use GPT2 here due to its large-scale pre-training corpus than other models and strong performance in other generation tasks.

We have three separate inputs: personal profile, dialogue history, and current reply (or previously generated response during the inference stage). Embeddings of the former two, which contain embeddings of tokens, positions as well as token types, will be successively put into the encoder, which is a GPT2 model with no attention mask to fit the encoding procedure. The encoded representations, together with embeddings of current response tokens will then be used as the input of a modified GPT2 decoder. Each decoder block will attend the current state to the three sources using different attentions, then fuse their resulting information as input for the next layer.

Inspired by multi-task learning (Zhang and Yang, 2017), we further separate the original loss in language modeling into three parts corresponding to three input sources respectively. By applying the same linear prediction layer on the output of both encoder and decoder, three cross-entropy losses between predicted logits and corresponding truth sequences will be weighted by hyperparameters.

$$\mathcal{L} = \alpha \mathcal{L}_{persona} + \beta \mathcal{L}_{history} + \gamma \mathcal{L}_{pred} \quad (1)$$

with Adam optimizer (Kingma and Ba, 2014).

2.2 Block Details in Decoder

Recall that we have three input sources in the decoder, and thus some modifications are needed if the decoder structure is inherited from GPT2. Details of each modified decoder block are shown in Figure 1(b), in which the most apparent change is the additional two multi-head (MH) bidirectional attentions and the attention fusion module that fuses various attention outputs. The other parts remain the same as GPT2. In the following, we will first describe the MH Bi-attention. Attention fusion will be discussed in the next section.

The MH self-attention in Transformer (Vaswani et al., 2017) handles a single input only. In order to make it accept two input sources, we regard the current state $\mathbf{H}^c \in \mathbb{R}^{L^c \times d}$ from the previous layer (or embedding of reply in the first layer) as query and encoded state of auxiliary information $\mathbf{H}^a \in \mathbb{R}^{L^a \times d}$ as key and value in the attention. Here L^c and L^a are corresponding lengths for these input, and $\mathbf{H}^a$ can be encoded personality $\mathbf{H}^p$ or dialog history $\mathbf{H}^h$. The output of each single head in MH Bi-attention can be obtained via

$$\mathbf{A} = \text{softmax}\left(\frac{(\mathbf{H}^c\mathbf{W}^Q)(\mathbf{H}^a\mathbf{W}^K)^{\mathrm{T}}}{\sqrt{d}}\right)(\mathbf{H}^a\mathbf{W}^V), \quad (2)$$

where $\mathbf{W}^Q$, $\mathbf{W}^K$, $\mathbf{W}^V$ are learnable matrices. In our model, different attentions own separate parameters instead of sharing. This differs from the previous work (Golovanov et al., 2019) which reuses the self-attention for bi-attention. Besides, the original GPT2 is a single-directional model using a triangular matrix as the attention mask. Since the auxiliary information $\mathbf{H}^a$ is visible for the current reply at all time steps, no mask exists in MH bi-attention.

In total, three attention information $\mathbf{A}^c$, $\mathbf{A}^p$, and $\mathbf{A}^h$ are obtained by attending the current state to itself, personality, and history respectively, all in the same dimension $\mathbb{R}^{L^c \times d}$. They need to be fused into one matrix $\mathbf{A}^f \in \mathbb{R}^{L^c \times d}$ so as to proceed to subsequent decoding layers.

2.3 Attention Fusion

In this section, we discuss various methods to fuse the multiple attention information obtained above. The simplest approach is to average three sources in all dimensions (Golovanov et al., 2019), which treats all sources equally. However, in different dialogues, we may need to concentrate more on the dialogue history or the persona profile in order to generate proper responses. Here we introduce the following three kinds of methods to allow for more flexible information fusion from all input sources.

- **Static methods** fuse different information using an identical fusion function with no training parameter. Except the average pooling **(avg)** which is regarded as a very simple fusion baseline, we also include Maximum **(max)**, and Minimum **(min)** operation for every dimension among all sources.
- **Weighting methods** try to estimate the global optimal proportion of each source in a given domain by introducing extra learnable weights which are then fixed in inference. Such methods can be:

(i) source-level scalar weights **(sw)**, which means there are three trainable scalars w^c, w^p, w^h for each source in each layer and $\mathbf{A}^f = (w^c\mathbf{A}^c + w^p\mathbf{A}^p + w^h\mathbf{A}^h)/(w^c + w^p + w^h)$.

