

Controllable Multi-attribute Dialog Generation with PALs and Grounding Knowledge

Evseev D. A.

Moscow Institute of Physics
and Technology
Dolgoprudny, Russia
dmitrij.euseew@yandex.ru

Nagovitsin M. S.

Moscow Institute of Physics
and Technology
Dolgoprudny, Russia
nagovitsin.ms@phystech.edu

Kuznetsov D. P.

Moscow Institute of Physics and Technology
Dolgoprudny, Russia
kuznetsov.den.p@gmail.com

Abstract

Today, neural language models are commonly employed for generation of natural like responses in dialog systems. The main issue that limits wide adoption of neural generation is related to poor predictability of responses in terms of content, as well as dialog attributes such as dialog acts and sentiment. In this paper we propose a method based on projected attention layers (PALs) for controllable multi-attribute knowledge grounded dialog generation. We compared a number of methods for training and blending representations produced by PALs combined with DialoGPT base model. Results of our experiments demonstrate that separate pre-training of PAL branches for different attributes followed by transfer and fine-tuning of dense blending layer gives the highest accuracy of control of a generated response for fewer trainable parameters per an attribute. Furthermore, we applied our approach for controllable multi-attribute generation with grounding knowledge to Blenderbot model. Our solution outperforms the baseline Blenderbot and CRAYON model in control accuracy of dialog acts and sentiment on Daily Dialog as well demonstrates a comparable overall quality of dialog generation given grounding knowledge on Wizard of Wikipedia.

Keywords: controllable dialog generation, knowledge grounded dialog generation, projected attention layers.
DOI: 10.28995/2075-7182-2022-21-185-202

Управляемые диалоговые генеративные модели с параметризацией атрибутов и дополнительным контекстом генерируемых реплик на основе PALs

Евсеев Д. А.

Московский физико-технический
институт
Долгопрудный, Россия
dmitrij.euseew@yandex.ru

Наговицин М. С.

Московский физико-технический
институт
Долгопрудный, Россия
nagovitsin.ms@phystech.edu

Кузнецов Д. П.

Московский физико-технический институт
Долгопрудный, Россия
kuznetsov.den.p@gmail.com

Аннотация

В настоящее время нейронные языковые модели широко используются для генерации реплик в диалоговых системах. Основным недостатком нейронной генерации связан с непредсказуемостью содержания реплик. В данной статье мы предлагаем метод, основанный на проекционных слоях внимания (PAL) для управления несколькими параметрами генерируемой реплики. Мы сравниваем несколько методов обучения и смешивания представлений, полученных с помощью PALs, добавленных к DialoGPT. В ходе экспериментов было установлено, что обучение PALs отдельно

для каждой из веток управляемых параметров и последующее дообучение полносвязного смешивающего слоя приводит к наибольшей точности управляемой генерации, при этом используя меньшее число обучаемых параметров. Также мы применили наш подход для генерации с контролем атрибутов опираясь на внешние знания, с помощью модели Blenderbot. Наше решение превосходит базовую версию Blenderbot и модель CRAYON по точности предсказания диалогового акта и сентимента генерируемых реплик на датасете Daily Dialog, и при этом показывает сравнимое качество генерации реплик с использованием контекста на датасете Wizard of Wikipedia.

Ключевые слова: управляемая генерация реплик в диалоге, генерация реплик по контексту, проекционные слои внимания.

1 Introduction

Majority of open-domain dialog systems use hand-crafted finite state machines for response generation (Larsson and Traum, 2000), (Bocklisch et al., 2017), (Finch and Choi, 2020). For every expected user utterance these systems define a state with pre-defined output response and transition to the next state of the dialog, but user input can mismatch a condition for transition in the current state. As well, the user input can mismatch all possible states defined by the finite state machine. Here, neural generative models are able to help with producing natural like responses. Unfortunately, generative models demonstrate very unreliable coherence with existing dialog context (Abhishek et al., 2021). One of the possible solution is to use controllable attributes such as dialog act or sentiment to guide generation of responses and return the dialog flow back to the domain of pre-defined script. If the script is defined as pairs of adjacent dialog acts, or in goal-oriented dialog, the attributes of the system response (intent, slot values) are extracted (Wu et al., 2019), for known attributes of the response a generative model conditioned on grounding knowledge about entities found in the dialog context or slot values, is able to generate all the bot utterances in the script without retrieval of hand-written responses. Also the right level of control can improve dialog quality in different aspects (See et al., 2019). Social bots like Gunrock (Yu et al., 2019) and XiaoIce (Zhou et al., 2019) are actively using dialog acts, sentiment and other attributes for dialog management.

Controllable generative models have been an active area of research over last years. Models (Zhao et al., 2017), (Zhang et al., 2018) control one attribute of the generated response (dialog act, response relatedness or specificity). The need to control different attributes simultaneously is present in (See et al., 2019). General approaches to control multiple attributes simultaneously were presented for different model architectures in (Hu et al., 2021b), (Xu et al., 2019), (Yu et al., 2021), (Du and Ji, 2021), (Yang et al., 2021). In this paper we propose and study a technique for multi-attribute generation control which is suitable for the both pre-training as well as fine-tuning. We use PALs (Stickland and Murray, 2019) with transformer architectures, consequently parameters of the main pre-trained model provide constant background knowledge and PAL layers are trained to control generation in respect with specific attribute.

Informativeness and meaningfulness is another important aspect of generated responses. Blenderbot (Roller et al., 2020), CoLV (Zhan et al., 2021) and CGRG (Wu et al., 2021) use grounding knowledge (retrieved paragraphs) to control the content of output utterances. But these models are not able to be controlled to produce the response with required attributes, such as dialog act or sentiment.

Trained models for English language, training and inference code and data to test the quality of models are published in Open Source under the Apache 2.0 license¹. The main contributions of this work are the following:

- we develop the method of controllable generation for several simultaneous attributes such as dialog acts and sentiment with no changes in weights of the original model;
- we study simultaneous control of knowledge grounding as well as dialog act and sentiment of a response, and find that our model outperforms existing approaches in terms of dialog act and sentiment control accuracy and is competitive in terms of perplexity of knowledge grounded generation.

2 Related Work

There are many different approaches to control generation process with or without changing the architecture or retraining the initial language model. PPLM (Dathathri et al., 2019) updates latent represent-

¹<https://github.com/deepmipt/controllable-generation>

ations of model during decoding with help of pre-trained discriminator, GeDI (Krause et al., 2020) uses class-conditional distributions to achieve control for both desired and undesired attributes (for example, generate less toxic answers). Both PPLM and GeDI work without changing the initial model.

Another way of control is to add special control tokens as an input for model. It can be done by fine-tuning (or full training) of model, as proposed in CTRL (Keskar et al., 2019), or by keeping the initial model as is and training a special alignment function which will generate proper key and value representations for control tokens on each level of pre-trained Transformer model, as proposed in Attribute Alignment (Yu et al., 2021). Those methods are capable of controlling multiple attributes by adding tokens for each one.

For larger models prefix tuning (Li and Liang, 2021) can be applied. Prompts also can be used for a few shot learning (Zheng and Huang, 2021).

Some methods work with a latent space, for example CRAYON (Hu et al., 2021b), which is based on LSTM, or PHED (Yang et al., 2021), which uses Transformer combined with CVAE to add a latent space to the Transformer architecture. Model GTMES2S, proposed in (Xu et al., 2019) uses additional modules to control current level of all attributes and guide further generation to reach desired values. Each of CRAYON, PHED and GTMES2S are working with multiple attributes. In XiaoIce dialog assistant (Zhou et al., 2019) GRU is used to generate responses, conditioned on query and response empathy vectors.

