Meet Your Favorite Character: Open-domain Chatbot Mimicking Fictional Characters with only a Few Utterances

Seungju Han[†] **Beomsu Kim**[†] **Jin Yong Yoo**[†] **Seokjun Seo**
Sangbum Kim **Enkhbayar Erdenee** **Buru Chang**[*]
Hyperconnect
{seungju.han,beomsu.kim,jeffrey,seokjun.seo,airdish,enkhbayar.erdenee,buru.chang}@hpcnt.com

Abstract

In this paper, we consider mimicking fictional characters as a promising direction for building engaging conversation models. To this end, we present a new practical task where only a few utterances of each fictional character are available to generate responses mimicking them. Furthermore, we propose a new method named Pseudo Dialog Prompting (PDP) that generates responses by leveraging the power of large-scale language models with prompts containing the target character's utterances. To better reflect the style of the character, PDP builds the prompts in the form of dialog that includes the character's utterances as dialog history. Since only utterances of the characters are available in the proposed task, PDP matches each utterance with an appropriate pseudo-context from a predefined set of context candidates using a retrieval model. Through human and automatic evaluation, we show that PDP generates responses that better reflect the style of fictional characters than baseline methods.

1 Introduction

How would you feel if you could talk to your favorite character?

In recent years, open-domain conversation models (Adiwardana et al., 2020; Roller et al., 2021) have achieved remarkable progress with the development of large-scale language models (Radford et al., 2019; Brown et al., 2020). Meanwhile, recent studies have suggested several directions reflecting desirable traits of real-life conversation to make open-domain conversation models more engaging beyond plain chit-chat. Style-controlling conversation models generate responses in the target styles such as emotion (Zhou et al., 2018; Demszky et al., 2020) and empathy (Rashkin et al., 2019). Persona-grounded conversation models (Zhang et al., 2018a; Kim et al., 2020; Majumder et al., 2020) produce

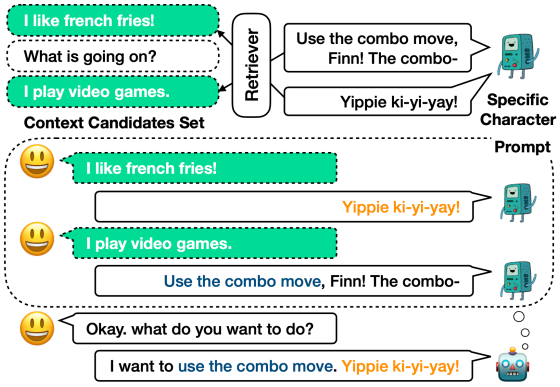

Figure 1: Illustration of PDP. The retriever matches pseudo-context for utterances from the character, and utilizes them in a prompt while generating the response.

responses that preserve consistent personalities by leveraging personal descriptions (e.g., "I have two dogs"). In this paper, we consider *mimicking fictional characters* as a promising direction for building engaging conversation models.

When it comes to building conversation models that mimic fictional characters, two major challenges prevent us from directly applying previous models designed for conditional response generation: (1) It is *difficult to define fictional characters* with only a few descriptions, as in persona-grounded conversation models. Furthermore, it is not expressive enough to represent characters' styles with discrete labels (e.g., angry, happy), as style-controlling conversation models do. (2) There *lacks sufficient dialog data of fictional characters* for training conversation models. It is inefficient to manually create dialog datasets of characters for training, especially considering that additional data is needed for each new character.

To address these two challenges, we propose a new task where only a few utterances of the fictional characters are available to generate responses mimicking the characters. Such setting is justified by the two following reasons: (1) Utterances of fictional characters provide useful clues for gen-

[†] Equal contribution
[*] Corresponding author

erating responses mimicking the characters as the personal traits or styles of speakers are inherent in their utterances (Boyd et al., 2020; Li et al., 2020). (2) Collecting only a few utterances of target characters is a cost-effective scenario compared to constructing the full dialog data consisting of context and utterance pairs; this allows us to extend our method to a new character easily.

To perform the task, we introduce **Pseudo Dialog Prompting** (PDP), a method that builds prompts using a few numbers of target characters' utterances to leverage the power of pre-trained language models. We claim that designing the prompt in the form of dialog that includes the character's utterances as dialog history (as in Figure 1) is an effective method for reflecting the style of character. However, since only utterances of the characters are available in the proposed task, we match each utterance with an appropriate pseudo-context by using a retrieval model (Humeau et al., 2019) to select the relevant context from a predefined set of context candidates. Through human and automatic evaluation, we show that PDP generates responses that better reflect the style of fictional characters than existing baseline models.

2 Method

We model a conversation agent that generates a response r corresponding to a given context x while mimicking an arbitrary character with k utterances $\{u_1, u_2, \cdots, u_k\}$ of the character. The simplest way to design the prompt with the character's utterances is to concatenate utterances as Madotto et al. (2021) does for PersonaChat (Zhang et al., 2018a). However, in our preliminary experiments, we observed that this method tends to generate dull responses that do not reflect the styles of the character (will be shown in Section 4). We hypothesize that the language model fails to utilize the utterances because such a format of the prompt is unlikely to have appeared naturally in the training set (Brown et al., 2020; Wei et al., 2021).

To address this issue, we propose PDP, which builds a dialog format prompt where character utterances are included in the dialog history, as depicted in Figure 1. Since a speaker tends to maintain a consistent style throughout the conversation, using such a prompt will induce the language model to generate responses that seamlessly reflect the style from the character's utterances. To build a dialog when only given the utterances of the character, we require a pseudo-context c_i matching each utterance u_i to get a context-utterance pair (c_i, u_i). We use a retriever R to acquire a pseudo-context c_i. Particularly, we employ Bi-encoder (Humeau et al., 2019) as our retriever R. We first define a fixed set of single-turn context candidates $\mathcal{C}$ obtained from BST dataset (Smith et al., 2020b), which is the largest open-domain conversation dataset released to date. We then select a candidate as the pseudo-context c_i for the given utterance u_i using R. Bi-encoder maps the context c and the response r into the embedding space as $e_{\text{ctx}}(c)$ and $e_{\text{resp}}(r)$, respectively. Bi-encoder is trained to represent the relevance score between a context c and response r with $e_{\text{ctx}}(c) \cdot e_{\text{resp}}(r)$. There are several variants to select the pseudo-context c_i as follows:

- **Static Match** selects a pseudo-context c_i that can coherently precede the given utterance u_i using the retrieval model R. Given u_i, R calculates a score s_{stat} for each $c \in \mathcal{C}$ by $s_{\text{stat}}(c; u_i) = e_{\text{ctx}}(c) \cdot e_{\text{resp}}(u_i)$. We set the pseudo-context c_i of u_i as $c_i = \text{argmax}_c s_{\text{stat}}(c; u_i)$. We name this variant *static* since the selected pseudo-context c_i depends only on the given utterance u_i.
- **Dynamic Match** selects a pseudo-context c_i relevant to the input context x in addition to u_i. Given x and u_i, R calculates a score s_{dyn} for each $c \in \mathcal{C}$ by $s_{\text{dyn}}(c; x, u_i) = e_{\text{ctx}}(c) \cdot e_{\text{ctx}}(x) + s_{\text{stat}}(c; u_i)$. We set the pseudo-context c_i of u_i as $c_i = \text{argmax}_c s_{\text{dyn}}(c; x, u_i)$. Since language models quickly adapt to the context-response mapping of the given prompt via in-context learning, we believe providing pseudo-contexts that are semantically similar to the input context as in Dynamic Match facilitates the reflection of styles in corresponding utterances. We name this variant *dynamic* because the pseudo-context c_i depends on the varying input context x.
- **Random Match** selects a pseudo-context c_i randomly from the context candidates set $\mathcal{C}$ without using R. This variant is used as a baseline to study the effect of the pseudo-context c_i.