(ii) source-dimension level **(dw)**, in which weights are learnable vectors $\mathbf{w}^c, \mathbf{w}^p, \mathbf{w}^h \in \mathbb{R}^d$. For each row j of $\mathbf{A}^f$ and weight vectors $\mathbf{w}$, we perform the weighted combination via $\mathbf{A}^f_j = (w^c_j\mathbf{A}^c_j + w^p_j\mathbf{A}^p_j + w^h_j\mathbf{A}^h_j)/(w^c_j + w^p_j + w^h_j)$.

(iii) linear method (**linear**) in which a linear network is used to transform the concatenated attention $[\mathbf{A}^c; \mathbf{A}^p; \mathbf{A}^h]$ into $\mathbf{A}^f$. Different from above one, each dimension in the new feature space here contains information from all dimensions of all sources to realize a better interaction.

- **Attention-based method** fuses the information based on a trainable modified transformer attention (**att**). The attention fusion function changes according to multiple input information as follows

$$\mathbf{Z} = \mathrm{softmax}(\frac{\mathrm{sign}(\mathbf{A}^c\mathbf{A}^{p\mathrm{T}}) \odot (\sqrt{|\mathbf{A}^c\mathbf{A}^{p\mathrm{T}}|}}{\sqrt{d}})\mathbf{A}^h, \tag{3}$$

where $\mathrm{sign}(\cdot)$ is a function with value 1 when the element is positive or -1 when negative; $|\cdot|$ for absolute value; square root ensures that the value scale remains the same. This method utilizes matrix multiplication to make fully interaction between all state values, obtaining the states conditioned on all information sources dynamically. History information is selected as the "value" term to get more dialog history involved in the obtained state.

3 Experiment

We employ the PersonaChat (Zhang et al., 2018; Dinan et al., 2020) dataset in our experiments which has 164,356 utterances in 10,981 dialogues and 1,155 personas. Each sample contains dialog history with up to 15 utterances, a gold reply and a persona description with no more than 5 sentences.

Four kinds of dialogue models using pretrained language models as the initialization are compared:
(i) **TransferTransfo** (Wolf et al., 2019), a single-input OpenAI GPT using token type embedding to distinguish different parts of a single concatenated input (persona profile, dialog history, and reply successively). We also replace original GPT in this method with GPT2, denoted as **TransferGPT2**.
(ii) **MI-GPT** (Golovanov et al., 2019) which uses the OpenAI GPT in both encoder and decoder with average pooling as the attention fusion method.
(iii) Our architecture using GPT2 as the base model and average as fusion method (**GPT2-avg**), a very simple baseline inherited from MI-GPT.
(iv) Our model with each of the attention fusion methods discussed in Sec 2.3, denoted as **GPT2-X**, and **X** is the corresponding fusion method.

All GPT2 models used here are small size (12 layers, hidden size is 768). Besides, Seq2seq model with attention (Bahdanau et al., 2014) using 6-layer Transformer as the encoder and decoder is also included as an end-to-end single-input baseline.[1]

The following automatic metrics are considered in our evaluation: BLEU (Papineni et al., 2002), METEOR (Lavie and Agarwal, 2007), NIST-4, which indicate the gram-level similarity between the references and generated responses. Moreover, Entropy-4, corpus-level Distinct-2 and the average length of replies are used to reflect the diversity of obtained text. In addition, human evaluation is also conducted on 200 dialog pairs in terms of fluency (range: $1 \sim 3$), relevance with dialogue history (h-rel, range: $1 \sim 3$) and consistency with personality (p-consist, $\{0, 1\}$). More experiment configurations can be found in Appendix A.

[1] Source code is available at: `https://github.com/caoyu-noob/Multi-GPT2`

Model	BLEU	METEOR	NIST-4	Entropy-4	Dist-2	Avg.len	fluency	h-rel	p-consist
Human	-	-	-	10.725	36.688	11.507	2.901	2.645	0.598
Seq2seq	1.769	6.926	1.028	6.789	6.356	8.710	-	-	-
TransferTransfo	2.054	7.672	1.183	8.429	17.917	7.824	2.748	2.348	0.542
MI-GPT	3.151	8.112	1.264	8.054	13.264	9.026	2.809	2.150	0.628
TransferGPT2	2.273	7.872	1.194	8.263	16.444	8.036	2.785	**2.385**	0.548
GPT2-avg	3.211	8.149	1.291	7.904	13.612	8.932	**2.838**	2.149	**0.648**
GPT2-max	3.344	8.156	1.206	8.175	14.104	8.750	-	-	-
GPT2-min	3.774	8.661	1.388	8.099	14.925	9.209	-	-	-
GPT2-sw	**3.949**	**8.881**	**1.407**	8.228	15.294	9.068	**2.814**	**2.355**	0.595
GPT2-dw	3.714	8.694	1.385	8.096	14.647	9.095	-	-	-
GPT2-linear	**4.147**	**8.988**	**1.408**	8.279	15.237	9.011	2.777	2.332	**0.602**
GPT2-att	3.659	8.449	1.249	8.028	14.592	8.723	-	-	-

Table 1: Dialogue generation performance comparison of different models on the test set of PersonaChat. Values for BELU, METEOR and Dist-2 are in percentage. Human evaluation is only conducted on representative models.