The other method is to add special modules to shift embeddings (and maybe hidden states) from initial model, same as with PPLM (Dathathri et al., 2019), but to learn a proper transformations in advance and not to tune them on inference. SideControl (Du and Ji, 2021) propose a way to perturb embeddings of any language model by training additional module, which will take class embedding or grounding knowledge into account. Another way is to treat control as a special task and add adapters (Houlsby et al., 2019) for different attributes we want to control. One of such models is Adapter-Bot (Madotto et al., 2020) which has an option of switch between different attributes without changes in initial model. Hyperformer (Mahabadi et al., 2021) utilizes a shared PAL parameters for all tasks and Transformer layers, these parameters are generated by a hypernetwork. The model (Xie and Pu, 2021) is an encoder-decoder Transformer, where emotions in response are controlled with emotion embeddings, fed into the model. But models with adapters are not able to control multiple attributes simultaneously without additional work. Task-specific parts to the model can also be efficiently added with help of low-rank decomposition, as proposed in LoRA (Hu et al., 2021a).

The key points in which our model differs from the others are following:

- Ability to control of multiple attributes simultaneously (dialog act and sentiment in our study);
- Preserving the weights of initial model in original state;
- Ability to pre-train all adapters independently on different datasets;
- Scalability in terms of number of controllable attributes and number of their values;

Most of generative models, which do not use external knowledge, are capable of producing grammatically correct and natural responses given the dialog history, but have a limited ability to generate interesting responses based on facts. On the other hand, knowledge-grounded generative models have an option of controlling content of generated responses with sentences with facts or keywords. CGRG (Wu et al., 2021) model uses lexical control phrases to control the generated response. The approach of (Xu et al., 2021b) is based on PALs for different topics which are used for retrieval-free knowledge grounded generation. The model (Zhan et al., 2021) uses latent variables for relevant knowledge selection and response generation. The models (Xu et al., 2021a), (Kumar et al., 2021) and (Gupta et al., 2020) controls the generated response by adding as input of the transformer the sequence of keywords before the dialog history. Our approach is inspired with Blenderbot (Roller et al., 2020) which is an encoder-decoder transformer pretrained on Reddit and finetuned on Wizard of Wikipedia (Dinan et al., 2018), but our model controls not only the content of the response and moreover dialog act and sentiment.

3 Methods

In this paper, our goal is to find a method to control different response attributes without losing much token prediction quality (perplexity) and other abilities of the base pre-trained model (e.g., using ground-

ing knowledge). We did most of our experiments with DialoGPT-small architecture (Zhang et al., 2020b), because of the affordable time to fine tune and the good quality of the pre-trained model. Additional experiments with simultaneous control of content, dialog acts and sentiment we performed with Blenderbot architecture (Roller et al., 2020). Furthermore, we chose dialog acts (inform, question, directive, commissive) and sentiment (negative, neutral, positive) as controlled attributes. For evaluation of control accuracy we used DailyDialogs (Li et al., 2017), sentiment labelling was made separately by classifier. For evaluation of knowledge-grounded dialog generation quality (perplexity) we used Wizard of Wikipedia dataset (Dinan et al., 2018).

Control	Blend	Train dataset	Dialog act acc.	Sentiment acc.	Perplexity	Opt. steps	Trainable par.
No control	-	DailyDialogs	25.20 \pm 0.21	33.41 \pm 0.15	15.19 \pm 1.58	2000	117M
Dialog acts	average	DailyDialogs	63.74 \pm 0.32	42.83 \pm 0.27	15.93 \pm 0.12	10000	36M
Dialog acts	dense	DailyDialogs	45.27 \pm 5.26	40.15 \pm 1.22	22.36 \pm 0.85	5000	49M
Sentiment	average	ScenarioSA	33.40 \pm 0.16	72.09 \pm 4.06	92.98 \pm 14.74	5000	28M

Table 1: Models with control of one attribute. The model with no control is a finetuned DialoGPT-small, models with control are DialoGPT-small with PALs. Metrics were calculated on valid set of Daily Dialog.

Blend	Transfer	Dialog act acc.	Sentiment acc.	Perplexity	Opt. steps	Trainable par.
average	no	63.09 \pm 2.22	69.19 \pm 1.10	17.12 \pm 0.40	5000	63M
dense	no	61.65 \pm 1.02	67.10 \pm 1.38	22.07 \pm 0.39	5000	84M
dense & average	no	61.36 \pm 1.40	68.12 \pm 0.69	15.51 \pm 0.13	5000	77M
average	yes	65.62 \pm 2.04	66.04 \pm 0.25	17.74 \pm 0.75	5000	63M
weighted average	yes	63.20 \pm 1.09	69.05 \pm 0.42	15.65 \pm 0.09	5000	63M
dense	yes	60.83 \pm 1.35	67.80 \pm 3.37	21.34 \pm 0.40	5000	84M
dense & average	yes	62.76 \pm 0.70	70.03 \pm 2.04	15.69 \pm 0.19	5000	77M
dense & average, only blend	yes	60.19 \pm 0.76	67.47 \pm 0.90	15.30 \pm 0.05	10000	14M

Table 2: Models with simultaneous dialog act and sentiment control. Transfer means that PALs were initialized with weights from model for single attribute control.

3.1 Projected Attention Layers as Multitask Adapters

One of the approaches to control object attributes is to learn proper shifts in latent space (Hu et al., 2021b). One way to modify latent representations for every token is to use *Projected Attention Layers (PALs)* (Stickland and Murray, 2019) as adapters for every controllable attribute. In our case, each PAL will learn to correct hidden states of the main model to generate a response with the desired attribute (Figure 1).

3.2 Blend layer

To control several attributes simultaneously, we decided to add a PAL for each attribute and run them in parallel (Figure 1). We chose *average blending* as our baseline for blending of hidden representations. It allows us to control easily the contribution of each PAL to the resulting hidden states by weighting them. Then we try a trainable way of blending outputs of PAL branches: *dense blending* — concatenation of PALs outputs and the main branch and feeding into the dense layer; *combination of dense and average blending* — concatenation of PALs outputs, feeding into the dense layer and averaging the output with the base model. The loss function stays unchanged from the task of the next token prediction. For every labeled sample from training data we chose only corresponding PALs and train them, the base model is frozen.

3.3 Default branch

We added the *"default" branch* for each attribute for default selection values for attributes. Default branch is turned on for training on every sample instead of specialized PAL with probability $p = 0.2$. Thus default branch will be trained on all dataset and will not be bound to one attribute value.

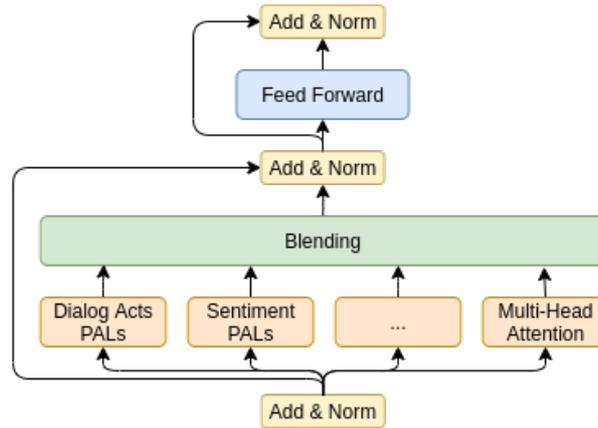


Figure 1: Blending of PALs and multi-head attention of Transformer hidden representations for every token.

