Controllable Multi-attribute Dialog Generation with PALs and Grounding Knowledge

Anonymous ACL submission

Abstract

Today, neural language models are commonly employed for generation of natural like responses in dialogue system. The main issue that limits wide adoption of neural generation is related to poor predictability of responses in terms of a content, as well as dialogue 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 dialogue generation. We compared a number of methods for training and blending representations produced by PALs combined with Dialo-GPT 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 less numbers of 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 dialogue generation given grounding knowledge on Wizard of Wikipedia.

1 Introduction

Majority of open-domain dialogue 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 predefined output response and transition to the next state of the dialogue. 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 dialogue 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 a script is defined as pairs of adjacent dialog acts then a generative model conditioned on grounding knowledge about entities found in the dialogue context, are able to generate all the bot utterances in the script without retrieval of hand-written responses.

Controllable generative models have been an active area of research over last years. Models (Zhao et al., 2017), (See et al., 2019), (Zhang et al., 2018) control only one attribute of the generated response (dialog act, response relatedness or specificity). CRAYON (Hu et al., 2021), which inserts control embeddings into LSTM architecture, mixes several attributes in the response but requires pre-training of the whole model. 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 attribute, such as dialog act or sentiment.

Trained models, train and inference code and data to test the quality of models published in Open Source under the Apache 2.0 license *(anonymized link, see submitted archive)*. 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;

- 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, one of them was proposed by Adapter bot (Madotto et al., 2020) model which has an option of switch between different attributes without changes in initial model by adding adapter layers. Hyperformer (Karimi 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. The limitation of hyperformer, adapter bot and (Xie and Pu, 2021) is inability to mix different attributes in one response (e.g., topic and emotion). CRAYON (Hu et al., 2021) is the model for multi-attribute response generation (response length, question/statement, sentiment, response relatedness). Our models generates responses for more dialog acts (not only question/statement) and does not require training the base model.

Most of generative models, which do not use external knowledge, are capable of producing grammatically correct and natural responses given the dialogue 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 dialogue 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 grounding 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 dialogue generation quality (perplexity) we used Wizard of Wikipedia dataset (Dinan et al., 2018).

One of the approaches to control object attributes is to learn proper shifts in latent space (Hu et al., 2021). 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).

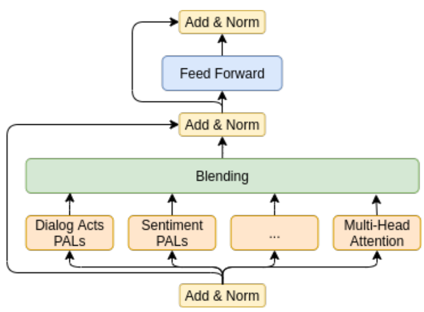


Figure 1: Blending of PALs and multi-head attention of Transformer hidden representations for every token.

To control several attributes simultaneously, we decided to add a PAL for each attribute and run

Control	Blend	Train dataset	Dialog act acc.	Sentiment acc.	Perplexity	Opt. steps	Trainable par.
No control	-	DailyDialogs	25.20 ±0.21	33.41 ±0.15	**15.19** ±1.58	2000	117M
Dialog acts	average	DailyDialogs	**63.74** ±0.32	42.83 ±0.27	15.93 ±0.12	10000	36M
Dialog acts	dense	DailyDialogs	45.27 ±5.26	40.15 ±1.22	22.36 ±0.85	5000	49M
Sentiment	average	ScenarioSA	33.40 ±0.16	**72.09** ±4.06	92.98 ±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 ±2.22	69.19 ±1.10	17.12 ±0.40	5000	63M
dense	no	61.65 ±1.02	67.10 ±1.38	22.07 ±0.39	5000	84M
dense & average	no	61.36 ±1.40	68.12 ±0.69	15.51 ±0.13	5000	77M
average	yes	**65.62** ±2.04	66.04 ±0.25	17.74 ±0.75	5000	63M
weighted average	yes	63.20 ±1.09	69.05 ±0.42	15.65 ±0.09	5000	63M
dense	yes	60.83 ±1.35	67.80 ±3.37	21.34 ±0.40	5000	84M
dense & average	yes	62.76 ±0.70	**70.03** ±2.04	15.69 ±0.19	5000	77M
dense & average, only blend	yes	60.19 ±0.76	67.47 ±0.90	**15.30** ±0.05	10000	14M

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

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.

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.

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 are capable to be finetuned on the target dataset.