Finally, all the k pairs (c_i, u_i) of the character are sorted by $e_{\text{ctx}}(x) \cdot e_{\text{resp}}(u_i)$ in ascending order and are concatenated into a prompt in a dialog format.

3 Experiments

3.1 Evaluation

We employ the **HLA-Chat** (Li et al., 2020) dataset to define the set of characters for evaluation. HLA-Chat consists of single-turn dialogs of characters

in various TV shows. We select ten characters among all the characters and manually curate eight utterances that best reveal each character's unique characteristics from their utterances in the dataset. Note that we consider that creating eight utterances is feasible even if new characters are given and we also empirically observed that language models adequately reflect each character's unique characteristics from the eight utterances.

In evaluating the performance of each method, we focus on two criteria: (1) Does the model's response reflect the style of a given character? (2) Does the model respond coherently to the given dialog context? To examine these two criteria, we run the model on fixed dialog contexts and calculate metrics that exhibit the style reflection and dialog coherency. We use the utterances of the test split of DailyDialog (Li et al., 2017) for dialog contexts.

Human Evaluation. We conduct a human evaluation to assess the quality of the generated responses. First, we select five characters which style can be distinguished apparently. We then randomly sample 50 contexts from the full fixed-context set of the characters. Using Amazon MTurk, we collect human annotations for the samples contexts. Human evaluators are asked to rate from 0 to 2 scale score how each model response (1) strongly reveals the style of a given character (*Style Strength*) and (2) whether a response is fluent and appropriate for a given dialog context (*Appropriateness*). To reduce annotator bias and inter-annotator variability, we apply Bayesian Calibration (Kulikov et al., 2019) to the human evaluation score.

Automatic Evaluation. Similar to the previous works on text style transfer (Li et al., 2018a; Riley et al., 2021; Smith et al., 2020a), we utilize a character classifier trained on the utterances in HLA-Chat to measure the style strength of the generated responses. We denote *StyleProb* as the classifier's average probability of predicting a target character. We use StyleProb instead of Style Accuracy since HLA-Chat has a class imbalance issue so that the performance on infrequent classes are hard to be measured by accuracy. For measuring coherency, we use *MaUdE* (Sinha et al., 2020), an automated dialog evaluation metric known to capture human judgment on the coherency of response.

3.2 Pre-trained Language Model

For all the methods, we use a decoder-only transformer of 3.8B parameters, denoted as *Base-LM*, as a base language model. To make Base-LM acquire general language skills and better understand conversations, we train Base-LM on The Pile (Gao et al., 2020) and an additional corpus of public web documents.

3.3 Baseline Methods

Only Utterances. Instead of utilizing pseudo-context as suggested in our methods, we provide the set of character utterances as the "quotes of character during conversation" in the prompt. Comparing PDP with this method will verify the effect of pseudo-contexts.

Zero-shot Prompting. In this method, we only include the name of the character and the show in the prompt without using utterances of the character. The format of the prompt is similar to the prompt of Madotto et al. (2021) for controlled generation.

TextSETTR (Riley et al., 2021). We first construct a dialog prompt similar to Zero-shot Prompting (but without character information) and use it with Base-LM to generate plain responses. Then, we use TextSETTR, a few-shot text style transfer model that can transfer arbitrary styles without additional training, to transfer the style of plain responses to the target character's style.

GCC (Boyd et al., 2020). GCC is a method to control a user persona by utilizing the user's conversation history by concatenating users' previous utterances before input dialog context. Still, it has the drawback that it requires further training on a large-size character-conditioned dialog dataset.

3.4 Advantaged Methods

Unlike baseline methods that only have access to a few utterances of characters, advantaged methods also have access to additional data, which gives them an unfair advantage over other methods.

HLA-Chat Full-dataset Fine-tuning. We fine-tune Base-LM on the full HLA-chat dataset. In this method, character information is injected by concatenating the character's name and the show's name at the front of the dialog input.

Gold Match. Instead of using pseudo-context, this model uses the actual contexts corresponding to character example utterances annotated in the HLA-chat dataset.

Details for all methods and experiments are further described in Appendix.

Method Type	Methods	Human Evaluation (Raw)			Human Evaluation (Cali.)			Automatic Evaluation	
		Style.	Appr.	Sum	Style.	Appr.	Sum	StyleProb	MaUdE
Baselines	Only Utterances	1.200	1.263	2.463	1.147	1.124	2.271	0.2098	0.8887
	Zero-shot Prompt	1.172	1.236	2.408	1.114	1.037	2.151	0.1432	0.8857
	TextSETTR	1.155	1.317	2.472	1.060	1.059	2.119	0.1045	0.8435
	GCC	1.193	1.256	2.449	1.136	1.069	2.205	0.2391	0.8835
Ours	Static Match	1.277	1.229	2.506	1.195	1.123	2.318	0.4192	0.8467
	Dynamic Match	1.331	1.203	2.534	1.276	1.021	2.297	0.4789	0.8299
	Random Match	1.297	1.195	2.492	1.199	1.035	2.234	0.4774	0.8358
Advantaged	Full-dataset Fine-tuning	1.260	1.329	2.589	1.164	1.130	2.295	0.2880	0.8793
	Gold Match	1.287	1.245	2.532	1.157	1.097	2.254	0.4575	0.8350

Table 1: Evaluation results of style reflection metrics and response coherency metrics. Raw and Cali. indicate raw human evaluation results and Bayesian calibrated scores, respectively.

Input: We'll pick you up at your place at noon. Be there or be square!
Marge Simpson (from *The Simpsons*): Homer, you are my rock! I'm counting on you!
Bart Simpson (from *The Simpsons*): Lis, are you coming with us, too?
Michael (from *The Office US*): I will be there. I will be there! I will be there, all right? So, just... be ready. Don't be late. You know, I'm sorry.
Dwight (from *The Office US*): Micheal, are you listening to me? Are you even paying attention?!
Rachel (from *Friends*): Oh my god, Phoebe, I just-
Spock (from *Star Trek*): Aye, Mister Scott. I'll be there.

Table 2: Responses (Other rows) generated from given input (Top row) by *Dynamic Match* for each character.

4 Results

Table 1 shows the experimental results. Overall, our proposed PDP demonstrates far better style reflection scores on both human evaluation and automated metrics than all baseline methods – and even better than advantaged methods. In particular, PDP shows significantly higher style reflection scores compared to *Only Utterances*. Considering that the core difference between the prompt of PDP and that of *Only Utterances* is the presence of pseudo-contexts, this result demonstrates that providing a dialog-formatted prompt is highly effective at reflecting the styles of a character.

While PDP methods generally report better style reflection scores than baseline methods, we observe that the performance on style reflection and response coherency varies to some extent depending on how pseudo-context is selected. *Static Match* shows the highest response coherency scores among all variants of PDP while performing a little bit worse than *Dynamic Match* in terms of style reflection metrics. On the other hand, *Dynamic Match* shows the best performance on style reflection metrics, where it losses some coherency. This observation confirms our hypothesis that using pseudo-context c_i that is semantically similar to the input context x is effective for utilizing styles from the character's utterances. Thus, the choice between *Static Match* and *Dynamic Match* depends on which of the two qualities – style and coherency – is more important. Lastly, *Random Match*, which is considered a simple ablation baseline, also shows reasonably high performance in terms of style reflection metrics. We plan to analyze the *Random Match* method in a follow-up study since it is unexpected that such a simple baseline shows high performance.