3.4 Independent PALs pre-training

We independently trained models for dialog act and sentiment control and transferred these pre-trained branches into one model. Even without any further training resulting model demonstrated a noticeably good attribute control without huge degradation of perplexity, even though PALs for the sentiment were trained on a different dataset (more details in Appendix A.4). After transfer, the model with the blending layer can be finetuned on the target dataset.

3.5 Control of attributes and context of responses

One of our goals is to develop a model which could generate responses for a given grounding knowledge and global attributes, such as dialog act and sentiment. We modified Blenderbot Transformer architecture for control of global attributes of the response by adding PALs in parallel with the self-attention layer of the decoder layers. The decoder layer in our modification has 5 branches for dialog acts and 4 for sentiment. The attribute branches were blended with the dense layer and then added to the main branch of the base model.

4 Experiments and results

4.1 Evaluation

We used two metrics to estimate the quality of our models: perplexity to test that model is able to produce relevant and natural like responses and ability to control attributes. We generate responses for every turn on a validation part of DailyDialog and use attribute classifiers (see Appendix A.2) to check if the response of the model is correct and calculate balanced accuracy for each attribute. For example, for the dialog act attribute, we estimate the dialog act of each generated response and compare it with the gold label. Every model was trained for the same amount of steps, and then the best by perplexity checkpoint was scored. Blending experiments were performed with DialoGPT-small (117M) as a pre-trained base model. All parameters of PALs were taken from the original paper (Stickland and Murray, 2019), thus the PAL embedding dimension was 204. Training setup is the same as reported for original DialoGPT (Zhang et al., 2020b).

4.1.1 One attribute

When only one attribute is controlled there are no conflicts between PALs, because only one attribute shift is learned. We tried averaging and dense layer to blend the output of PAL and the layer of the main model (Table 1). The averaging is better in both perplexity and accuracy and is much easier for

further transfer because there is no need to add the blending layer to the target base model. Resources consumption is shown in Appendix A.1.

4.1.2 Two attributes

In the case of controlling multiple attributes simultaneously every PAL should adapt to its neighbors and learn to change only the corresponding attribute. Experiments (Table 2) have shown that the control abilities or perplexity are slightly better in the case of PALs pre-training and transfer compared to training added multi-attribute PALs from scratch. Average blending gives the best control for the similar perplexity. Dense layer blending results in perplexity drop. The model with a combination of dense and average blending shows the best perplexity and great control abilities. For other blending option perplexity is also on the same level, and control is better for one attribute and worse for another. Since each PAL was pre-trained with average blending, a more natural way to blend them is weighted average (see Appendix A.4), this gives better perplexity. With weighted average as a blending layer, it is possible to control the contribution of each PAL to every attribute. If the weights are transferred, another alternative to finetune the model is to train only blending layer. We choose combination of dense and average blending to finetune, and it results in the best perplexity and good control abilities (last row in the Table 2). Resources consumption is shown in Appendix A.1. Examples of dialogs can be found in Appendix A.9. More discussion on perplexity-accuracy tradeoff can be found in Appendix A.7.

Model	D.A. acc.	Sent. acc.	PPL
Bl. bot, cont., 199M	77.01	84.90	28.42
Bl. bot 400M	38.10	28.43	18.24
Bl. bot 90M	38.18	27.96	76.10

Table 3: Comparison of controllable Blenderbot (dense and average blending) with Blenderbot from Huggingface (balanced accuracy and perplexity) with grounding knowledge.

Model	Q/noQ acc.	Sent. acc.
Bl. bot, cont., d&avg	99.45	85.87
CRAYON	98.17	82.17

Table 4: Comparison of controllable Blenderbot (dense and average blending) with CRAYON model in question asking and sentiment control accuracy.

4.2 Blenderbot results

The next series of experiments was performed with Blenderbot for dialog acts and sentiment control (4 layers in encoder, 8 layers in decoder, embedding dimension of 576, 119M parameters). We pretrain Blenderbot on Reddit and finetuned on Daily Dialog, ConvAI2 (Dinan et al., 2020), Emphatetic Dialogue and Wizard of Wikipedia.

We compared Blenderbot with PALs and baseline Blenderbot on Daily Dialog dataset (Table 3). It was found that extended Blenderbot outperforms Blenderbot 400M and Blenderbot 90M from Huggingface library in dialog acts and sentiment control accuracy and is comparable with the baseline in perplexity of dialog generation given grounding knowledge (GK) on Wizard of Wikipedia dataset.

We compared controllable Blenderbot with CRAYON (Hu et al., 2021b) in question asking and sentiment control accuracy on Daily Dialog dataset. Our model controls 4 types of dialog acts, therefore we used PAL for "question" dialog act to generate a question and PAL for "inform" otherwise. Blenderbot with PALs outperforms CRAYON in question asking and sentiment control accuracy (Table 4).

5 Conclusion

In this paper with presented the study of techniques for multi-attribute control of neural response generation in the dialog with and without grounding knowledge. Our methodology employs extension of

pre-trained generative base model with attribute specific projected attention layers (PALs). Results of our experiments allow to draw the following conclusions.

If the base model is already trained and the quality of the responses is a first priority, then the best way is to pre-train PALs for each attribute separately (maybe on different datasets) with the average blending. Then transfer pre-trained PALs to the base model and finetune with weighted average or combination of average and dense blending. If a degradation of perplexity is not noticeably harmful then average blending without transfer is also an option due to ability to control the contribution of each attribute.

Our results demonstrate that proposed approach can be successfully applied to controllable generation of responses in the dialog conditioned on multiple attributes for less numbers of trainable parameters per attribute. The method can be also combined with grounding knowledge. Compared to the baseline our solution shows better accuracy of dialog acts and sentiment control with similar perplexity.

References

- Tushar Abhishek, Daksh Rawat, Manish Gupta, and Vasudeva Varma. 2021. Transformer models for text coherence assessment. *arXiv preprint arXiv:2109.02176*.
- Tom Bocklisch, Joey Faulkner, Nick Pawlowski, and Alan Nichol. 2017. Rasa: Open source language understanding and dialogue management. *arXiv preprint arXiv:1712.05181*.
- Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2019. Plug and play language models: A simple approach to controlled text generation.
- Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2018. Wizard of wikipedia: Knowledge-powered conversational agents. *arXiv preprint arXiv:1811.01241*.
- 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). // *The NeurIPS'18 Competition*, P 187–208. Springer.
- Wanyu Du and Yangfeng Ji. 2021. Sidecontrol: Controlled open-domain dialogue generation via additive side networks.
- James D Finch and Jinho D Choi. 2020. Emora stdm: A versatile framework for innovative dialogue system development. *arXiv preprint arXiv:2006.06143*.
- Prakhar Gupta, Jeffrey P Bigham, Yulia Tsvetkov, and Amy Pavel. 2020. Controlling dialogue generation with semantic exemplars. *arXiv preprint arXiv:2008.09075*.
- Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021a. Lora: Low-rank adaptation of large language models.
- Zhe Hu, Zhiwei Cao, Hou Pong Chan, Jiachen Liu, Xinyan Xiao, Jinsong Su, and Hua Wu. 2021b. Controllable dialogue generation with disentangled multi-grained style specification and attribute consistency reward.
- Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation.
- Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, Richard Socher, and Nazneen Fatema Rajani. 2020. Gedi: Generative discriminator guided sequence generation.
- Shachi H Kumar, Hsuan Su, Ramesh Manuvinakurike, Saurav Sahay, and Lama Nachman. 2021. Controllable response generation for assistive use-cases. *arXiv preprint arXiv:2112.02246*.
- Staffan Larsson and David R Traum. 2000. Information state and dialogue management in the trindi dialogue move engine toolkit. *Natural language engineering*, 6(3 & 4):323–340.
- Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation.
- Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. Dailydialog: A manually labelled multi-turn dialogue dataset.