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 attibute branches were blended with the dense layer and then added to the main branch of the base model.

4 Experiments and results

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).

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.

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

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.

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 Dialy 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 dialogue generation given grounding knowledge (GK) on Wizard of Wikipedia dataset.

We compared controllable Blenderbot with CRAYON (Hu et al., 2021) 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*.

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. *arXiv preprint arXiv:1811.01241*.

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*.

Zhe Hu, Zhiwei Cao, Hou Pong Chan, Jiachen Liu, Xinyan Xiao, Jinsong Su, and Hua Wu. 2021. Controllable dialogue generation with disentangled multi-grained style specification and attribute consistency reward.

Rabeeh Karimi Mahabadi, Sebastian Ruder, Mostafa Dehghani, and James Henderson. 2021. Parameter-efficient multi-task fine-tuning for transformers via shared hypernetworks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 565–576, Online. Association for Computational Linguistics.

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.

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.

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.

Zeqiu 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. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 133–147.

Heng-Da Xu, Xian-Ling Mao, Zewen Chi, Fanshu Sun, Jingjing Zhu, and Heyan Huang. 2021a. Generating informative dialogue responses with keywords-guided networks. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 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*.

Haolan Zhan, Lei Shen, Hongshen Chen, and Hainan Zhang. 2021. Colv: A collaborative latent variable model for knowledge-grounded dialogue generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 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. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1108–1117, Melbourne, Australia. Association for Computational Linguistics.

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*.

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. DailyDialog 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. onsequently 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%, the confusion matrix is in Figure 3.

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 **??**. 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 averages 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.

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

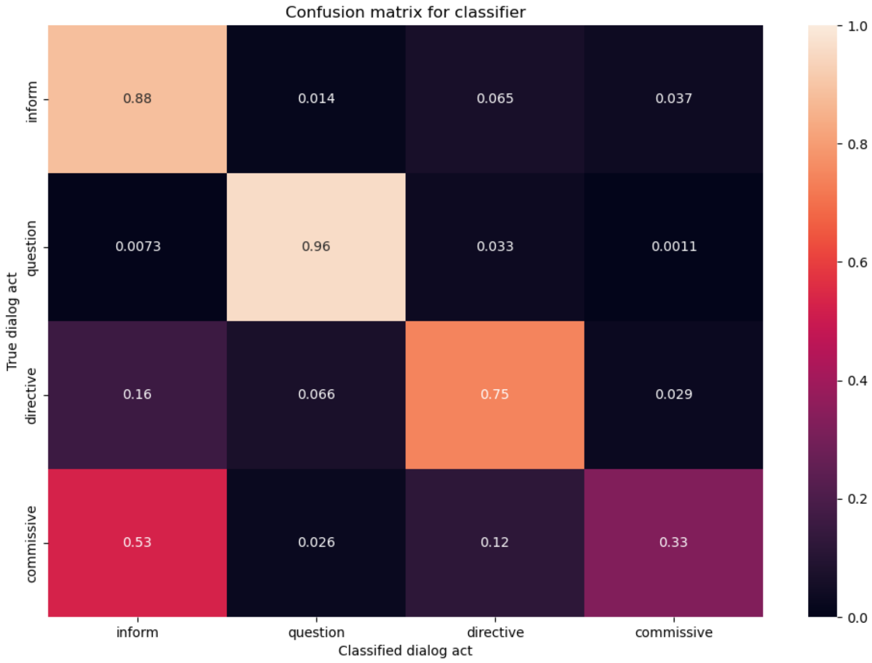


Figure 2: Confusion matrix for dialog acts classifier.