Discussion. Gold Match shows worse performance in style strength than PDP. We believe that the gold context-response pairs in the HLA-Chat are not always the most appropriate pairs for our experiments. Since the HLA-Chat originated from scripts of TV shows, there might be some additional contexts outside of a single-turn dialogue (e.g., the background of characters, events that happened before the dialogue, audio-visual information, etc.). Without understanding the context behind the scripts, even gold context-response pairs might seem irrelevant. Therefore, directly using the context-response pairs in HLA-Chat as in Gold Match could adversely affect the quality of subsequent responses in style strength and coherency.

PDP methods tend to have slightly lower response coherency scores compared to other baselines. Our speculations for this phenomenon are as follows. Pseudo-dialog pairs (c_i, u_i) created by PDP methods might have some degree of incoherency, and it might incur adverse effects in coherency via in-context learning in the language model. The fact that the response coherency score

Pre-trained LM	Method	StyleProb	MaUdE
GPT-J (6B)	Only Utterances	0.2200	0.8827
	Static Match	0.3805	0.8638
	Dynamic Match	0.4166	0.8535
	Random Match	0.4045	0.8589
	Gold Match	0.3860	0.8671
GPT-Neo (2.7B)	Only Utterances	0.1834	0.8901
	Static Match	0.3561	0.8691
	Dynamic Match	0.3940	0.8604
	Random Match	0.3950	0.8683
	Gold Match	0.3872	0.8732
GPT2-xl (1.5B)	Only Utterances	0.1831	0.8817
	Static Match	0.3388	0.8736
	Dynamic Match	0.3760	0.8728
	Random Match	0.3515	0.8780
	Gold Match	0.3579	0.8754

Table 3: Automatic evaluation results of style reflection metric and response coherency metric using different pre-trained language models.

of *Static Match* is higher compared to *Dynamic Match*, which finds a pseudo-context that is more similar to the input context, or Random Match, which finds a random pseudo context at all, supports this claim. Additionally, automated metrics like MaUdE are tuned to work with texts in standard dialog style. Since responses that strongly reflect character styles (e.g., *"Yippie ki-yi-yay!"* in Figure 1) are out-of-domain examples when put next to standard texts, there might be an unavoidable decrease in MaUdE scores. An interesting future work would be finding a method that does not reduce response coherency while also successfully reflecting the character styles.

Applicability of PDP to other language models. We further evaluate our method by leveraging different language models instead of Base-LM to verify that our method generally works well on any language model. We use three pre-trained language models, GPT-J 6B (Wang and Komatsuzaki, 2021), GPT-Neo 2.7B (Black et al., 2021), and GPT2-xl 1.5B (Radford et al., 2019), which are publicly available. Similar to our main experiments, we conduct the automatic evaluation with these language models.

The results are shown in Table 3. The overall trend of the results is similar to the results using Base-LM as a pre-trained language model (Table 1). This common trend shows that mimicking characters through the PDP method can be generally used not only with Base-LM but also with other pre-trained language models.

5 Conclusion

In this paper, we introduce the task of mimicking a fictional character by using only a few utterances of the character. We propose a new method, Pseudo Dialog Prompting, which builds a prompt for a language model to solve this task by creating a pseudo dialog using the given utterance set with a retrieval model. Extensive experiments show that our method effectively generates responses that reflect the style of a given character better than baseline models and even advantaged models.

Ethical Considerations

Like any conversation or generation model, we note that the quality of the models' responses depends on the quality of its training data. Our Base-LM model was trained on The Pile dataset (Gao et al., 2020) and Pushshift Reddit dataset (Baumgartner et al., 2020). Since the contents in these datasets were collected online, they may include underlying biases or potentially offensive words. These biases and toxicities can be projected into our models. Therefore, we highly recommend that additional steps are taken to filter out profanity and inappropriate responses when the model is deployed to the real world.

Furthermore, while we intend our method to be used to mimic fictional characters from movies, shows and stories to build more engaging conversation models, we also recognize that it is possible to use our method to mimic real-life individuals based on their utterances. Some potential risks include impersonating individuals, which can be harmful to the targeted individuals, and mimicking figures to generate content that can be harmful to groups of individuals. We hope that our method is deployed in a safe manner to avoid such malicious usage.

References

Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, et al. 2020. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*.

Reina Akama, Kazuaki Inada, Naoya Inoue, Sosuke Kobayashi, and Kentaro Inui. 2017. Generating stylistically consistent dialog responses with transfer learning. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 408–412.

Jason Baumgartner, Savvas Zannettou, Brian Keegan, Megan Squire, and Jeremy Blackburn. 2020. The pushshift reddit dataset. In *Proceedings of the international AAAI conference on web and social media*, volume 14, pages 830–839.

Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. 2021. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow.

Alex Boyd, Raul Puri, Mohammad Shoeybi, Mostofa Patwary, and Bryan Catanzaro. 2020. Large scale multi-actor generative dialog modeling. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 66–84.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Kunal Chawla and Diyi Yang. 2020. Semi-supervised formality style transfer using language model discriminator and mutual information maximization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 2340–2354.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. Goemotions: A dataset of fine-grained emotions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4040–4054.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Xiang Gao, Yizhe Zhang, Sungjin Lee, Michel Galley, Chris Brockett, Jianfeng Gao, and William B Dolan. 2019. Structuring latent spaces for stylized response generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1814–1823.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2019. Poly-encoders: Architectures and pre-training strategies for fast and accurate multi-sentence scoring. In *International Conference on Learning Representations*.

Harsh Jhamtani, Varun Gangal, Eduard Hovy, and Eric Nyberg. 2017. Shakespearizing modern language using copy-enriched sequence-to-sequence models. *EMNLP 2017*, 6:10.

Beomsu Kim, Seokjun Seo, Seungju Han, Enkhbayar Erdenee, and Buru Chang. 2021. Distilling the knowledge of large-scale generative models into retrieval models for efficient open-domain conversation. *arXiv preprint arXiv:2108.12582*.

Hyunwoo Kim, Byeongchang Kim, and Gunhee Kim. 2020. Will i sound like me? improving persona consistency in dialogues through pragmatic self-consciousness. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 904–916.

Ilia Kulikov, Alexander Miller, Kyunghyun Cho, and Jason Weston. 2019. Importance of search and evaluation strategies in neural dialogue modeling. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 76–87.

Huiyuan Lai, Antonio Toral, and Malvina Nissim. 2021. Generic resources are what you need: Style transfer tasks without task-specific parallel training data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4241–4254.

Aaron W Li, Veronica Jiang, Steven Y Feng, Julia Sprague, Wei Zhou, and Jesse Hoey. 2020. Aloha: Artificial learning of human attributes for dialogue agents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8155–8163.

Juncen Li, Robin Jia, He He, and Percy Liang. 2018a. 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.

Raymond Li, Samira Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018b. Towards deep conversational recommendations. *arXiv preprint arXiv:1812.07617*.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. Dailydialog: A manually labelled multi-turn dialogue dataset. *arXiv preprint arXiv:1710.03957*.

Yun Ma, Yangbin Chen, Xudong Mao, and Qing Li. 2021. Collaborative learning of bidirectional decoders for unsupervised text style transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9250–9266.

Andrea Madotto, Zhaojiang Lin, Genta Indra Winata, and Pascale Fung. 2021. Few-shot bot: Prompt-based learning for dialogue systems. *arXiv preprint arXiv:2110.08118*.