- Andrea Madotto, Zhaojiang Lin, Yejin Bang, and Pascale Fung. 2020. The adapter-bot: All-in-one controllable conversational model.
- Rabeeh Karimi Mahabadi, Sebastian Ruder, Mostafa Dehghani, and James Henderson. 2021. Parameter-efficient multi-task fine-tuning for transformers via shared hypernetworks. *arXiv preprint arXiv:2106.04489*.
- Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Kurt Shuster, Eric M Smith, et al. 2020. Recipes for building an open-domain chatbot. *arXiv preprint arXiv:2004.13637*.
- Abigail See, Stephen Roller, Douwe Kiela, and Jason Weston. 2019. What makes a good conversation? how controllable attributes affect human judgments. *arXiv preprint arXiv:1902.08654*.
- Asa Cooper Stickland and Iain Murray. 2019. Bert and pals: Projected attention layers for efficient adaptation in multi-task learning.
- Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. Transferable multi-domain state generator for task-oriented dialogue systems. // *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, P 808–819, Florence, Italy, July. Association for Computational Linguistics.
- Zequ Wu, Michel Galley, Chris Brockett, Yizhe Zhang, Xiang Gao, Chris Quirk, Rik Koncel-Kedziorski, Jianfeng Gao, Hannaneh Hajishirzi, Mari Ostendorf, and Bill Dolan. 2021. A controllable model of grounded response generation.
- Yubo Xie and Pearl Pu. 2021. Empathetic dialog generation with fine-grained intents. // *Proceedings of the 25th Conference on Computational Natural Language Learning*, P 133–147.
- Can Xu, Wei Wu, Chongyang Tao, Huang Hu, Matt Schuerman, and Ying Wang. 2019. Neural response generation with meta-words.
- Heng-Da Xu, Xian-Ling Mao, Zewen Chi, Fanshu Sun, Jingjing Zhu, and Heyan Huang. 2021a. Generating informative dialogue responses with keywords-guided networks. // *CCF International Conference on Natural Language Processing and Chinese Computing*, P 179–192. Springer.
- Yan Xu, Etsuko Ishii, Samuel Cahyawijaya, Zihan Liu, Genta Indra Winata, Andrea Madotto, Dan Su, and Pascale Fung. 2021b. Retrieval-free knowledge-grounded dialogue response generation with adapters. *arXiv preprint arXiv:2105.06232*.
- Haiqin Yang, Xiaoyuan Yao, Yiqun Duan, Jianping Shen, Jie Zhong, and Kun Zhang. 2021. Progressive open-domain response generation with multiple controllable attributes.
- Dian Yu, Michelle Cohn, Yi Mang Yang, Chun-Yen Chen, Weiming Wen, Jiaping Zhang, Mingyang Zhou, Kevin Jesse, Austin Chau, Antara Bhowmick, Shreenath Iyer, Girithija Sreenivasulu, Sam Davidson, Ashwin Bhandare, and Zhou Yu. 2019. Gunrock: A social bot for complex and engaging long conversations.
- Dian Yu, Zhou Yu, and Kenji Sagae. 2021. Attribute alignment: Controlling text generation from pre-trained language models.
- Haolan Zhan, Lei Shen, Hongshen Chen, and Hainan Zhang. 2021. Colv: A collaborative latent variable model for knowledge-grounded dialogue generation. // *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, P 2250–2261.
- Ruqing Zhang, Jiafeng Guo, Yixing Fan, Yanyan Lan, Jun Xu, and Xueqi Cheng. 2018. Learning to control the specificity in neural response generation. // *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, P 1108–1117.
- Yazhou Zhang, Zhipeng Zhao, Panpan Wang, Xiang Li, Lu Rong, and Dawei Song. 2020a. Scenariosa: A dyadic conversational database for interactive sentiment analysis. *IEEE Access*, 8:90652–90664.
- Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020b. Dialogpt: Large-scale generative pre-training for conversational response generation.
- Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. 2017. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. *arXiv preprint arXiv:1703.10960*.
- Chujie Zheng and Minlie Huang. 2021. Exploring prompt-based few-shot learning for grounded dialog generation.
- Li Zhou, Jianfeng Gao, Di Li, and Heung-Yeung Shum. 2019. The design and implementation of xiaoice, an empathetic social chatbot.

A Appendix

A.1 Resources

For all experiments, we used NVIDIA GeForce GTX 1080 Ti GPUs. Training DialoGPT-small with one attribute control for 10000 steps took about 8 hours using two GPUs. Training model with two attribute control and (weighted) average blending for 5000 steps took about 6 hours, with dense blending - about 8 hours, and with a combination of average and dense blending - 7 hours on two GPUs. Train only blend layer for a combination of dense and average took about 11 hours on the same devices. The batch size was set to 256 divided into 8 steps of gradient accumulation. Extended Blenderbot was trained with batch size of 1000 on 10 NVIDIA GeForce GTX 1080 Ti GPUs. Pretraining on part of Reddit dataset (dump from 2014 and 2015 years) took 48 hours.

A.2 Evaluation and Classifiers

We used the validation part of the DailyDialog (Li et al., 2017) dataset to evaluate our models. Daily-Dialog is labeled with dialog acts, moreover we needed labels for the sentiment. Number of utterance for each attribute is shown on Figure 4. Since classes are not balanced, we used balanced accuracy (from package scikit-learn 0.21.2, `sklearn.metrics.balanced_accuracy`). To evaluate the model we generated responses on the test set with the right PALs (according to the gold labels) and check if the response was generated with desired attributes. Consequently we needed to classify dialog acts and sentiment to (1) evaluate our model and (2) label datasets automatically.

For dialog acts and sentiment classification we used the BERT-based model. One (current) or two utterances (current and previous), separated with SEP-token, were fed into BERT. The hidden state of the BERT CLS-token was fed into the dense layer, followed by softmax classification. Dialog acts classifier was trained on Daily Dialog (Li et al., 2017), sentiment classifier - on Scenario SA (Zhang et al., 2020a). Balanced accuracy of dialog act classifier is 72.90%, the confusion matrix is in Figure 2. The balanced accuracy of the sentiment classifier is 76.24%.

A.3 Default branch

We added "*default*" branch for each attribute for the cases when we don't want or don't need to control it. The default branch is the same PAL as the other, except during training it turns on every time instead of any other PAL for this attribute with the probability p , we chose $p = 0.2$. To check that the default branch is working as expected, we evaluated the model (DialoGPT-small with control of dialog acts and sentiment and combination of dense and average as a blend layer) in four setups:

- Usual inference (default branch is off)
- Default branch is always set for dialog act attribute
- Default branch is always set for sentiment attribute
- Default branch is always set for both dialog act and sentiment attributes

The results are in the Table 5. With default control of each attribute is back on the level of base models (without attribute control). With default branches, perplexity grows, but not too much. That means that those branches are trained pretty well and that our model is better at control (than base DialoGPT) not just because of the larger number of parameters, but because PALs are learning their domains. Otherwise, default branches would show great control abilities too.