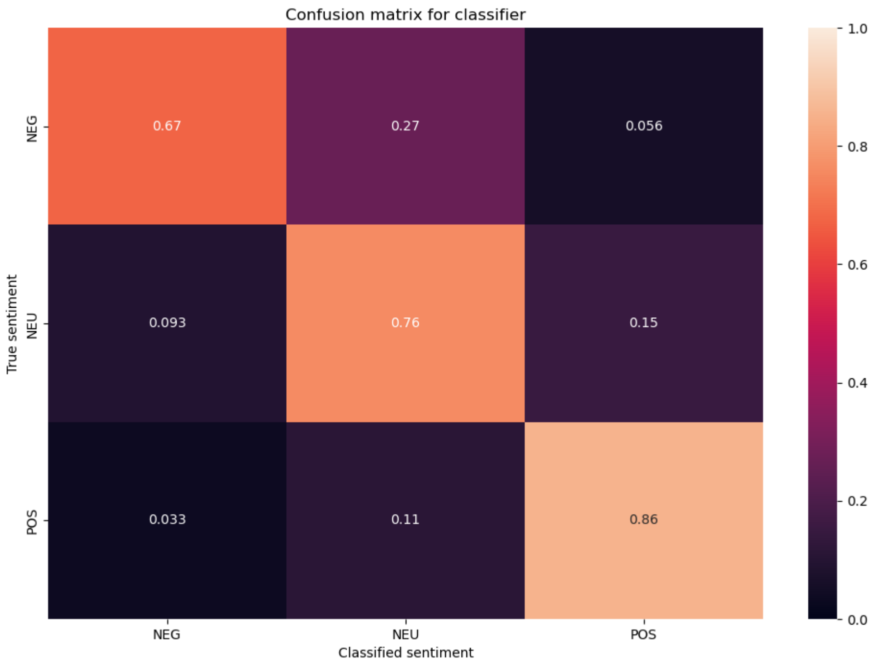
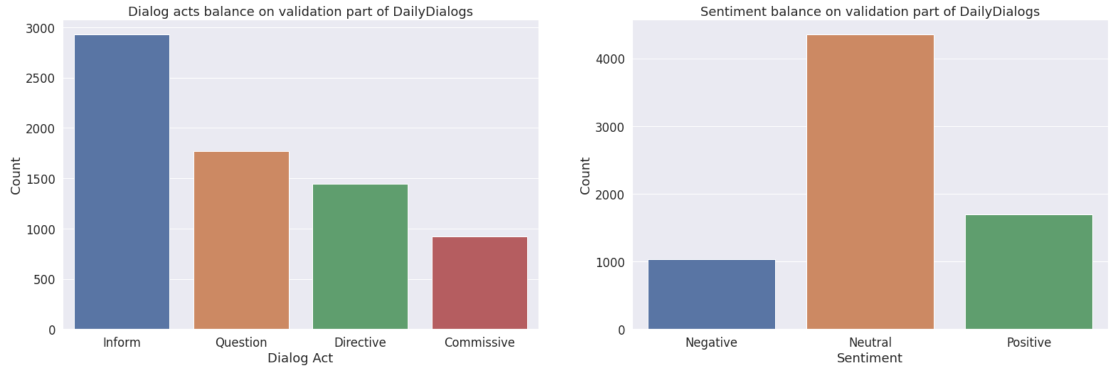


Figure 3: Confusion matrix for sentiment classifier.



(a) Dialog acts balance

(b) Sentiment balance

Figure 4: Attributes balance on validation set of DailyDialog

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.

has the following formula:

$$Emb = \frac{Main + PAL}{2} \tag{1}$$

Average blending for several attributes has the following formula:

$$Emb = \frac{Main + PAL_1 + \cdots + PAL_N}{N + 1} \tag{2}$$

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:

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

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

$$Emb = \frac{N \cdot Main + PAL_1 + \cdots + PAL_N}{2N} \tag{4}$$

For two attributes is:

$$Emb = \frac{2 \cdot Main + PAL_1 + PAL_2}{4} \tag{5}$$

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	Sentiment	Perplexity
Dialog act	Sentiment	Main	acc.	acc.	
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	Sentiment	Perplexity
Dialog act	Sentiment	Main	acc.	acc.	
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 freezed 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.

The Figure 6 contains confusion matrices for dialog acts and sentiment of different training settings. Pretraining of PALs results in higher accuracy of attribute generation than fine-tuning.

A.6 Blenderbot evaluation

Experiments with DialoGPT-small (more details in Appendix A.5) showed that pretraining of the model with PALs result in higher control accuracy than training only PALs when the main model is freezed, therefore We pretrain Blenderbot on Reddit and finetuned on Daily Dialog, ConvAI2, Emphatetic 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 Limitations and future work

Our results have limitations with respect classifiers quality for both dialog acts and sentiment (more details in Appendix A.2). Another one is increasing a number of parameters for adding new attribute. Furthermore, we utilized up to two attributes with more attributes quality can be affected. One of the risks for generative models produce harm text, probability of which reduces compared to control-

Branch weights			Dialog act	Sentiment	Perplexity
Dialog act	Sentiment	Main	acc.	acc.	
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 ± 0.86	71.20 ± 1.91	**315.06** ± 3.11
PALs, freezed main model	70.50 ± 2.62	62.07 ± 3.27	368.54 ± 8.97
PALs, different attributes in batch	**80.32** ± 2.79	**74.13** ± 3.43	365.60 ± 11.50

Table 9: Comparison of PALs training methods on small DialoGPT

lable generative models, but is not excluded. Moreover, generative models can be used unethically when a certain quality of generation is achieved.

