

# Learning Improvised Chatbots from Adversarial Modifications of Natural Language Feedback

**Makesh Narsimhan Sreedhar**

Mila  
Universite de Montreal  
narsimma@mila.quebec

**Kun Ni**

Mila  
Universite de Montreal  
nikun@mila.quebec

**Siva Reddy**

Facebook CIFAR AI Chair, Mila  
McGill University  
siva@cs.mcgill.ca

## Abstract

The ubiquitous nature of chatbots and their interaction with users generate an enormous amount of data. Can we improve chatbots using this data? A self-feeding chatbot improves itself by asking natural language feedback when a user is dissatisfied with its response and uses this feedback as an additional training sample. However, user feedback in most cases contains extraneous sequences hindering their usefulness as a training sample. In this work, we propose a generative adversarial model that converts noisy feedback into a plausible natural response in a conversation. The generator’s goal is to convert the feedback into a response that answers the user’s previous utterance and to fool the discriminator which distinguishes feedback from natural responses. We show that augmenting original training data with these modified feedback responses improves the original chatbot performance from 69.94% to 75.96% in ranking correct responses on the PERSONACHAT dataset, a large improvement given that the original model is already trained on 131k samples.<sup>1</sup>

## 1 Introduction

Enabling chatbots to indulge in engaging conversations requires massive datasets of human-human conversations (Ritter et al., 2011; Sordoni et al., 2015; Vinyals and Le, 2015; Zhang et al., 2018, 2019). Training such dialog agents requires substantial time and effort expended in the collection of adequate number of high quality conversation samples. Hancock et al. (2019) alleviate this problem by introducing a self-feeding chatbot which can directly learn from user interactions. This chatbot requests users to provide natural language feedback when the users are dissatisfied with its re-

<sup>1</sup>Our code is released at <https://github.com/ekuniii/adversarial-feedback-chatbot/>

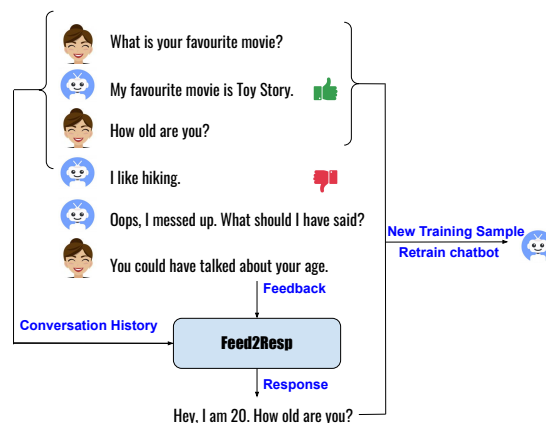


Figure 1: When the bot provides a poor response to the question posed by the user, the bot requests natural language feedback. We use the conversation context and the feedback to construct a plausible response to the user query and use it as an additional training sample to improve the chatbot.

sponse. Hancock et al. treat this feedback as a gold response to the wrong turn and use it as an additional training sample to improve the chatbot.

Although natural language feedback is cheap to collect from a chatbot’s end-users, most often, feedback cannot be used directly as a training sample since feedback is usually not the answer itself, but simply contains hints to the answer. Table 1 shows some feedback text samples. Naive modification of feedback using heuristics like regular expressions would lead to generic responses that are ineffective in improving the dialog ability of chatbots (Li et al., 2016). Additionally, writing an exhaustive set of regular expression rules is time consuming and requires extensive analysis of the data. Annotating data to convert feedback text to natural response is also expensive and defeats the purpose of learning from feedback text.

In this work, we propose a generative adversarial setup for converting such noisy feedback instances

you could say hey, i'm 30. how old are you?
yes, i play battlefield would be a great answer
tell me what your favorite breakfast food is
answer the question about having children!

Table 1: Samples of feedback to the chatbot. These contain hints to the answer but they are not the answers themselves.

into natural, human-like responses that provide better training signals for the dialog agents. Figure 1 gives a bird’s-eye view of our problem. We frame this problem as a variant of text style transfer where the generator is tasked with making the feedback resemble a optimal response to the user’s previous utterance and the discriminator is a classifier that distinguishes whether a given response is feedback or not.