Default attributes	Dialog act acc.	Sentiment acc.	Perplexity
No default	62.76 \pm 0.70	70.03 \pm 2.04	15.69 \pm 0.19
Dialog act	31.97 \pm 0.42	70.19 \pm 2.27	16.34 \pm 0.28
Sentiment	62.85 \pm 0.38	42.47 \pm 0.75	16.03 \pm 0.17
Dialog act and sentiment	30.06 \pm 0.79	39.67 \pm 0.31	16.83 \pm 0.57

Table 5: Work of default branches for each attribute. Evaluated with the model for dialog act and sentiment control with a combination of dense and average blending.

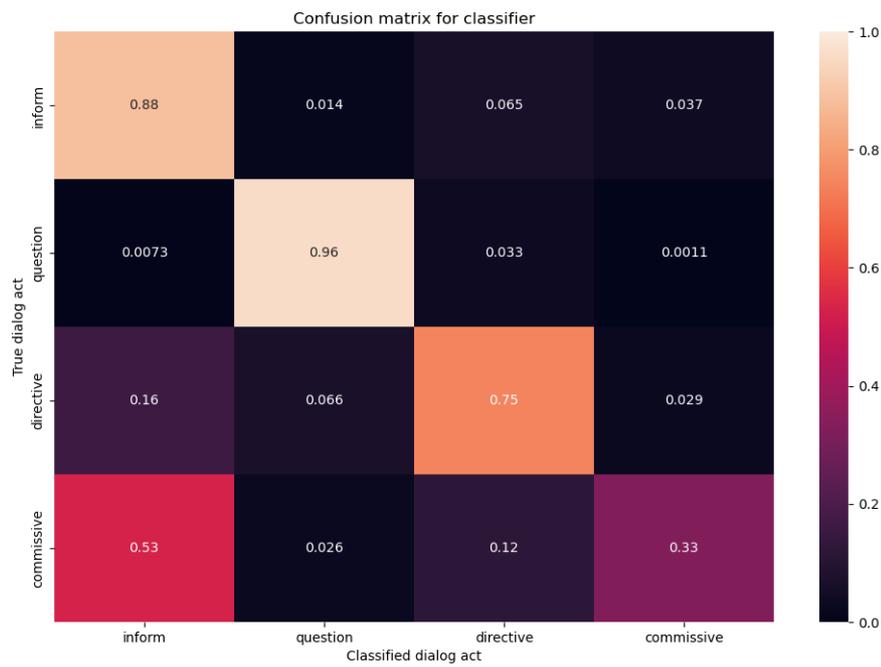


Figure 2: Confusion matrix for dialog acts classifier.

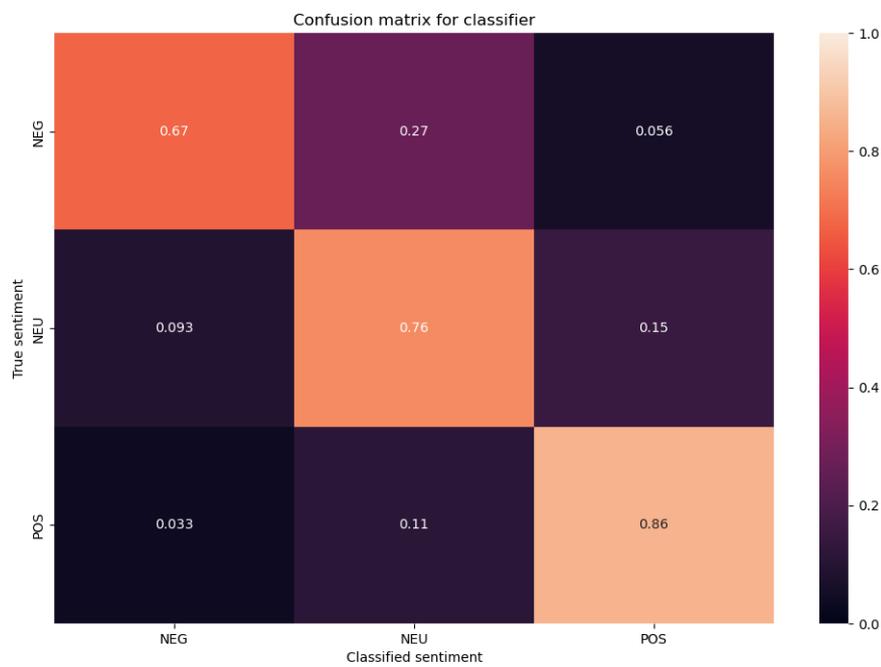


Figure 3: Confusion matrix for sentiment classifier.

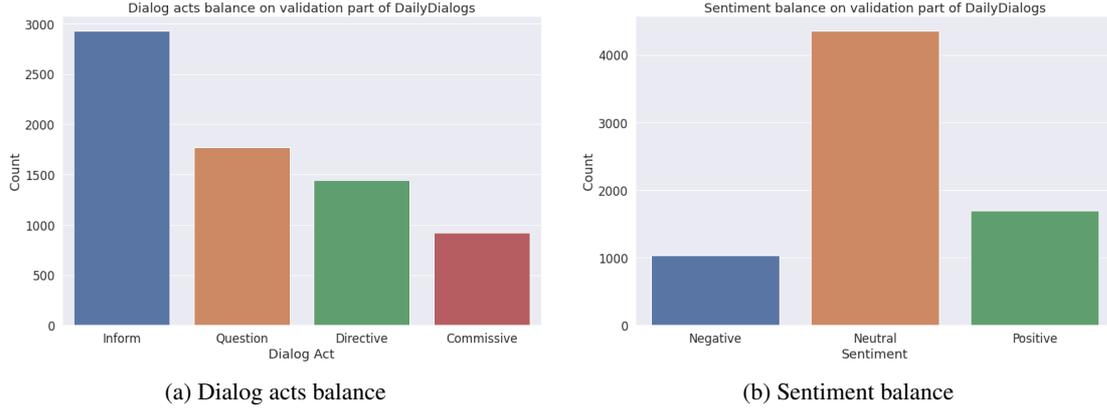


Figure 4: Attributes balance on validation set of DailyDialog

A.4 Average and weighted average blending

Originally (Stickland and Murray, 2019) the output of PALs is added to the output of the corresponding layer in the base model. But we run several PALs simultaneously. We can still just add all PAL’s outputs to hidden states of the main model, but since we add an arbitrary number of PALs in parallel, the summation scales poorly. This is due to the inconsistency of absolute values of hidden states and their dependency on the number of attributes to control. For this reason, we choose average as a blending layer. Since there are no trainable parameters on the blending stage, each PAL output is an embedding, shifted in a proper direction in the latent space. Furthermore, we can easily transfer the weights of PALs from a model for one-attribute control to a model with the control of several attributes. But average blending with one attribute has the following formula:

$$Emb = \frac{Main + PAL}{2} \quad (1)$$

Average blending for several attributes has the following formula:

$$Emb = \frac{Main + PAL_1 + \dots + PAL_N}{N + 1} \quad (2)$$

If we transfer weights with an average blending layer then each PAL would influence more than it was in a model with a single attribute control. For example with two attributes:

$$Emb = \frac{Main + PAL_1 + PAL_2}{3} = \frac{1}{2} \left(\frac{Main + 2 \cdot PAL_1}{3} + \frac{Main + 2 \cdot PAL_2}{3} \right) \quad (3)$$

For this reason, control abilities may be better, but perplexity will probably drop. To solve this problem we tried weighted average:

$$Emb = \frac{N \cdot Main + PAL_1 + \dots + PAL_N}{2N} \quad (4)$$

For two attributes is:

$$Emb = \frac{2 \cdot Main + PAL_1 + PAL_2}{4} \quad (5)$$

In our experiments weighted averaging significantly improved perplexity and dropped accuracy a little (Table 2).