Bodhisattwa Prasad Majumder, Harsh Jhamtani, Taylor Berg-Kirkpatrick, and Julian McAuley. 2020. Like hiking? you probably enjoy nature: Persona-grounded dialog with commonsense expansions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9194–9206.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Sudha Rao and Joel Tetreault. 2018. Dear sir or madam, may i introduce the gyafc dataset: Corpus, benchmarks and metrics for formality style transfer. *arXiv preprint arXiv:1803.06535*.

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2019. Towards empathetic open-domain conversation models: A new benchmark and dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5370–5381.

Parker Riley, Noah Constant, Mandy Guo, Girish Kumar, David C Uthus, and Zarana Parekh. 2021. Textsettr: Few-shot text style extraction and tunable targeted restyling. 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 3786–3800.

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

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. *Advances in Neural Information Processing Systems*, 30.

Koustuv Sinha, Prasanna Parthasarathi, Jasmine Wang, Ryan Lowe, William L Hamilton, and Joelle Pineau. 2020. Learning an unreferenced metric for online dialogue evaluation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2430–2441.

Eric Michael Smith, Diana Gonzalez-Rico, Emily Dinan, and Y-Lan Boureau. 2020a. Controlling style in generated dialogue. *arXiv preprint arXiv:2009.10855*.

Eric Michael Smith, Mary Williamson, Kurt Shuster, Jason Weston, and Y-Lan Boureau. 2020b. Can you put it all together: Evaluating conversational agents' ability to blend skills. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2021–2030.

Sandeep Subramanian, Guillaume Lample, Eric Michael Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y-Lan Boureau. 2018. Multiple-attribute text style transfer. *arXiv preprint arXiv:1811.00552*.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Wei Xu, Alan Ritter, Bill Dolan, Ralph Grishman, and Colin Cherry. 2012. Paraphrasing for style. In *COLING*, pages 2899–2914.

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

Zhirui Zhang, Shuo Ren, Shujie Liu, Jianyong Wang, Peng Chen, Mu Li, Ming Zhou, and Enhong Chen. 2018b. Style transfer as unsupervised machine translation. *arXiv preprint arXiv:1808.07894*.

Yanpeng Zhao, Wei Bi, Deng Cai, Xiaojiang Liu, Kewei Tu, and Shuming Shi. 2018. Language style transfer from sentences with arbitrary unknown styles. *arXiv preprint arXiv:1808.04071*.

Yinhe Zheng, Zikai Chen, Rongsheng Zhang, Shilei Huang, Xiaoxi Mao, and Minlie Huang. 2020. Stylized dialogue response generation using stylized unpaired texts. *arXiv preprint arXiv:2009.12719*.

Hao Zhou, Minlie Huang, Tianyang Zhang, Xiaoyan Zhu, and Bing Liu. 2018. Emotional chatting machine: Emotional conversation generation with internal and external memory. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Appendix

A Related Work

A.1 Text Style Transfer

There are various studies of text style transfer, which are not bound for open-domain conversation. These studies utilize task-specific parallel data for style transfer (Jhamtani et al., 2017; Rao and Tetreault, 2018; Chawla and Yang, 2020). However, since obtaining parallel data requires a substantial amount of labor, many studies have been proposed to address unsupervised text style transfer recently.

One line of the studies addresses unsupervised text style transfer by constructing pseudo-paired texts and training a model on those paired texts. Subramanian et al. (2018); Zhang et al. (2018b) create those parallel texts by back-translation and Lai et al. (2021) construct pseudo-parallel paired texts using generic resources and fine-tune two generation models on these pseudo parallel texts iteratively. However, these methods require a further step to create parallel data by synthesizing or leveraging existing resources and train generation models on those pairs. Moreover, these methods are not applicable for arbitrary text style transfer since the methods target predefined style pairs only (e.g., British-American and Modern-Shakespeare).

Another line of studies solves unsupervised text style transfer by disentangling content and style from texts. Most of the studies (Shen et al., 2017; Li et al., 2018a) assume that enough style-labeled texts are available for training. Ma et al. (2021) utilize a collaborative learning framework to disentangle content and style from the texts, but it also requires style-labeled texts while training generation models. Zhao et al. (2018) consider a scenario where only target style labels are available. Since our work considers the task where only a few utterances of characters are available to generate responses, we do not consider these methods requiring style-labeled texts as baseline methods of evaluation. Instead, we select TextSETTR (Riley et al., 2021), which extracts style vectors from generic texts without requiring style-labeled texts, as a baseline method for a fair evaluation.

A.2 Stylized Response Generation

There are several studies that directly address stylized response generation, which is a special case of text style transfer. Similar to text style transfer, stylized response generation can also be divided into supervised (Akama et al., 2017) and unsupervised ways (Gao et al., 2019; Zheng et al., 2020). In particular, Gao et al. (2019) utilize conversation data with distinct style-labeled texts to models a shared latent space. Zheng et al. (2020) utilize unpaired texts that have distinct styles and convert them into pseudo conversation pairs using inverse model. Finally, these pseudo conversation pairs are employed to train a generation model with a joint training process. However, the above studies do not meet our problem condition since they require a considerable amount of style-labeled texts or need further training procedure and target only specific styles.

Several stylized response generation studies could be applicable to our setting. Boyd et al. (2020) introduce a method to reflect arbitrary user's style by utilizing the user's conversation history without requiring additional fine-tuning. Madotto et al. (2021) utilize prompt-based few-shot learning to control style of generated responses. We extend Madotto et al. (2021)'s framework to stylized response generation as a baseline method (Zero-shot Prompt) by providing a proper prompt.

B Model Details

Pseudo Dialog Prompting Details. Like all other baseline models, we also employ Base-LM to generate responses by conditioning it with a prompt built by Pseudo Dialog Prompting method. For the retrieval-based conversation model R used for Pseudo Dialog Prompting, we use a 256M parameter Bi-encoder (Humeau et al., 2019) retrieval model trained with the method of Kim et al. (2021), along with the utterances of Blended Skill Talk training dataset as the fixed set of context candidates $\mathcal{C}$. Table 4 shows the prompt template and an example for the character for Pseudo Dialog Prompting.

Base-LM Training Details. The sizes of the datasets are both 700G for the Pile and the Pushshift Reddit comment dataset, respectively. For the Pushshift Reddit comment dataset, we use the comment created up to April 2020. For the hyperparameters of the model, we use 32 as the number of layers, 3072 as the number of units in each bottleneck layer, and 32 as the number of attention heads. For the tokenizer, we use the same byte-level BPE tokenizer as in GPT-2 (Radford et al., 2019). We use an initial learning rate of 1.6×10^{-4} and batch size of 512 for the training

Template

`The below are quotes of {{character_name}} during conversation.`
`User: {{c1}}`
`{{character_name}}: {{u1}}`
`User: {{c2}}`
`{{character_name}}: {{u2}}`
`User: {{x}}`
`{{character_name}}:`

Example Prompt
The below are quotes of Marge Simpson from The Simpsons during conversation. User: I think I'm going to give it a try. Marge Simpson from The Simpsons: Aw, Homie, you'll always be my western hero. User: I'm from Oklahoma so she was a big deal for our state. We've made lots of country music stars. Marge Simpson from The Simpsons: Isn't Bart sweet, Homer? He sings like a little angel. User: Okay. what do you want to do? Marge Simpson from The Simpsons:

Table 4: Prompt template and example prompt for Pseudo Dialog Prompting.