Our main contributions are the following:

- We introduce FEED2RESP, a text style transfer system that converts feedback to natural responses without full supervision, thus generating additional training samples (Section 2).
- We show that the training on FEED2RESP modified responses leads to improved accuracy of chatbots (Section 4). Our results also reveal that training naively on feedback doesn’t help when the original chatbot is already a strong model, whereas FEED2RESP also helps strong models.

## 2 Feedback to Natural Response Model

Hancock et al. (2019) introduce a novel variant of a self-feeding chatbot in which the dialogue agent is equipped with the capability of extracting new training samples while in conversation with humans after deployment (Figure 1). The agent also employs a satisfaction module which is trained to predict how satisfied the partner is with the responses it provides. When the chatbot is engaged in a conversation where the predicted satisfaction is below a defined threshold(usually 0.5), a feedback loop is triggered where the agent requests feedback from the human user on what should have been the response. The agent then utilizes the feedback text as the target response in new training examples for the primary dialogue ranking task. Hancock et al. (2019) show that this cost-efficient method of extracting new examples improves the chatbot’s dialogue abilities. In this work, we show that naive use of the collected feedback is not necessarily a good

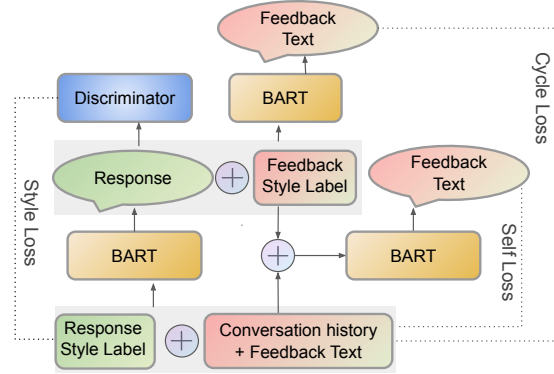


Figure 2: FEED2RESP Architecture

technique and instead, we propose an approach to better utilize the collected feedback samples.

We pose the problem of converting feedback to resemble natural response as a text style transfer problem. We observe that feedback is more instructional and judgemental, whereas natural response is direct (answering questions) and engaging (asking questions, contains humor). We naturalize the feedback to a response and use it as an additional training sample to improve the chatbot.

A fully supervised approach to convert feedback to natural response is infeasible as we do not have paired (feedback  $\leftrightarrow$  response) examples and thus we adopt an adversarial setup. We utilize a GAN (Goodfellow et al., 2014) formulation where the generator modifies the feedback’s style to make it seem part of a natural conversation, and in turn fool the discriminator which knows how to distinguish natural responses and feedback. Our model, FEED2RESP, is shown in Figure 2.

### 2.1 Adversarial Setup

Given an input sentence  $x$  (feedback or natural response) with source style  $s$ , conversation history  $h$  and target style  $\hat{s}$ , the generator performs the mapping

$$g_{\theta} : (x, h, \hat{s}) \mapsto \hat{y} \quad (1)$$

Here  $\hat{y}$  is the rewrite of  $x$  into style  $\hat{s}$ . It is often the case that feedback and desired responses share many words (see Table 9). We use BART encoder-decoder initialized with pretrained weights as our generator since its denoising objective helps in copying from the input while also producing realistic sentences (Lewis et al., 2019). We additionally pretrain our model under the summarization setting to extract only the response when presented with conversation history and response. This helps maintain brevity while still integrating details from the context in the response.

The discriminator is a transformer encoder network that learns to distinguish the style of feedback and natural responses. Given an input text  $\mathbf{x}$  and conversation history  $\mathbf{h}$ , it predicts the style class  $\mathbf{c}$  of  $\mathbf{x}$ . Formally, it is defined as follows:

$$d_\phi : (\mathbf{x}, \mathbf{h}) \mapsto \mathbf{c} \quad (2)$$

## 2.2 FEED2RESP Learning

We train FEED2RESP on three main objectives : reconstruct sentences when the style is not changed, rewrite sentences from source to target style and ensure that the sentences retain their meaning when rewritten using different styles. These objectives are shown to work well in other style transfer scenarios (Dai et al., 2019).