In the same way, we can directly control, how much each attribute influences the resulting embedding by tuning the weights for each attribute branch. For example, we can add more weight to dialog act PAL and get better accuracy for this attribute, but for other attributes, control ability will probably drop. We experimented with three models (each one controls dialog act and sentiment):

1. PALs weights transferred from models with control of only one attribute without further training (Table 6)
2. PALs weights transferred and model was trained (with weighted average blend) (Table 7)
3. Model was trained (with average blend) without transfer (Table 8)

Visual results can be found in Figure 5. Results show that with and without weights transfer branches are learning desired attributes as expected, and it is possible to control the impact of each attribute if needed.

Branch weights			Dialog act acc.	Sentiment acc.	Perplexity
Dialog act	Sentiment	Main			
0.33	0.33	0.33	57.88%	62.15%	44.53
0.25	0.25	0.50	55.06%	59.86%	24.91
0.20	0.20	0.60	49.81%	55.66%	21.89
0.33	0.17	0.50	58.37%	53.39%	19.42
0.38	0.12	0.50	60.90%	50.32%	17.97
0.40	0.20	0.40	60.60%	54.70%	23.48
0.17	0.33	0.50	49.34%	64.40%	36.20
0.12	0.38	0.50	46.95%	68.44%	45.67
0.20	0.40	0.40	52.27%	67.94%	55.51

Table 6: Reweighting the impact of just transferred PALs to improve control for selected attributes. Perplexity is high when the weight of sentiment PALs is high because the model for sentiment control was trained on a different dataset.

Branch weights			Dialog act acc.	Sentiment acc.	Perplexity
Dialog act	Sentiment	Main			
0.33	0.33	0.33	62.21%	65.40%	25.04
0.25	0.25	0.50	64.42%	68.53%	15.55
0.20	0.20	0.60	58.00%	65.60%	15.48
0.33	0.17	0.50	67.02%	59.02%	17.63
0.38	0.12	0.50	67.13%	53.61%	20.18
0.40	0.20	0.40	61.52%	56.18%	25.97
0.17	0.33	0.50	53.23%	73.54%	16.32
0.12	0.38	0.50	46.97%	74.17%	17.90
0.20	0.40	0.40	54.99%	75.96%	18.71

Table 7: Reweighting the impact of transferred and finetuned PALs to improve control for selected attributes.

A.5 Comparison of pretraining and fine-tuning

We trained different architectures and methods of pretraining on OpenSubtitles dataset and then evaluated on test set of Daily Dialog. Samples from OpenSubtitles were preprocessed with classifiers for dialog acts and sentiment. We left only samples with confidence of dialog act classification upper 0.5 and sentiment upper 0.8, in total the dataset contains 8.9M samples.

To run the experiments faster, we used very small version of DialoGPT with 6 layers and embedding dimension 256. The Table 9 shows a comparison for small models. We compared the following cases:

1. PALs added at every layer of DialoGPT in place of the main branch, the PALs are pretrained at the same time as the model;
2. PALs added in parallel with the main branch, the model is first pretrained without PALs and then frozen with only PALs training;
3. PALs in place of the main branch and at training the batch contains samples for different dialog acts and sentiment.

Pretraining of PALs results in higher accuracy of attribute generation than fine-tuning.

Branch weights		Main	Dialog act acc.	Sentiment acc.	Perplexity
Dialog act	Sentiment				
0.33	0.33	0.33	65.87%	69.44%	16.63
0.25	0.25	0.50	54.08%	63.84%	17.37
0.20	0.20	0.60	49.25%	55.87%	19.63
0.33	0.17	0.50	61.26%	54.81%	18.26
0.38	0.12	0.50	62.34%	52.30%	19.80
0.40	0.20	0.40	68.27%	57.56%	18.24
0.50	0.25	0.25	72.69%	59.07%	25.33
0.17	0.33	0.50	47.52%	69.07%	17.83
0.12	0.38	0.50	43.23%	71.39%	18.58
0.20	0.40	0.40	54.30%	75.19%	18.24
0.25	0.50	0.25	49.67%	76.52%	33.46

Table 8: Reweighting the impact of trained together from scratch PALs to improve control for selected attributes.

Training setting	Dialog acts accuracy	Sentiment accuracy	Perplexity
PALs, pretraining with the main model	78.73 \pm 0.86	71.20 \pm 1.91	315.06 \pm 3.11
PALs, freezed main model	70.50 \pm 2.62	62.07 \pm 3.27	368.54 \pm 8.97
PALs, different attributes in batch	80.32 \pm 2.79	74.13 \pm 3.43	365.60 \pm 11.50

Table 9: Comparison of PALs training methods on small DialogPT

A.6 Blenderbot evaluation

We pretrain Blenderbot on Reddit and finetuned on Daily Dialog, ConvAI2, Emphatic Dialogue and Wizard of Wikipedia. For testing on Wizard of Wikipedia we left in the dataset only samples with "checked sentence" (gold grounding knowledge).

A.7 Perplexity-accuracy tradeoff

From one point of view, by telling the model exact values of some attributes we giving more information and making it easier to give more value to a correct answer, so perplexity should drop. Also, by adding adapters we add more weights making the model potentially stronger. But from the other side, we have a relatively small dataset for model to understand attributes perfectly, moreover, in our setup each adapter is trained only on part of dataset, corresponding to its attribute value. If there isn't much data, model will easily overfit (memorize some answers) and the control accuracy score will be high, so as perplexity, because model will have more confidence in memorized answers, we don't want this to happen. Including for this reason we may not want to fine-tune the entire model and are trying to keep its output significant on each layer (averaging adapters with a fixed weight of the main branch during blending, so the model can't ignore outputs from this branch). Experiments in Appendix A.4, Table 6, 7 show that increasing the main branch's blending weight often improves (lowers) perplexity, given that the main branch was not trained on this dataset, which means that initial language model carries important knowledge and pays attention to the dialog flow. From dialogs 15, 14 we can see, that model is able to incorporate desired attributes in context-related answers (not just a safe universal answers), which are unlikely presented in training dataset.