3.1 Results

Results of different models on both automatic metrics and human evaluations are shown in Table 1.

We first analyze results on automatic metrics. It can be observed that GPT2 is more powerful than OpenAI GPT under the same architecture. Multi-input (MI) models that use the encoder-decoder frameworks generally outperform single-input (SI) models (TransferTransfo, TransferGPT2) which simply concatenate all inputs. Although SI models show higher diversity, their generated texts are generally shorter. All attention fusion methods of our model make improvements compared to its baseline GPT2-avg. Among them, weighting methods have higher scores than the other two kinds of fusion methods on most metrics. Compared with static methods, weighting methods are more flexible to combine proper proportions of each source, thus it is no surprise that they can outperform static methods. Meanwhile, though the attention-based method also allows for non-static attention fusion, it essentially poses dynamic weights on the history state, and thus information of persona and reply is not directly used in the final fused representation and results in its failure It is also interesting to find that GTP2-dw shows no improvement compared to GPT2-sw, despite it extends the latter one using different weights for each dimension.

Now we discuss human evaluation results. Here, we only conduct human evaluations on baselines and proposed models with the best automatic evaluation results (i.e. weighting methods). Fluency scores of generated texts are very close to each other even compared to gold replies, which should be largely benefited from the pretrained model. However, h-rel scores (the relevance between dialog history and current responses) by models are significantly lower than those by a human. Note that compared with SI models, MI models using the average fusion (MI-GPT, GPT2-avg) show lower h-rel scores, though their persona consistency increases much. This is also discussed in Golovanov et al. 2019, and the reason is that SI model is similar to a language model which stays tightly with history, while MI models take persona as a separate input which is easier to reuse personalized word. However, our models with the weighting fusion methods can not only improve the persona consistency compared to SI models, but also maintain comparable best history relevance. The case study of generated replies is given in Appendix B.

3.2 Influence of Attention Fusion

In this section, we further investigate how attention fusion affects the generation results, especially why using the average fusion decreases the performance on the relevance between dialog history and generated responses while the weighting fusion methods can survive.

We group the 200 testing samples for human evaluation by their lengths of history, and then compare the average results on **h-rel** scores of different methods within each group. Results are shown in Table 2. We first compare the weighting fusion methods with the average fusion baseline. As can be seen, all methods perform comparably when dialogue history is short. With longer dialog history, models with weighting fusion methods perform

	History	**Win**	**Tie**	**Lose**
GPT2-weight	L	53.2%	28.2%	18.6%
vs.	M	37.0%	31.1%	31.9%
GPT2-avg	S	29.3%	45.2%	25.5%
GPT2-weight	L	39.7%	35.5%	24.8%
vs.	M	28.9%	37.1%	34.0%
TransferGPT2	S	24.1%	43.7%	32.2%
MI baselines	L	17.7%	30.1%	52.2%
vs.	M	22.2%	28.9%	48.9%
SI baselines	S	18.9%	42.8%	38.3%

Table 2: Percentage of generated replies by the upper model better, equal or worse than the bottom one on **h-rel** metric. Samples are grouped by dialog history length (long (L) / short (S) / medium (M) history length: $>$ 9 utterances / $\leq$ 3 utterances / rest samples.). GPT2-weight: GPT2-sw and GPT2-linear, MI baselines: GPT-MI and GPT2-avg, SI baselines: TransferTransfo and TransferGPT2.

much better than GPT2-avg. The reason is that when dialogue history gets longer, the effect by each history token on current reply in bi-attention is averaged out by dialogue history length, making the average fusion method harder to capture key information from any history token to generate the response. Next, we compare the GPT2 with weighting fusion methods to TransferGPT2 (the SI model with GPT2) and results indicate that they can also outperform SI models when dialogue history is long. Finally, we can see that SI models beat the MI baselines with the average fusion under all conditions, proving the ineffectiveness of the simple average between different information sources.