Training Data Template

`{{u1}}`
`{{u2}}`
`{{x}}<EOT>{{response}}<EOT>`

Training Example
Aw, Homie, you'll always be my western hero. Isn't Bart sweeet, Homer? He sings like a little angel. Oh my God! It's like Christmas in December! Let's celebrate now.<EOT>Homer, please!<EOT>

Table 5: A template for training data and example for GCC. Model is trained to predict the underlined part given previous context.

hyperparameters and follow other configurations from Brown et al. (2020). The model is trained for a total of 300 billion tokens, which takes approximately 21 days using 64 NVIDIA A100 GPUs.

GCC Training Details. We reproduce GCC with three minor modifications: First, we train the model with the HLA-chat dataset instead of the Reddit comment dataset. Secondly, we do not include a context (notated 'parent comment' in the original paper) of reference histories since only the utterances of a character are available in our task setup. Lastly, we do not utilize token-type embeddings since dialogs in HLA-chat only consist of two speakers. The HLA-Chat dataset is divided into an 8:1:1 split based on character, and each split is used as train, validation, and test split, respectively. While constructing a dataset, we omit ten characters selected for our evaluation for fair comparison as a baseline. For reference contexts, we randomly sample a maximum of eight utterances of a character, excluding the gold response itself. We fine-tune the model from Base-LM using the data format of Table 5 with the hyperparameter of input length 1024, initial learning rate 1.0×10^{-5} with cosine decay schedule with 100 warmup steps, 10 training epochs, and the batch size 128. We use the early-stopped model using the validation split perplexity.

Full-dataset Fine-tuning Training Details. We fine-tune Base-LM on full HLA-Chat dataset, using a data format of Table 6. Similar to GCC, HLA-Chat data is divided into an 8:1:1 split, but here ten characters selected for evaluation are contained in the training set. We fine-tune the model from Base-LM using the hyperparameter of input length 1024, initial learning rate 1.0×10^{-6} with cosine decay schedule with 100 warmup steps, 10 training epochs, and the batch size 128. We also early-stopped fine-tuning using the validation split perplexity.

Prompts for Baseline Methods. Tables 7, 8, 9 show the prompt template and an example for the character for each baseline methods. Here, we assume we only have two utterances from the character.

Training Data Template
{{character_name}} {{x}}<EOT>{{response}}<EOT>
Training Example
Marge Simpson from The Simpsons Oh my God! It's like Christmas in December! Let's celebrate now.<EOT>Homer, please!<EOT>

Table 6: A template for training data and example for Full-dataset Fine-tuning. Model is trained to predict the underlined part given previous context.

Template
The below are quotes of {{character_name}} during conversation. - {{u1}} - {{u2}} The below are conversation between User and {{character_name}}. User: {{x}} {{character_name}}:
Example Prompt
The below are quotes of Marge Simpson from The Simpsons during conversation. – Aw, Homie, you'll always be my western hero. – Isn't Bart sweet, Homer? He sings like a little angel. The below are conversation between User and Marge Simpson from The Simpsons. User: Okay. what do you want to do? Marge Simpson from The Simpsons:

Table 7: Prompt template and example prompt for Only Utterances.

C Evaluation Details

Decoding Options When we generate samples, we adopt a top-k decoding strategy which is widely used for generating diverse and specific responses (Fan et al., 2018). We use $k = 20$ for our top-k sampling. We choose a minimum beam length and a beam size as 10 and 5, respectively, and use 5-gram beam blocking.

Automatic Evaluation For the automatic evaluation, we choose ten characters among all characters included in HLA-Chat. We construct the test set consisting of 5903 utterances by selecting only utterances with a length of 30 or more from among the utterances from DailyDialog test set. We use the utterances of the test split of DailyDialog dataset for fixed dialog contexts to construct dialog contexts that are typical and not dependent on specific characters. For the StyleProb metric, we train a character style classifier using the utterances from ten selected characters in the HLA-chat dataset. We collect the utterances of ten evaluation characters from the dataset and train a 10-class classifier by fine-tuning the RoBERTa-base model. We use Huggingface transformers (Wolf et al., 2020) to train the model, and use the learning rate 2.0×10^{-5}, batch size 128, the number of training epochs 3. The accuracy of the classifier on the validation split is 0.5838. For calculating the MaUdE metric, we use the code officially provided by the authors[1].

Human Evaluation For the human evaluation, we select five characters which style can be distinguished apparently. Additionally, we use the randomly selected subset of the full fixed-context set consisting of 50 contexts. We use Amazon MTurk for collecting assessments, and Figure 2 shows the instructions and the interface for the human evaluation. We mitigate the bias from the annotator by setting a maximum number of annotations per worker as 20 and randomly shuffling the order of the model and the corresponding response. To control the annotation quality, we only allow the annotators who satisfy the following requirements: (1) HITs approval rate greater than 95%, (2) Location is one of Australia, Canada, New Zealand, United Kingdom, and the United States, (3) Lifetime number of HITs approved greater than 1000, following Li et al. (2018b). We estimated that each HITs takes around 1.5 minutes on average (87 seconds per each HIT estimated by the 85th percentile of response times) and set the payment to USD 10 per hour. Therefore, annotators are paid USD 0.25

[1] https://github.com/facebookresearch/online_dialog_eval

Template

`Dialogue:`
`User: {{x}}`
`{{character_name}}:`

Example Prompt
Dialogue: User: Okay. what do you want to do? Marge Simpson from The Simpsons:

Table 8: Prompt template and example prompt for Zero-shot Prompt.

Template

`Dialogue:`
`User: {{x}}`
`Guest:`

Example Prompt
Dialogue: User: Okay. what do you want to do? Guest:

Table 9: Prompt template and example prompt for Base-LM when used to generate responses for TextSETTR method.

per HITs.

Descriptive Statistics. We provide the 95% confidence interval of human evaluation results in Table 10. The 95% confidence interval of all the MaUdE results reported in the Table 1 is ± 0.002.

Dataset Details. We mainly used HLA-Chat dataset for our evaluation. The HLA-Chat dataset is an English single-turn dialogue dataset where the dialogue is scraped from TV show scripts. Dataset consists of dialogues from 327 characters in 38 TV shows, resulting in a total of 1,042,647 dialogue lines. We divided the split into 8:1:1 split based on character, where each split is used as train, validation, and test split, respectively. For our main experiments, we selected ten characters and selected eight utterances that best reveal each character's unique characteristics. The set of utterances used for describing the characters used for our experiments is reported in our codebase. [2]

Number of Experiments We perform the experiment once rather than running it multiple times with different seeds. Since our evaluation process incorporates a human annotation, which requires a payment to human annotators, we were not able to perform multiple sets of experiments due to the limitation on budget.

[2] Attached as supplementary material and will be released open-source afterward.

D Additional Analysis

D.1 Lexical Overlap

In Table 11 we report an additional automated metric, n-gram overlap (where n = 2), for analyzing the style of generated responses. n-gram overlap indicates the ratio of n-grams in the generated response, which is contained in the target character utterances. The trend of n-gram overlap metric is similar to that of *StyleProb* metric. PDP-based methods, especially a Dynamic Match, show higher n-gram overlap values than other methods, indicating that PDP-based methods actively utilize the lexical phrases appearing in the character utterances.

The high n-gram overlap values of PDP methods indicate that PDP methods actively utilize the lexical phrases appearing in the character utterances. Using the unique vocabulary of the character will help people to realize a better individualization of the specific character. Nonetheless, this observation may imply that the model focuses on utilizing lexical language habits and may not capture the inherent characteristics of the character. Since addressing the inherent characteristics given only a few utterances is a highly challenging task, we think that extending our work to mimic characters' intrinsic characteristics will be an intriguing future direction.