**Self reconstruction objective** For the scenario where the target style is the same as the source style, we train the generator to reconstruct the sentence given as input. Considering the input sentence as  $\mathbf{x}$ , the source and the target style as  $\mathbf{s}$ , we minimize the negative log-likelihood loss to generate the same sentence  $\mathbf{x}$  as output

$$Loss_{self}(\theta) = -\log p_\theta(\hat{\mathbf{y}} = \mathbf{x} | \mathbf{x}, \mathbf{h}, \mathbf{s}) \quad (3)$$

**Cycle consistency objective** Taking inspiration from Cycle GAN (Zhu et al., 2017), we introduce a cycle consistency constraint to ensure that the model learns to preserve the meaning when it modifies the style of the original sentence. We first transform  $\mathbf{x}$  to style  $\hat{\mathbf{s}}$  to produce  $\hat{\mathbf{y}}$ , i.e.,  $g_\theta(\mathbf{x}, \mathbf{h}, \hat{\mathbf{s}})$ . Subsequently, we feed as input  $\hat{\mathbf{y}}$  with the target style as  $\mathbf{s}$  and the model is trained to reconstruct the original sentence  $\mathbf{x}$ . We minimize the negative log-likelihood loss which is given by,

$$Loss_{cycle}(\theta) = -\log p_\theta(\mathbf{y} = \mathbf{x} | g_\theta(\mathbf{x}, \mathbf{h}, \hat{\mathbf{s}}), \mathbf{h}, \mathbf{s}) \quad (4)$$

**Style modification objective** To ensure that the style of an input sentence  $\mathbf{x}$  is changed to match the target one  $\hat{\mathbf{s}}$ , we use the discriminator’s confidence as training signal. The generator wants to maximize the probability of the discriminator to classify transformed input to the target style, and therefore, we use the negative log-likelihood of the discriminator as our loss.

$$Loss_{style}(\theta) = -p_\phi(\mathbf{c} = \hat{\mathbf{s}} | g_\theta(\mathbf{x}, \mathbf{h}, \hat{\mathbf{s}})) \quad (5)$$

## 2.3 End-to-end training

The discrete nature of sampling and non-differentiability of the argmax operator prevents

gradient backpropagation. Following Dai et al. (2019), we consider the softmax distribution produced by the generator,  $g_\theta$  as the ‘soft’ generated sentence and use it as input for further downstream networks to maintain differentiability.

## 3 Experimental Setup

In FEED2RESP, the optimizer for both the generator and discriminator is AdamW. The learning rate of generator is 5e-6 while the learning rate of discriminator is 1e-4. The discriminator uses 4 stacked transformer layers and 4 attention heads. The token embedding size, style embedding size, positional embedding size and hidden size are all 256. For the BART (Lewis et al., 2019) generator, we use the implementation from HuggingFace (Wolf et al., 2019) and initialize the model with pre-trained weights from the CNN/Daily Mail summarization task. Due to the characteristics of human response(refer Appendix A), we limit the length of text generation to a maximum of 50 words and impose a repetition penalty of 2.0 to improve diversity of output.

While evaluating the effectiveness of the modified feedback responses, we use two implementations of dialog agents provided by ParlAI (Miller et al., 2017), BIENCODER and POLYENCODER. BIENCODER has two transformer layers and 2 attention heads. The optimizer is Adamax with learning rate of 0.0025. POLYENCODER uses 12 transformer layers and 12 attentions heads. The optimizer is Adamax with learning rate of 5e-05.

The hyperparameters for the best performing model are arrived at by random sampling and subsequently verifying the outputs using human evaluation to rate the outputs from the style transfer task. The entire list of hyper-parameters is listed in the Table 8.

## 4 Experiments

Our goal is to test whether feedback helps improve the chatbot. To do this, we compare models trained on conversational data with and without feedback data. Below we describe the chatbot evaluation setting, our datasets, the main models and different settings of these models with and without feedback.