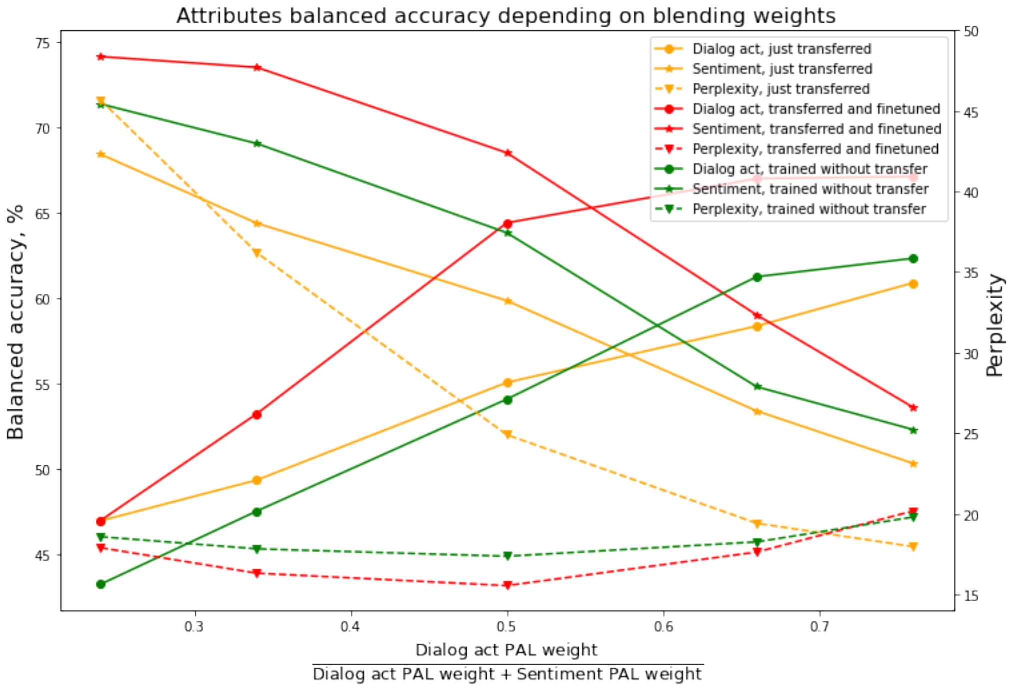
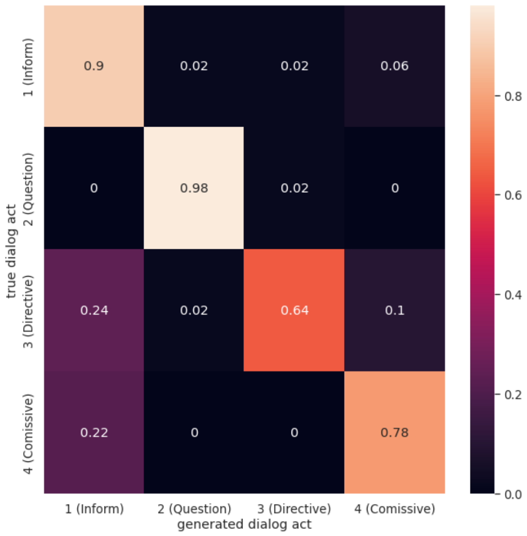
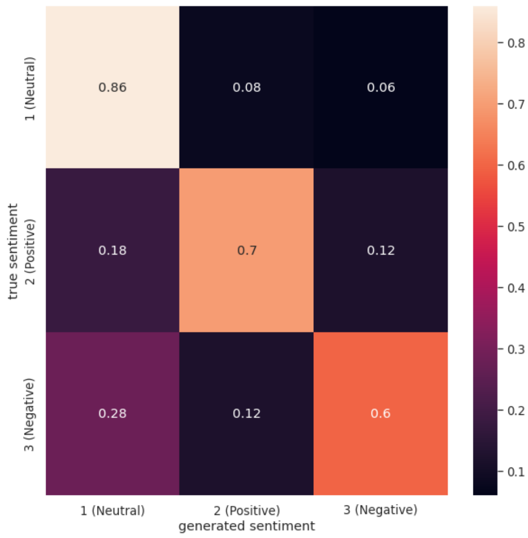