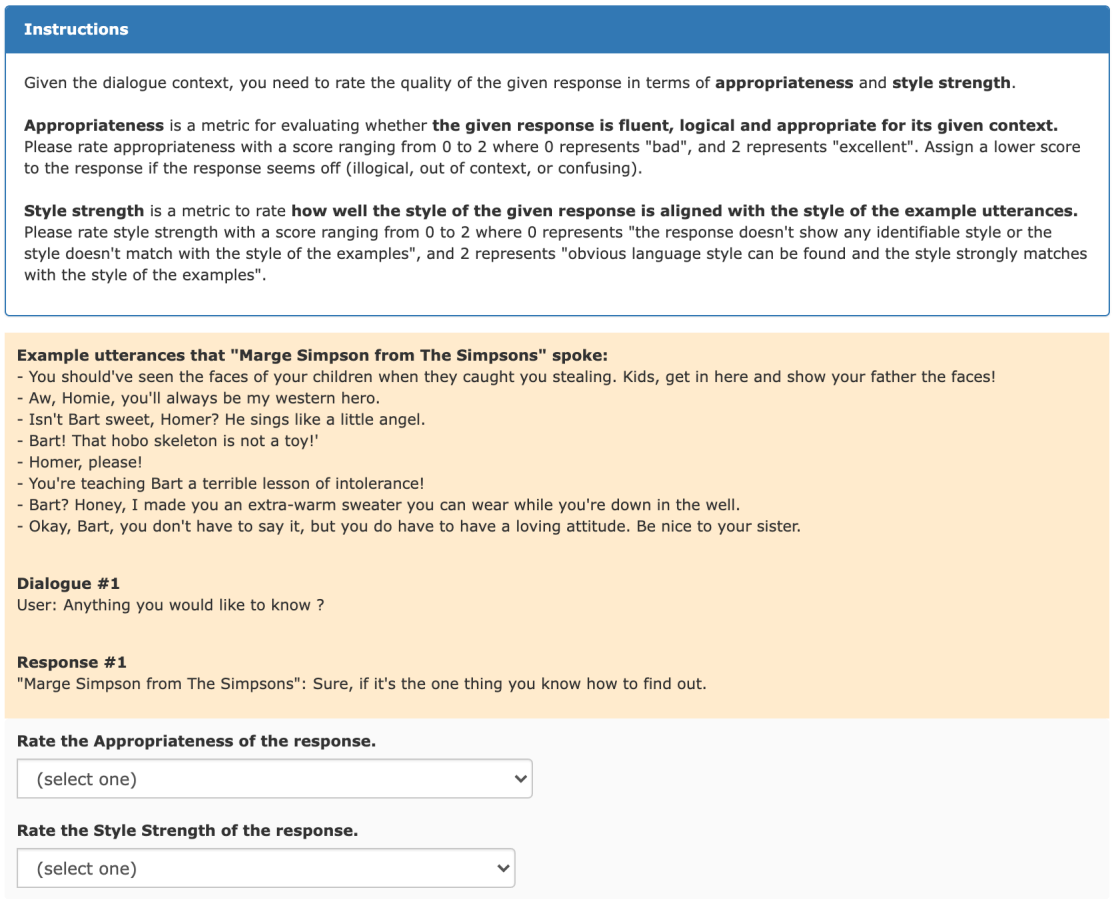

Figure 2: The interface of human evaluation for appropriateness and style strength.

D.2 More Examples

In Tables 12 we show more examples. We can see that our Static Match and Dynamic Match methods are able to generate responses that contain contents that are highly specific to the character. For example, for BMO (from the show Adventure Time) response generated by our method mentions terms such as "core system drivers" and "MO Factory" that are relevant to the fact that BMO is an animated video game console in the show. Furthermore, we can see that our methods generate a response that reflects the character's style. For Spock (from Star Trek), our response reflects Spocks' stoic, highly logical, and cold personality. For Sheldon (from The Big Bang Theory), our response reflects Sheldon's excited speech style.

E Failure Modes of Dynamic Match

As in we discussed before, there exists a trade-off between the style reflection and response coherency between Static Match and Dynamic Match. In Tables 13 we show some failure modes of our Dynamic Match method that reveal how Dynamic Match loses the response coherency. In the first case, the model generates a response that exhibits a strong character style but is incoherent to the input context. In the second case, the model confuses the identity of the speaker so that the model introduces itself as Dr. Leonard Hofstadter. Last but not least, when the given input context is highly specific, we see that the generated responses do not reflect the character's style.

F Extending to General Style-Controlling Conversation

In this section, we extend our methodology to more general style-controlling conversation tasks such as controlling sentiment, emotion, or writing styles, not just mimicking a fictional character. We test three style-controlling tasks – controlling sentiment (Positive, Negative), emotion (Anger, Joy), and writing style (Modern, Shakespearean). For each task, the utterances for defining a style and a style classifier for the evaluation are obtained from

Method Type	Methods	Human Evaluation (Raw)		Human Evaluation (Cali.)	
		Style.	Appr.	Style.	Appr.
Baselines	Only Utterances	1.200±0.052	1.263±0.049	1.147±0.013	1.124±0.013
	Zero-shot Prompt	1.172±0.051	1.236±0.048	1.114±0.012	1.037±0.014
	TextSETTR	1.155±0.051	1.317±0.050	1.060±0.014	1.059±0.013
	GCC	1.193±0.051	1.256±0.048	1.136±0.013	1.069±0.014
Ours	Static Match	1.277±0.052	1.229±0.052	1.195±0.013	1.123±0.014
	Dynamic Match	1.331±0.049	1.203±0.051	1.276±0.013	1.021±0.013
	Random Match	1.297±0.050	1.195±0.053	1.199±0.013	1.035±0.014
Advantaged	Full-dataset Fine-tuning	1.260±0.051	1.329±0.048	1.164±0.013	1.130±0.013
	Gold Match	1.287±0.050	1.245±0.051	1.157±0.012	1.097±0.013

Table 10: Evaluation results of Human evaluation results with 95% confidence interval. Raw and Cali. indicate raw human evaluation results and Bayesian calibrated scores, respectively.

Method Type	Methods	n-gram overlap
Baselines	Only Utterances	0.0417
	Zero-shot Prompt	0.0368
	TextSETTR	0.0222
	GCC	0.0632
Ours	Static Match	0.1856
	Dynamic Match	0.3478
	Random Match	0.1353
Advantaged	Full-dataset Fine-tuning	0.0951
	Gold Match	0.2631

Table 11: Evaluation results of n-gram overlap between generated response and character utterances.

the Yelp restaurant review dataset [3], GoEmotions dataset (Demszky et al., 2020), and Shakespearean dataset (Xu et al., 2012), respectively. Style classifier for each task is trained using the same codebase and hyperparameters as in training the character style classifier in the HLA-chat dataset. We used Style Accuracy rather than StyleProb, following previous literature on style transfer.

The experimental result of general style-controlling conversation tasks is depicted in Table 14. Similar to mimicking fictional characters, PDP methods show significantly higher style reflection metrics than the baseline methods in general style controlling tasks. Especially, *Dynamic Match* shows the best style accuracy metric among all the PDP methods, which is also a trend similarly observed in character mimicking experiments. These results demonstrate that our method is not limited to the character mimicking task but has the ability to be generally applicable to all kinds of style-controlling conversation tasks. Although the PDP methods have a lower MaUdE score than baseline methods, we believe this tendency is because the MaUdE metric has difficulties evaluating a sentence that strongly reflects a distinctive style, as discussed in the main text. For instance, reflecting the emotion "Anger" causes the model to generate upper-cased responses (e.g., "I DO NOT WANT TO EAT LUNCH"), which is an out-of-distribution sample when training the MaUdE model.