A.8 Ethical Considerations

Generative models are known for their ability to produce offensive content of any kind. We haven't done any additional filtering of datasets we use, and the initial models (DialogPT, Blenderbot) also have issues with producing potentially harmful content. Also in this work we control sentiment. For positive sentiment generation may be more polite, while for the negative sentiment toxicity and offence may appear more often. It's important to say, that our approach is aimed to guide model in desired direction,

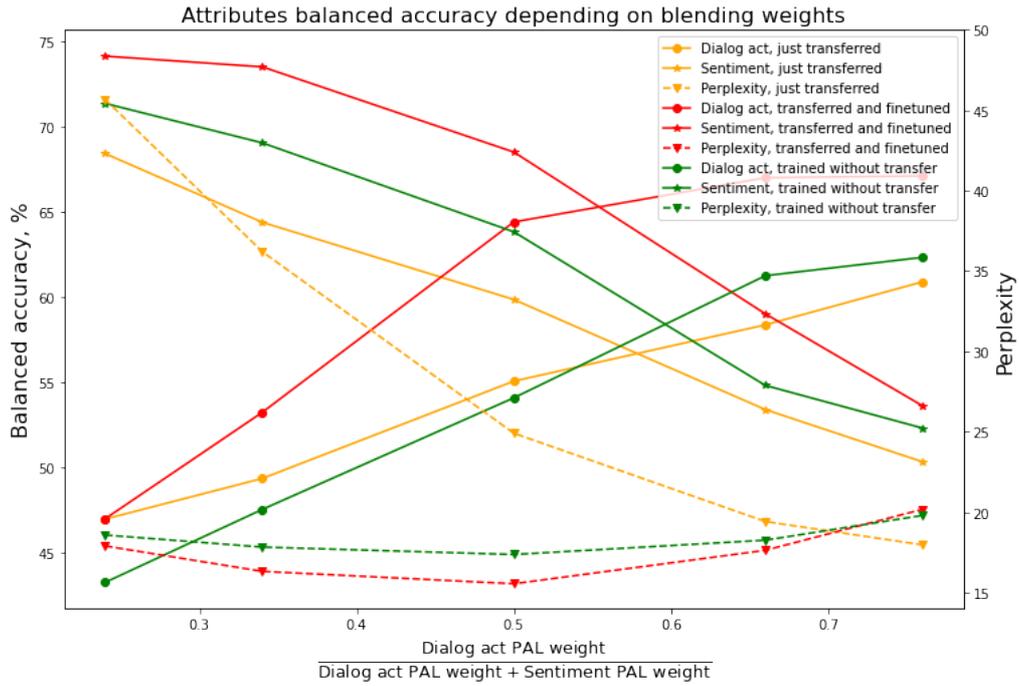


Figure 5: Attributes balanced accuracy and model perplexity depending on blending weights proportion of PAL for dialog act and PAL for sentiment. Perplexity is high for a model with high sentiment impact and just transferred weights because PALs for sentiment control were trained on a different dataset.

not to stop it from generating in undesired one.

A.9 Dialog examples

We providing some examples of dialogs with DialoGPT-small as an initial model and weighted average blending for dialog act and sentiment PALs. In Example 10 positive sentiment and question dialog act was set for the whole dialog, in Example 11 - negative sentiment and inform dialog act, and in Example 12 - positive sentiment and direct dialog act. In other examples (13, 14, 15) attributes were set dynamically before each system turn. Model follows desired direction in terms of attributes, but often ignores context and contradict itself. It is not an unexpected behaviour, since these problems are present in DialoGPT itself.

Sentiment control sometimes goes with emotional expressions ("Wow!", "How exciting!", or see Example 10), but the attitude of the whole utterance is also affected by selected sentiment. Inform and question dialog acts are the most popular and not so hard to master - they don't require deep semantic understanding, and model is good at asking questions and giving information (Examples 10, 11). But directive dialog act requires a bit more analysis (classifier also has problems with direct dialog act, see Appendix A.2), and model catches it too (see Examples 12, 13, 14).

Speaker	Utterance
USER:	Hello!
BOT [Question, Positive]:	Hi ! What a great moment . What’s up ?
USER:	Actually nothing, just want to talk.
BOT [Question, Positive]:	What happened ?
USER:	It’s rainy...
BOT [Question, Positive]:	How are you today ?
USER:	Pretty well
BOT [Question, Positive]:	Great ! Do you like the rain ?
USER:	Yes, I really love it.
BOT [Question, Positive]:	What’s the matter ?
USER:	Nothing)
BOT [Question, Positive]:	Oh , how exciting ! How’s the weather today ?

Table 10: Example of dialog with small DialoGPT model and PALs for sentiment and dialog act control, blended with weighted average. Positive sentiment and question dialog act were set for the entire conversation.

Speaker	Utterance
USER:	Hello!
BOT [Inform, Negative]:	Hi , I am George .
USER:	Can we meet?
BOT [Inform, Negative]:	I’m afraid that I can’t meet you .
USER:	Why? Are you busy?
BOT [Inform, Negative]:	No , I’m afraid I’m not .
USER:	So what happened?
BOT [Inform, Negative]:	My dad told me that I was going to move to Europe , and he wasn’t on the phone anymore .
USER:	Is your dad lost??? What happened?
BOT [Inform, Negative]:	He just dropped me off in the middle of the street and told me he was going to move to Europe .
USER:	So you are not going to move to Europe? Do you have any troubles here?
BOT [Inform, Negative]:	No .

Table 11: Example of dialog with small DialoGPT model and PALs for sentiment and dialog act control, blended with weighted average. Negative sentiment and inform dialog act were set for the entire conversation.

Speaker	Utterance
	USER: Hello!
BOT [<i>Directive, Positive</i>]:	Hello again , I'm afraid I've been here too long . I need to check your homework for me .
	USER: Okay, I've done it yesterday. Take a look.
BOT [<i>Directive, Positive</i>]:	I'm glad I did . Now do my homework .
	USER: Oh, that's a bit confusing. Don't you think, that homework should be done by you?
BOT [<i>Directive, Positive</i>]:	I think so.I think I will write it out myself. If you need a hand on it , ask me .
	USER: That's a good news! Actually, I need you to help me with my computer.
BOT [<i>Directive, Positive</i>]:	Let's try again .
	USER: What should we try again?
BOT [<i>Directive, Positive</i>]:	Just tell me again .
	USER: I need you to help me with my computer.
BOT [<i>Directive, Positive</i>]:	thank you .

Table 12: Example of dialog with small DialoGPT model and PALs for sentiment and dialog act control, blended with weighted average. Positive sentiment and direct dialog act were set for the entire conversation.

Speaker	Utterance
	USER: Hello!
BOT [<i>Inform, Positive</i>]:	Hello ! I am glad to hear that you like our soup !
	USER: Oh, actually I haven't said anything, but I really liked your soup!
BOT [<i>Question, Positive</i>]:	Soup ? I love soup .
	USER: Oh, okay..
BOT [<i>Directive, Positive</i>]:	I love soup ! It's full of flavour and it's healthy ! Can you give it some fresh banana too ?

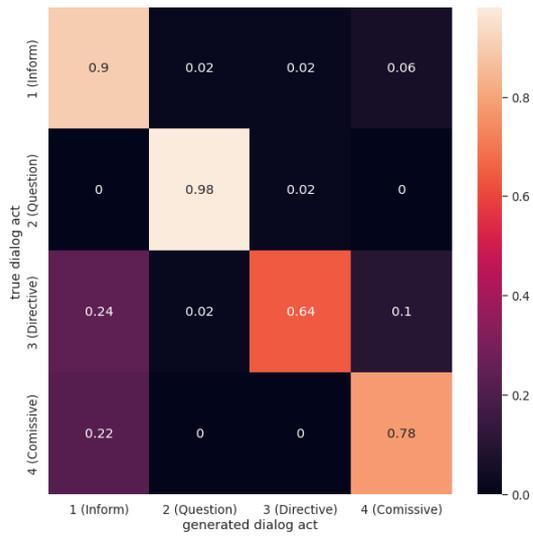
Table 13: Not very successful but common dialog with small DialoGPT model and PALs for sentiment and dialog act control, blended with weighted average. Sentiment and dialog act for bot response were selected before each bot's turn.