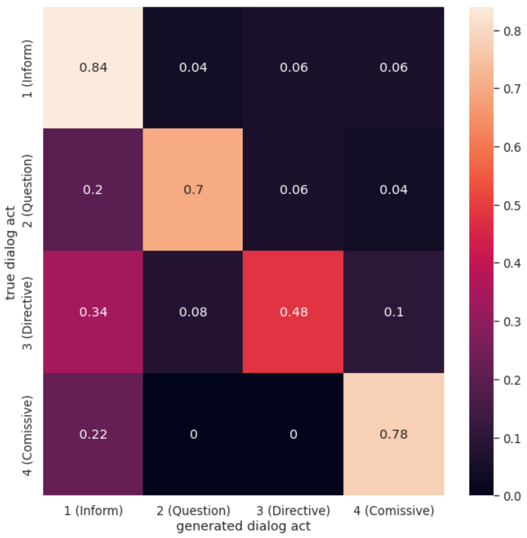
Figure 5: Attributes balanced accuracy and model perplexity depending on blending weights proportion of PAL for dialog act and PAL for sentiment. Perplexity is high for a model with high sentiment impact and just transferred weights because PALs for sentiment control were trained on a different dataset.



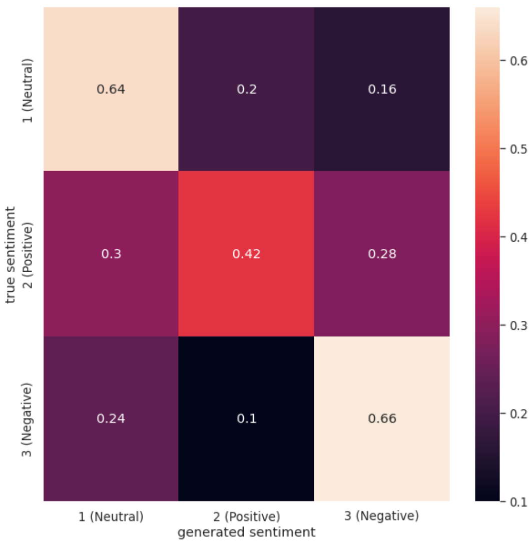
(a) Pretraining of PALs, dialog acts



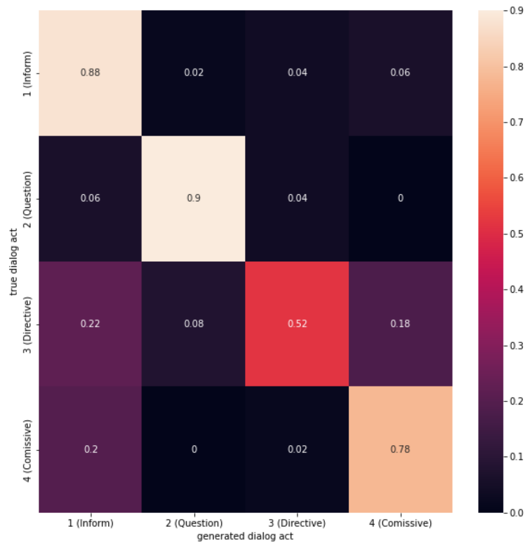
(b) Pretraining of PALs, sentiment



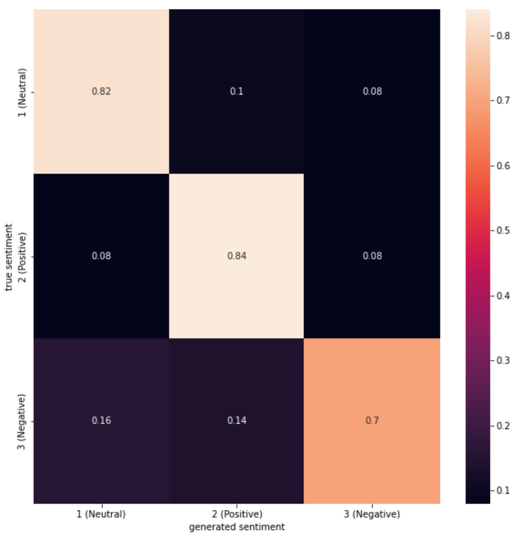
(c) Fine-tuning of PALs, dialog acts



(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