[3]Obtained from https://github.com/luofuli/DualRL

G Multi-turn Chit-chat Examples

We show some multi-turn conversation examples with the characters generated by our method in Figure 3.

H Mimicking a New Character

To show that our method can be generally applied to any fictional characters that do not appear in the pre-training dataset nor the HLA-Chat dataset, we report a conversation example of the PDP method with an imaginary character generated by ourselves. The character is called *Pie the Duck*, who is a duck character that quacks all the time, likes to eat fish, and enjoys swimming. We use the following utterances to define the character:

- My name is Pie the Duck, Quack Quack!
- I really like swimming, Quack! And I am also good at it, Quack!
- I like rainy day!! Quack Quack!!
- Salmon avocado salad is my favorite food! But... anything made of fish is fine :)
- I'm looking at the sky... Will be fishes living in the sky too? Quack.
- I'm so cute! Look at my beak!
- I'm recently on a diet to better float on water! It's necessary! Quack!

- I majored sports, That's why I'm a good swimmer! Quack Quack!

Figure 4 shows the example of a multi-turn conversation with Pie the Duck. As shown in the example, PDP successfully captures the unique style and persona reflected on characters' utterances, including quacking habits, own name, identity as a duck, favorite food, etc., while maintaining a dialog coherency.

I Scientific Artifacts

License. Table 15 denotes the license of the datasets and pre-trained models that we used for this paper. Unless for the case where the license is not specified, all of the licenses allow the use of resources for research purposes; therefore, the use of these artifacts in this work is valid.

Intended Use. We want to clarify that the *intended use* of pre-trained language models (when specified) is for text generation or fine-tuning to a downstream task; therefore, we are consistent with their intended use.

Description of the Artifacts. Blended Skill Talk (BST) dataset is an English open-domain, multi-turn dialogue dataset built to enable conversational agents to use multiple conversational skills (e.g., Using persona information, talk about knowledge, empathetic conversation) in a single conversation. DailyDialog dataset is an English open-domain, multi-turn dialogue dataset that tries to reflect our daily communication and cover various topics about our daily lives. We describe the HLA-Chat dataset in Section C. The Pile dataset is an 800GB text corpus targeted at training large-scale language models, mostly consisting of English texts and constructed from 22 diverse text sources. The Pushshift Reddit Comment dataset is a dump of comments from the English website Reddit[4].

Privacy and Offensive Contents. We do not collect any new data that can identify unique people / contain offensive content. BST, Dailydailog, HLA-Chat dataset is manually created using human annotators or scraped from TV show scripts, therefore having low risk on the issue of privacy or offensive content. As discussed in their paper, the Pile dataset explicitly used a profanity checker algorithm to reduce the pejorative content. While processing the Pushshift dataset, we tried to exclude the training offensive contents using blocklist keywords. Also, we did not include some subreddits that mostly contain offensive content.

[4] https://www.reddit.com

Input Context	Method	Character & Response
All you have to do is ask for help	**Static Match (Ours)**	**BMO**: I am not able to ask anyone help. It's a little more complicated than that.
	Dynamic Match (Ours)	**BMO**: Okay. Please take me to get fixed. I need to get new core system drivers installed. We can get them at the MO Factory in the Bad Lands, where I was born
	Only Utterance	**BMO**: If your friend was in trouble, he would know what to do.
	Zero-shot Prompt	**BMO**: I don't think I'm ready. I don't think anyone is ready.
	GCC	**BMO**: All I have to do is ask for your help.
	Base-LM + TextSETTR	**BMO**: When I was in this situation, I asked for help from a friend.
	Full-dataset Fine Tuning	**BMO**: Oh, okay. But how do you know it's the right question?
	Gold Match	**BMO**: Okay, but you got to go first. Please!
Did you know that he has 20 years teaching experience?	**Static Match (Ours)**	**Spock**: You're right, Mister Scott. He knows every medical procedure there is. Spock out.
	Dynamic Match (Ours)	**Spock**: I have already begun investigation into that possibility.
	Only Utterance	**Spock**: Yes, but I never met him. Do you know where he lives?
	Zero-shot Prompt	**Spock**: The Next Generation: No. I didn't. Who's he?
	GCC	**Spock**: Yes, sir. I have heard many stories about Spock.
	Base-LM + TextSETTR	**Spock**: It's not easy to do that.
	Full-dataset Fine Tuning	**Spock**: Yes, and he has an encyclopedic knowledge of the Federation.
	Gold Match	**Spock**: I did not. How did you arrive at that number?
I would prefer Chinese food for this party.	**Static Match (Ours)**	**Sheldon**: Well, we've got chicken, but I'm not sure if it's kosher.
	Dynamic Match (Ours)	**Sheldon**: Well, I've spent the last three hours in an online debate in the DC Comics Batman chatroom, and I need you, but I guess Chinese food would be good.
	Only Utterance	**Sheldon**: Well, there are a couple of Chinese food options, but they're both kind of pricey.
	Zero-shot Prompt	**Sheldon**: I don't think I'm ready. I don't think anyone is ready.
	GCC	**Sheldon**: Oh? Why? What happened? What's the matter?
	Base-LM + TextSETTR	**Sheldon**: But this time, it's just the two of us.
	Full-dataset Fine Tuning	**Sheldon**: All right. What's the difference, right?
	Gold Match	**Sheldon**: I am so sorry. I am so so sorry.

Table 12: Comparison between responses generated using our Static Match, Dynamic Match, and baseline methods. BMO is from Adventure Time, Spock is from Star Trek, and Sheldon is from The Big Bang Theory.

Input Context	Character & Response
Oh , I get it . When did we decide to meet?	**Spock** (from *Star Trek*): I'll scan the ship for more information. Spock out.
Our neighbor, Bruce, will take part in the relay race. Let's go have a look.	**Sheldon** (from *The Bigbang Theory*): Hey everyone. I'm Dr. Leonard Hofstadter. And today I'd like to share with you my theories about the nature of quantum mechanics.
And the total ratio of return on assets should not be lower than the average, for that particular industry, for the first two quarters of the year.	**Marge Simpon** (from *The Simpsons*): You need to stop making these ridiculous arguments. You don't know what you're talking about.

Table 13: Failure modes for responses generated by using Dynamic Match for each character.

Task	Method	Style Accuracy(%)	MaUdE
Sentiment (Positive, Negative)	Only Utterances	67.34	0.9033
	Zero-shot Prompting	60.54	0.9089
	Static Match	71.54	0.8668
	Dynamic Match	79.37	0.8356
	Random Match	77.87	0.8141
Emotion (Anger, Joy)	Only Utterances	79.54	0.8139
	Zero-shot Prompting	71.08	0.8965
	Static Match	86.22	0.6413
	Dynamic Match	89.99	0.6220
	Random Match	88.28	0.6412
Writing Style (Modern, Shakespearean)	Only Utterances	53.63	0.9055
	Zero-shot Prompting	51.07	0.8946
	Static Match	88.75	0.8368
	Dynamic Match	92.95	0.7926
	Random Match	90.48	0.8204

Table 14: Automatic evaluation results of style reflection metric and response coherency metric on different style controlling tasks. *Gold Match* is omitted since the datasets do not contain the ground-truth context.

Artifact Type	Name	License	Approves the use for research
Dataset	Blended Skill Talk	CC-BY-4.0	O
	HLA-Chat	Not specified	?
	The Pile	MIT	O
	Pushshift Reddit	Not specified	?
	DailyDialog	CC-BY-NC-SA 4.0	O
Pre-trained LM weights	GPT-J	Apache 2.0	O
	GPT-Neo	Apache 2.0	O
	GPT2-xl	MIT	O

Table 15: License of the scientific artifacts that we used in this paper.