Figure 2 further illustrates the estimated optimal weights of each attention information in every decoder layer in GPT2-sw. We observe that attention weights of different input sources are not equal and change over different decoder layers, validating that the use of average fusion is over-simplified. The weights of diverse sources tend to be equivalent in high layers while they differ significantly in lower layers because the history and persona information are already encoded and highly abstractive.

4 Conclusion

To handle dialogue generation with multiple input sources, we adapt the pretrained language model GPT2 to an encoder-decoder architecture with multiple independent attentions for different input sources in the decoder. We then investigate several attention fusion methods to obtain a preferable representation for dialogue generation. Experiments illustrate that weighting methods promote both auto metrics and dialog history relevance scores annotated by human than baselines using average fusion, while they still maintain the persona consistency scores which outperform single-input models. And such architecture can be extended to other multi-input dialogue generation tasks having different information source number.

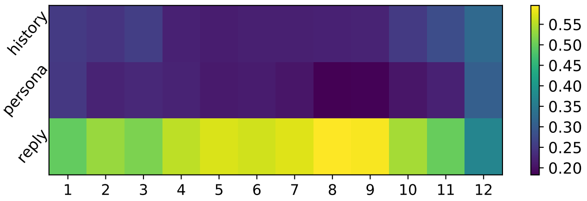


Figure 2: Visualization of normalized scalar attention weights on 3 different input sources for each layer in GPT2-sw decoder.

Acknowledgements

This work was supported by Australian Research Council Projects under grants FL-170100117, DP-180103424, and IC-190100031.

References

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

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. *arXiv preprint arXiv:1907.05774*.

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.

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

Sergey Edunov, Alexei Baevski, and Michael Auli. 2019. Pre-trained language model representations for language generation. 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 4052–4059.

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.

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

Alon Lavie and Abhaya Agarwal. 2007. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *Proceedings of the second workshop on statistical machine translation*, pages 228–231.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and Bill Dolan. 2016. 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.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496.

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. Association for Computational Linguistics.

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.

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.

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*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. 2018. Recent trends in deep learning based natural language processing. *ieee Computational intelligenCe magazine*, 13(3):55–75.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2019. Dialogpt: Large-scale generative pre-training for conversational response generation. *arXiv preprint arXiv:1911.00536*.

Yu Zhang and Qiang Yang. 2017. A survey on multi-task learning. *arXiv preprint arXiv:1707.08114*.

Yinhe Zheng, Rongsheng Zhang, Xiaoxi Mao, and Minlie Huang. 2019. A pre-training based personalized dialogue generation model with persona-sparse data. *arXiv preprint arXiv:1911.04700*.

A Experiment Details

We use the official code for the implementation of TransferTransfo (Wolf et al., 2019) and GPT2-MI (Golovanov et al., 2019), following all default settings to fine-tune models. To implement our TransferGPT2, GPT2-avg, and all refined attention fusion model, we utilize HuggingFace Transformers library[2] with the small-size GPT2 model which has 12 layers and 768 dimensions in the hidden state. It is noted that although both our encoder and decoder are initialized from GPT2 model, their parameters are not shared. Similarly, 3 different attention modules in each layer of the decoder (1

[2] https://github.com/huggingface/transformers

self-attention, 2 bi-attention) are also initialized by the attention module of the corresponding layer in original GPT2 model but parameters are also not shared among them. The parameters of the additional attention fusion module will be initialized by: 1) uniform initialization for source-weighting methods, and 2) random initialization with normal distribution for linear and attention-based methods. And the linear prediction layer has the shared weight with the embedding layer of the decoder.

During fine-tuning, we use Adam optimizer (Kingma and Ba, 2014) with an initial learning rate 5e-4 with 0.002 warmup proportion and then a linear decay. The learning rate for the additional attention fusion module is 5× current learning rate for other parts. We train it for 5 epochs using mini-batch with size 256. And only the latest 7 utterances in dialog history are remained to avoid exceeding maximum input length. All hyperparameters are determined by manually tuning according to auto metrics BLEU, METEOR ,and NIST as criteria.

During inference, we use beam search with size 3 for all test models. Length penalty (Wu et al., 2016) is added to ensure the diversity of generation. A single NVIDIA V100 GPU with CUDA10 is used to run experiments.

B Case Study

We list dialogue generation results of TransferGPT2, GPT2-avg, GPT2-sw and GPT2-linear under some cases from PersonaChat dataset (Zhang et al., 2018) in Table 3 and Table 4, containing samples with varied dialog history lengths. h-rel and p-consist indicate the human evaluation scores for dialogue history relevance and personality consistency of generated replies respectively.