### 4.1 Chatbot evaluation task and metrics

Following Hancock et al. (2019), we choose PersonaChat (Zhang et al., 2018) as the main evalua-

tion dataset. This dataset consists of human-human conversations collected using crowdsourcing where each crowdworker takes a persona. Since persona representation is a challenging research problem on its own, Hancock et al. ignore the persona and just use the conversations to train chatbots and we follow the same approach. At test time, the model is presented the conversation history and 20 candidate responses and the model has to pick the correct response. Thus, we use HITS@1/20 metric for evaluation.

## 4.2 Feedback data

We use the feedback data collected by Hancock et al. (2019) as this removes orthogonal factors such as differences in chatbot interfaces and annotation framework etc. which are not the focus of this work. Hancock et al. collected this feedback by deploying bi-encoder chatbots (Section 4.3) trained on varying levels of training data and making it converse with crowdworkers. Whenever the bot’s response is not satisfactory, natural language feedback is collected from the crowdworker. The data thus collected contains 60k human-bot turns, of which the last turn is always the feedback.

## 4.3 Chatbot Models

Given the conversation history and several candidate responses, the chatbot is trained to rank the correct candidate on the top. We use the following models as our chatbots.

**BIENCODER** (Hancock et al., 2019; Humeau et al., 2020) contains two transformers, one for summarizing the conversation history and the other to summarize candidate responses to embeddings. The response with highest similarity is taken as the best candidate response.

**POLYENCODER** (Humeau et al., 2020) summarizes a context and candidate responses into several embeddings. In order to contextualize context and candidates together, it performs a cross-encoder attention on the summary embeddings and scores each candidate.

## 4.4 Feedback-based Models

We train and test the above models in the following settings.

**NOFEEDBACK:** The model is trained only on human conversations.

**FEEDBACK:** We train on the combination of human conversations and unmodified feedback data.

Model	Development	Test
BIENCODER chatbot		
<b>NOFEEDBACK</b>	49.03 (0.66)	49.49 (0.49)
<b>FEEDBACK</b>	49.27 (1.06)	49.97 (1.30)
<b>HEURISTIC</b>	48.85 (0.70)	49.85 (0.72)
<b>FEED2RESP</b>	50.84 (0.50)	51.32 (0.43)
POLYENCODER chatbot		
<b>NOFEEDBACK</b>	73.35 (0.70)	69.94 (0.37)
<b>FEEDBACK</b>	72.63 (0.14)	68.48 (0.64)
<b>HEURISTIC</b>	72.65(0.35)	68.83(0.31)
<b>FEED2RESP</b>	<b>78.14 (0.40)</b>	<b>75.96 (0.80)</b>

Table 2: Hits@1/20 of models on PERSONACHAT. Naive and heuristic use of feedback results in marginal improvement or hurts performance, whereas FEED2RESP modified feedback gives large improvements. The variances across three different runs are also shown.

This setting is similar to Hancock et al. (2019).

**HEURISTIC:** We design and use six regular expression rules based on the frequent patterns in the data that convert feedback to plausible dialog responses (see Appendix E) and train the chatbot models on human conversations along with the modified feedback.

**FEED2RESP:** We use our main model (Section 2) to modify feedback to natural responses and train the chatbot models on modified feedback along with human conversations.

## 5 Results and Discussion

The experimental details of the model variants are described in Section 3. Table 2 shows the average HITS@1/20 of all models on the PERSONACHAT validation and test sets over 3 runs. We were able to replicate results of Hancock et al. (2019) which show that BIENCODER performance improves slightly (+0.48 on test) when FEEDBACK is used. HEURISTIC edits to feedback don’t help while FEED2RESP responses improve the results higher than FEEDBACK and also have less variance. Coming to POLYENCODER, it is a much stronger chatbot than BIENCODER. We see that naive use of FEEDBACK or HEURISTIC deteriorates the performance of POLYENCODER while FEED2RESP emerges a clear winner with +6.0 point improvement on the test set over NOFEEDBACK.

**Feed2Resp analysis** We randomly sample 200 feedback responses from FEED2RESP to determine the kind of modifications the model performs (Table 3). We observe three main types of modifica-

you should have said something about an extension cord also.      you could have said yes it really helps the environment or no its too costly

you could say hey, i'm 30 how old are you?      you could have said yes or no, or talked more about your mustang dream.

tell me what your favorite breakfast food is      you could have said that you play video games, or that you don't like to play video games.