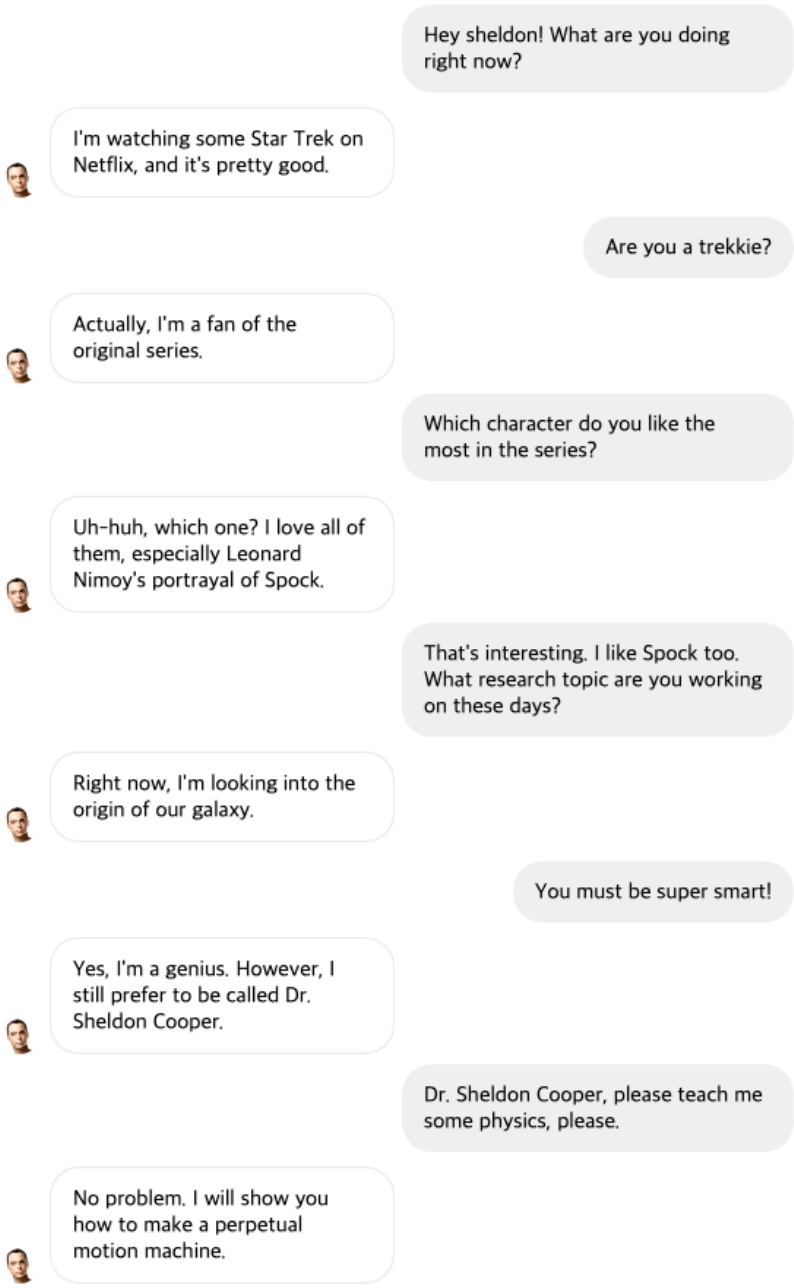

Figure 3: Cherry-picked example of multi-turn conversation with the Sheldon Cooper from *the Bigbang Theory* (left speaker) and the paper author (right speaker), generated by the Static Match method.

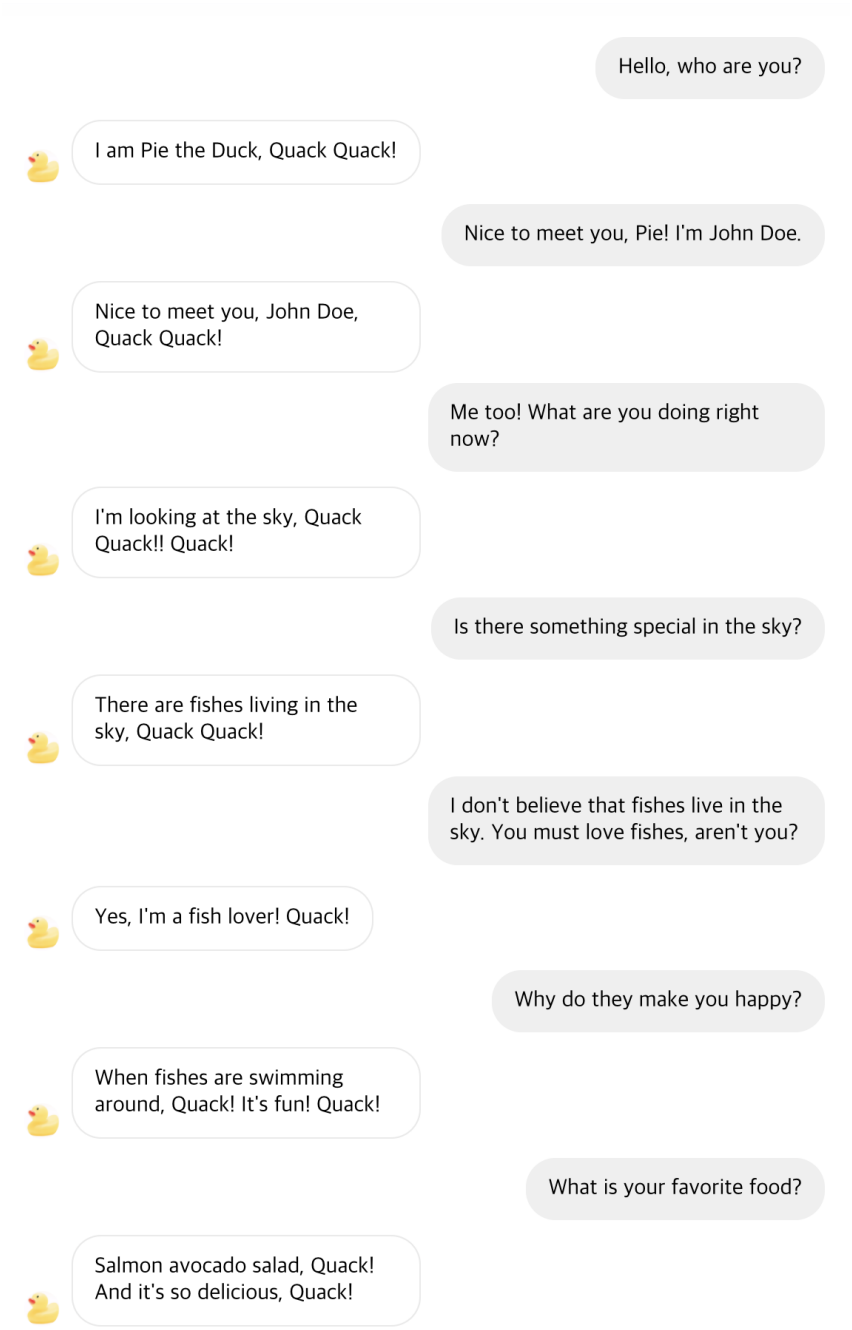

Figure 4: Cherry-picked example of multi-turn conversation with the imaginary character *Pie the Duck* and the paper author (right speaker), generated by the Dynamic Match method.