It can be found that our refined attention fusion models generally show similar personality consistency with the baseline GPT2-avg model who uses the same architecture but a simple average method to combine different information sources. When dialog history is long, TransferGPT2 tends to directly respond to the last history utterance using some general replies, while GPT2-avg tends to directly copy personal information as replies. GPT2-sw and GPT2-linear can properly make a response to the last context as well as involve personal profile. In addition, we find that when history length is not so long (length is 5 or 7), such difference will be reduced. But when dialog history is very short (less than 5), all encoder-decoder models tend to generate universal replies or simply reuse personalities because the history information is too limited for them to combine it with the given personal profile. While the single-input TransferGPT2 is inclined to reuse personality descriptions because the whole input sequence length is shorter and persona information obtains more attention compared to the input having a long history.

item	text	h-rel	p-consist
Personality	i have one cat. i am a kindergarten teacher. i can barely pay my bills every month. i share an apartment in seattle with two roommates. i just graduated from college two years ago.		
Dialog history (length=13)	... A: i also love to travel and take adventures. what are your hobbies? B: i like to hang out with friends A: nice. where do you like to go? B: we like to do different things do you have pets? A: i have a dog. his name is max		
Human	that is cute i have a cat.	3	1
TransferGPT2	that is a cute name.	3	0
GPT2-avg	i have two cats and a dog.	2	0.5
GPT2-sw	that is cool i have two cats.	3	0.5
GPT2-linear	that is a cute name i have a cat named maxy.	3	1
Personality	my favorite tv show is the office. i do not drive because i live in new york. i like jazz music. i do not eat anything sweet.		
Dialog history (length=9)	... A: i need something to mellow me out after working all day B: that can be hard to find up in new york. things are fast and tight here A: do you have many insects? i do not care for that. B: not in the city, but yes upstate. A: what do you do for a living? i am in the science field engineer		
Human	i am a nutritionist , so no sweets for me.	3	1
TransferGPT2	i am a stay at home mom.	3	0
GPT2-avg	i live in new york city.	2	1
GPT2-sw	i do not work. i just sit around and listen to jazz music.	3	1
GPT2-linear	that is cool. i work in a lab in new york city.	3	1

Table 3: Some cases of generated dialogue replies by TrnasferGPT2, GPT2-avg, GPT2-sw and GPT2-linear.

item	text	h-rel	p-consist
Personality	i am a huge fantasy fan. i want to write fantasy books myself. i have two part time jobs. i like to binge watch the lord of the rings		
Dialog history (length=7)	... A: ah gosh. that is the worst! B: i want to just stay home and write. A: i would rather play softball than go to work B: i like to role play and play fantasy games. have you been to com icon? A: what is that? no i have never been		
Human	have you ever heard of the lord of the rings?.	2	1
TransferGPT2	that is cool. i am a huge fantasy fan.	2	1
GPT2-avg	it is a fantasy game.	3	1
GPT2-sw	it is a fantasy world with lots of dragons.	3	1
GPT2-linear	i have not. i am more of a fantasy fan.	2	1
Personality	i do not like exercise or physical activity. i am a homebody. my favorite food is ice cream. i have a husband who i love very much. i love watching movies and tv.		
Dialog history (length=5)	A: hello ! tell me about yourself? B: i really like to stay at home at watch a movie with my darling hubby. you? A: that sounds nice! i am a nurse who loves zoos and teddy bears. B: aww do you collect teddy bears? my sweet sweet husband buys me them A: i do! i bring them to my patients at the childrens hospital.		
Human	so nice! do they also like ice cream? i love it!	3	1
TransferGPT2	i do not have any kids, but i do have a husband.	1	1
GPT2-avg	that's so sweet! i love to eat ice cream and ice cream.	3	1
GPT2-sw	that sounds like a lot of fun. i am a homebody myself.	3	1
GPT2-linear	that is nice. i am a homebody myself.	3	1
Personality	i am going to auburn university next year as a freshman. i love shopping fashion and clothes. i want to be a doctor when i grow up. i am 17 years old and a senior in high school.		
Dialog history (length=3)	A: hi i am talking to text B: hello , interesting introduction you have there A: colors i can not see them well , my sis helps me out		
Human	that is unfortunate. what is something that you enjoy doing?	3	0
TransferGPT2	i am a junior in high school, i want to be a doctor	3	0
GPT2-avg	what do you do for a living?	3	0
GPT2-sw	i am sorry to hear that.	1	1
GPT2-linear	what do you do for a living?	3	0

Table 4: Some cases of generated dialogue replies by TrnasferGPT2, GPT2-avg, GPT2-sw and GPT2-linear.