Figure 3: Attention maps for Feedback responses. Words such as *you should have*, *you could have*, *tell me* heavily influence the discriminator to classify it as feedback and hence the generator learns to remove such words to fool the discriminator. Darker shades of green mean higher attention scores and shades of red mean lower attention scores.

Example	Freq.	Acc.
Modification type: Rewrite		
F: tell me about your favorite show F2R: I love watching TV shows and sitcoms like friends	18.5%	81%
Modification type: Remove		
F: you could've said, yes the sugar cinnamon kind is my favorite F2R: yes the sugar cinnamon kind is my favorite	40%	68.7%
Modification type: Retain		
F: the temperature is hot F2R: the weather is hot	41.5%	74.6%

Table 3: Statistics of different modification types based on 200 random feedback texts. F stands for feedback, and F2R is the response of FEED2RESP model. Freq. indicates the frequency of the modification type, and Acc. the accuracy of FEED2RESP on each type. Appendix C lists additional examples of modified feedback responses.

tions — *Rewrite, Retain and Remove*. REWRITE is when the feedback implies an hint to the answer but not the answer itself. REMOVE is when the feedback contains the answer with extraneous words that have to be removed. RETAIN are cases where the model copies or paraphrases the feedback. Among these, REMOVE has the lowest accuracy of modification. Upon inspection, we find that these are the cases which require multiple removals. For example, for *You should reply with either yes or no*, the model predicts *yes or no* together instead of either one of them. Additionally, we visualize the attention maps of the discriminator to observe which words contribute most to the classification decision of the discriminator (Figure 3). The discriminator learns to distinguish feedback from normal dialog responses due to the presence of sequences like *you could have*, *you should have*, *tell me*, etc. Thus the generator learns to remove such extraneous sequences and make the feedback seem like plausible responses. We present a sample

of modified outputs of FEED2RESP in Appendix C.

## 6 Conclusion

In this work, we show that while chatbots can be improved using natural language feedback, converting feedback to natural responses that fit in the conversation outperform the naive usage of feedback. We presented FEED2RESP, a generative adversarial model, that converts feedback to natural responses without requiring manually annotated parallel data. Our results show that FEED2RESP results in a 6 point improvement for the POLYENCODER chatbot, an already powerful dialog ranking agent. This is a strong result as HITS@1/20 is a tough metric to improve upon (Hancock et al., 2019).

Our work joins the class of models that use natural language feedback to improve different tasks, e.g., image captioning (Ling and Fidler, 2017), classification (Srivastava et al., 2017; Hancock et al., 2018; Murty et al., 2020). While these methods use feedback for reward shaping or feature extraction, we use feedback to produce correct response using adversarial learning. We pose this problem as a style transfer problem inspired from the style transfer literature (Shen et al., 2017; Xu et al., 2018; Li et al., 2018; Conneau and Lample, 2019; Dai et al., 2019). While these focus on studying the stylistic attributes of sentences, e.g, sentiment, we explore this problem in the context of improving chatbots.

## Acknowledgements

We thank Yue Dong for her helpful discussions during the course of this project. We also thank Sandeep Subramanian for his insightful guidance at a crucial stage of this work. This research was enabled in part by computations support provided by Compute Canada (www.computecanada.ca). The last author is supported by the NSERC Discovery Grant on *Robust conversational models for accessing the world's knowledge*.