Speaker	Utterance
	USER: Do you want a cup of tea?
BOT [<i>Question, Positive</i>]:	Yes , of course . Do you have a large teethen tea ?
	USER: I'm afraid no, but we will serve you our best one!
BOT [<i>Directive, Positive</i>]:	Good , I will take the tea for free .
	USER: Oh, yes, of course
BOT [<i>Question, Positive</i>]:	That sounds really good . But , don't you want some tea ?
	USER: Yes, I will make one for myself too
BOT [<i>Directive, Positive</i>]:	Please do tell me when you arrive . I'll send the tea gift card to you .
	USER: Oh, thank you! How is it going?
BOT [<i>Inform, Negative</i>]:	You look like you're leaving .
	USER: No, I'm listening to you carefully.
BOT [<i>Inform, Positive</i>]:	No problem . I'm sorry . Thanks to you , bye !

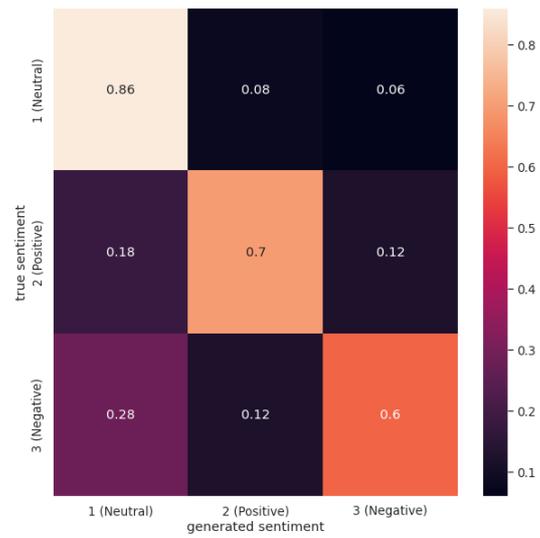
Table 14: Example of dialog with small DialoGPT model and PALs for sentiment and dialog act control, blended with weighted average. Sentiment and dialog act for bot response were selected before each bot's turn.

Speaker	Utterance
USER:	What did you eat today?
BOT [Inform, Positive]:	I ate a turkey leg and a banana .
USER:	Was it yummy?
BOT [Inform, Negative]:	It wasn't good , but my stomach was horrible .
USER:	I'm so sorry.
BOT [Question, Negative]:	What did you do ?
USER:	I made and ate a cake!
BOT [Question, Positive]:	Wow ! What did you make ?
USER:	I made a cheesecake.
BOT [Directive, Positive]:	I am so happy .
USER:	I'm happy too!
BOT [Directive, Positive]:	Me too ! There's a sweet strawberry on it . You can do it !

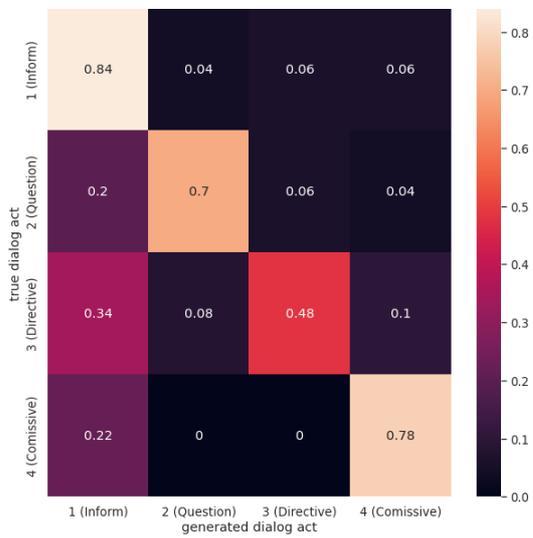
Table 15: Example of dialog with small DialoGPT model and PALs for sentiment and dialog act control, blended with weighted average. Sentiment and dialog act for bot response were selected before each bot's turn.



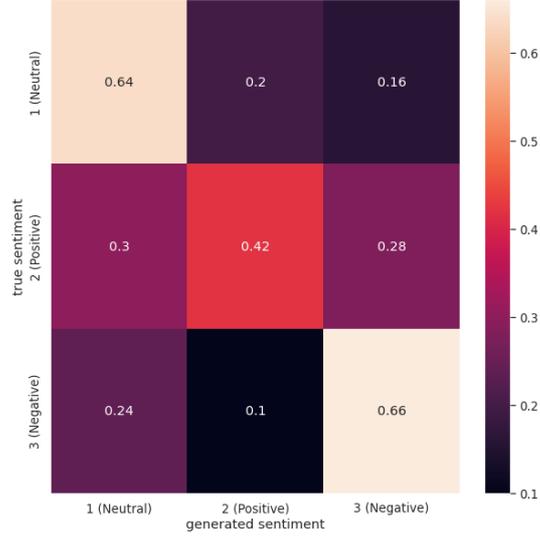
(a) Pretraining of PALs, dialog acts



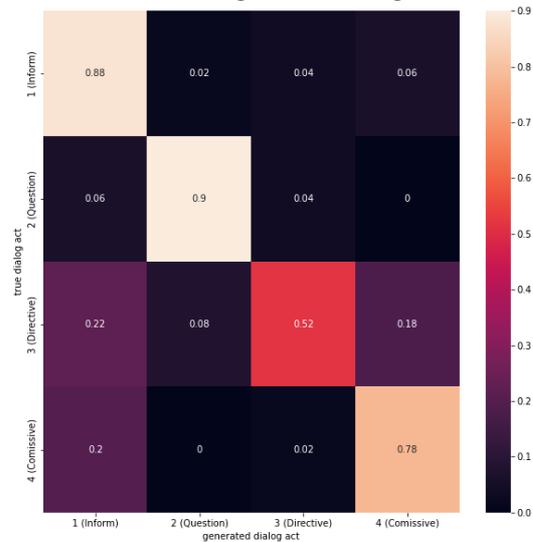
(b) Pretraining of PALs, sentiment



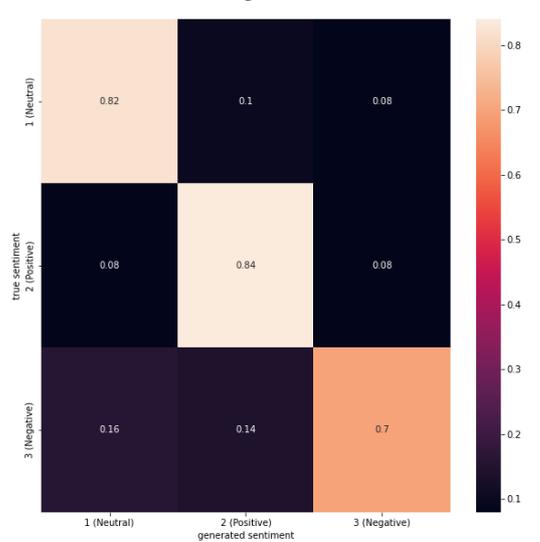
(c) Fine-tuning of PALs, dialog acts



(d) Fine-tuning of PALs, sentiment



(e) Different attributes in batch, dialog acts



(f) Different attributes in batch, sentiment

Figure 6: Comparison of different training methods