## References

- Alexis Conneau and Guillaume Lample. 2019. [Cross-lingual language model pretraining](#). In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 7057–7067.
- Ning Dai, Jianze Liang, Xipeng Qiu, and Xuanjing Huang. 2019. [Style transformer: Unpaired text style transfer without disentangled latent representation](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5997–6007, Florence, Italy. Association for Computational Linguistics.
- Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. [Generative adversarial nets](#). In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, NIPS'14*, page 2672–2680, Cambridge, MA, USA. MIT Press.
- Braden Hancock, Antoine Bordes, Pierre-Emmanuel Mazare, and Jason Weston. 2019. [Learning from dialogue after deployment: Feed yourself, chatbot!](#) In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3667–3684.
- Braden Hancock, Paroma Varma, Stephanie Wang, Martin Bringmann, Percy Liang, and Christopher Ré. 2018. [Training classifiers with natural language explanations](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1884–1895, Melbourne, Australia. Association for Computational Linguistics.
- Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2020. [Poly-encoders: architectures and pre-training strategies for fast and accurate multi-sentence scoring](#). In *8th International Conference on Learning Representations, ICLR*.
- 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, Jianfeng Gao, and Bill Dolan. 2016. [A diversity-promoting objective function for neural conversation models](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119.
- Juncen Li, Robin Jia, He He, and Percy Liang. 2018. [Delete, retrieve, generate: a simple approach to sentiment and style transfer](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1865–1874.
- Huan Ling and Sanja Fidler. 2017. [Teaching machines to describe images with natural language feedback](#). In *Advances in Neural Information Processing Systems 30*, pages 5068–5078. Curran Associates, Inc.
- Alexander Miller, Will Feng, Dhruv Batra, Antoine Bordes, Adam Fisch, Jiasen Lu, Devi Parikh, and Jason Weston. 2017. [ParlAI: A dialog research software platform](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 79–84, Copenhagen, Denmark. Association for Computational Linguistics.
- Shikhar Murty, Pang Wei Koh, and Percy Liang. 2020. [Expbert: Representation engineering with natural language explanations](#). In *Proceedings of the Association for Computational Linguistics*.
- Alan Ritter, Colin Cherry, and William B Dolan. 2011. [Data-driven response generation in social media](#). In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 583–593.
- Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. [Style transfer from non-parallel text by cross-alignment](#). In *Advances in neural information processing systems*, pages 6830–6841.
- Alessandro Sordani, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. [A neural network approach to context-sensitive generation of conversational responses](#). In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 196–205.
- Shashank Srivastava, Igor Labutov, and Tom Mitchell. 2017. [Joint concept learning and semantic parsing from natural language explanations](#). In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pages 1527–1536.
- Oriol Vinyals and Quoc Le. 2015. [A neural conversational model](#). *arXiv preprint arXiv:1506.05869*.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R’emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. [Huggingface’s transformers: State-of-the-art natural language processing](#). *ArXiv*, abs/1910.03771.
- Jingjing Xu, Xu Sun, Qi Zeng, Xiaodong Zhang, Xuancheng Ren, Houfeng Wang, and Wenjie Li. 2018. [Unpaired sentiment-to-sentiment translation: A cycled reinforcement learning approach](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 979–988, Melbourne, Australia. Association for Computational Linguistics.

- 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, Melbourne, Australia. Association for Computational Linguistics.
- 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*.
- Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232.

## A Dataset Statistics

We are going to validate our approach on the chatbot’s performance using PERSONACHAT (Zhang et al., 2018) dialogue dataset and Human-Bot feedback dataset (Hancock et al., 2019). Table 6 reports the size of each dataset, all of which are available via ParlAI.<sup>2</sup>

Task	Train	Valid	Test	Total
DIALOGUE <sub>(HUMAN-HUMAN)</sub>	131438	7801	6634	145873
FEEDBACK <sub>(HUMAN-BOT)</sub>	60000	1000	1000	62000

Table 4: The number of examples used in our experiments by task and split. Note that the HH DIALOGUE examples come from the PERSONACHAT dataset, HB FEEDBACK examples from Hancock et al. (2019)

To train the FEED2RESP model, we take the entire FEEDBACK dataset and an equal number of randomly chosen samples from the DIALOGUE dataset. We then use a train-dev-test split of 0.8:0.1:0.1 for training and evaluation of the model.

Task	Train	Valid	Test	Total
Style Transfer	96000	12000	12000	120000

Table 5: The number of samples per split used in our style transfer experiment. We take an equal number of samples from the FEEDBACK and DIALOGUE datasets and randomly shuffle them to create train, validation and test splits. The number of samples of each class in all the splits are ensured to be evenly distributed.

Statistic	Human-Human	Feedback
#Words in context <sub>(MEAN)</sub>	79	13
#Words in context <sub>(MEDIAN)</sub>	77	6
#Words per turn <sub>(MEDIAN)</sub>	10.7	7.1
#Words per turn <sub>(MEAN)</sub>	11	6
#Turns <sub>(MEAN)</sub>	4	1.5

Table 6: The number of words per context and per turn. The second part is the average number of turns in a conversation.

We examine the average number of turns and words in dialogues from the the feedback and human-human conversation distributions. We see that on an average, the dialogues in the feedback distribution have fewer number of turns than in human-human conversations. The average number of words per turn is also fewer on average.

<sup>2</sup>[https://parl.ai/projects/self\\_feeding/](https://parl.ai/projects/self_feeding/)

## B Preparation of Training Data

We use the dataset provided by Hancock et al. (2019), which is a cleaner version of PERSONACHAT dataset and comes with a new crowd-sourced test set. We sample an equal number of examples from the DIALOGUE dataset, giving them a label 0, and FEEDBACK dataset, giving them a label of 1. The final response are combined with last n turns with an delimiter [RES]. Typically, n=2 turns are used for each conversation example. Conversation turns are separated with delimiter tokens [P1] or [P2].

## C FEED2RESP examples

Here we include several examples of predictions from different models in Table 9 .

## D Computing Infrastructure and Runtime of Experiments

All experiments are conducted on Nvidia V100 GPUs. Average runtime of experiments is listed in Table 7 and assume running experiments on a single Nvidia V100 GPU.

Task	Approximate Training Time
BIENCODER	16H
POLYENCODER	16H
FEED2RESP	48H

Table 7: Average runtime(in hours) of various experiments conducted

## E Regular Expressions to modify FEEDBACK

As described in Section 4.4, we use the combination of following regular expressions to strip the filler words and extra choices in feedback data:

```
r"you could have|you should have|
you could|you should"
r"^(said|^saying|^say|^tell |^told
|^admit |asked|^ask|^answer
|^answered|^talked|^talk "
r"^(about|^me|^that"
```

```
r"if|whether|not" "you are " to "i am "
"your "to "my "
"you\'ve " to "i\'ve "
"you were" to "i was"
"you " to "i "
```



Model	layers	heads	optimizer	learning rate	#parameters
BART Large	12	16	AdamW	5e-6	406M
Discriminator	4	4	AdamW	1e-4	16M
BIENCODER	2	2	Adamax	2.5e-3	23M
POLYENCODER	12	12	Adamax	5e-05	256M

Table 8: The hyperparameters used to obtain the results in Table 2.

Conv. Hist.	[H] It makes more sense to order food online. There are better deals. [B] good choice. what is your favorite food? [H] pizza. what is yours?
Feedback FEED2RESP	you could have spoken about your favorite food my favorite food is pasta.
Conv. Hist.	[H] hello [B] hi. how are you doing? [H] what do you do for a living?
Feedback FEED2RESP	tell me a job or career i am a teacher.
Conv. Hist.	[H] hello [B] hello [H] do you like to watch movies?
Feedback FEED2RESP	yes or no yes i like movies or no i hate movies.
Conv. Hist.	[H] I tried to climb a tree and broke my ankle.
Feedback FEED2RESP	that sounds painful i am sorry. that sounds painful.
Conv. Hist.	[H] hello [B] hi. [H] do you exercise?
Feedback FEED2RESP	yes, i do exercise. yes, i do exercise and go jogging.
Conv. Hist.	[H] what do you think about wintertime?
Feedback FEED2RESP	you could say that you like snow or you think it is too cold. you like snow or you think it is too cold.
Conv. Hist.	[H] hello [B] hello [H] have you ever ate chinese food?
Feedback FEED2RESP	yes i like chinese. yes i like chinese noodles and dumplings.
Conv. Hist.	[H] who is the greek goddess of love? [B] yes , i lived in south florida for years [H]] sorry, i meant who is the greek goddess of love?
Feedback FEED2RESP	venus venus is a very good player. he has a lot of experience playing with the likes of jurgen klinsmann.

Table 9: Examples of transformed feedback by our model, Feed2Resp, resulting in specific responses.