Learning from Natural Language Feedback

Jérémy Scheurer[1] **Jon Ander Campos**[1 2] **Jun Shern Chan**[1]
Angelica Chen[1] **Kyunghyun Cho**[1 3 4] **Ethan Perez**[1]
[1]New York University, [2]University of the Basque Country, [3]Genentech, [4]CIFAR LMB
{jeremy.scheurer,perez}@nyu.edu

Abstract

Pretrained language models often do not perform tasks in ways that are in line with our preferences, e.g., generating offensive text or factually incorrect summaries. Recent work approaches the above issue by learning from a simple form of human evaluation: comparisons between pairs of model-generated task outputs. Comparison feedback conveys limited information about human preferences per human evaluation. Here, we propose to learn from natural language feedback, which conveys more information per human evaluation. We learn from language feedback on model outputs using a three-step learning algorithm. First, we condition the language model on the initial output and feedback to generate many refinements. Second, we choose the refinement with the highest similarity to the feedback. Third, we finetune a language model to maximize the likelihood of the chosen refinement given the input. In synthetic experiments, we first evaluate whether language models accurately incorporate feedback to produce refinements, finding that only large language models (175B parameters) do so. Using only 100 samples of human-written feedback, our learning algorithm finetunes a GPT-3 model to roughly human-level summarization ability.

1 Introduction

Language Models (LMs) achieve strong performance across diverse NLP tasks, from summarization to question answering and conversational assistants (Radford and Narasimhan, 2018; Radford et al., 2019; Brown et al., 2020; Rae et al., 2021, *interalia*). A key problem with LMs is that they generate text that violates human preferences, such as LM-generated misinformation (Lin et al., 2021), offensive language (Gehman et al., 2020), and factually incorrect outputs such as summaries (Stiennon et al., 2020). Current methods alleviate such issues by training LMs to generate text that scores highly according to human preferences, or a predictive model thereof (Ziegler et al., 2019; Stiennon et al., 2020; Nakano et al., 2021; Ouyang et al., 2022). In this line of work, human evaluators indicate their preferences by comparing text outputs. However, each comparison provides little information per evaluation about human preferences.

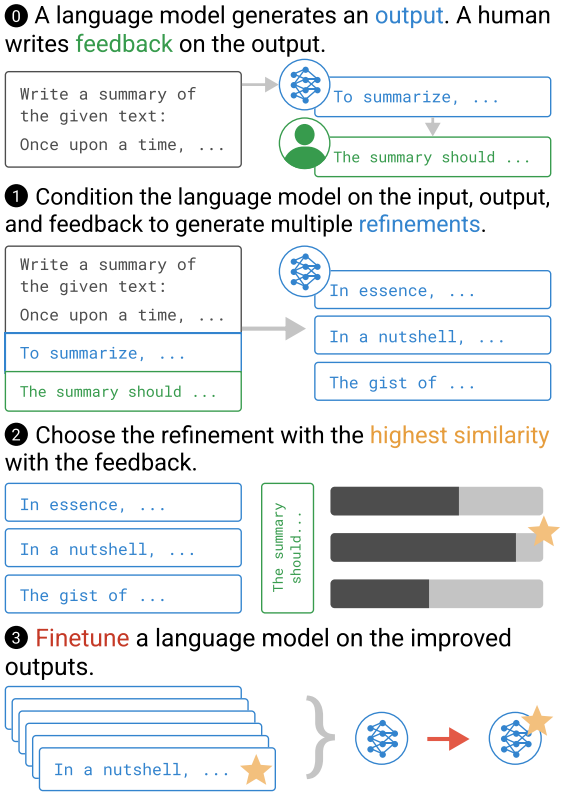

Figure 1: An overview of our algorithm for learning from natural language feedback.

We propose to use natural language feedback, which contains more information per evaluation. We introduce a three-step learning algorithm, as shown in Figure 1. First, we condition an LM on an input, model-generated output, and human-written feedback to sample many possible refinements of the output. Second, we choose the refinement with the highest embedding-based similarity with the feedback. Third, we finetune an LM on the chosen refinements. Our algorithm de-

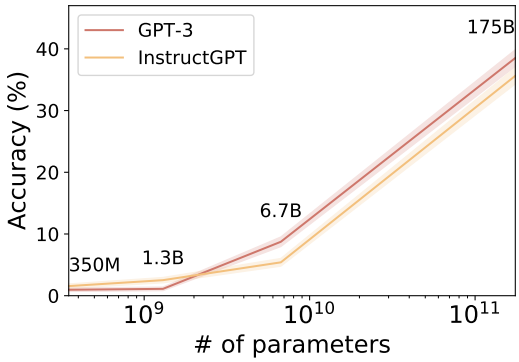

Figure 2: On the task of removing offensive words from a sentence, only large LMs incorporate feedback.

parts from prior work, which uses reinforcement learning methods (Ziegler et al., 2019, *inter alia*) or auxiliary losses (Stacey et al., 2021) that cannot be straightforwardly generalized to use natural language feedback.

We validate our algorithm on a carefully-controlled synthetic task of removing offensive words from a sentence with GPT-3-based models (Brown et al., 2020; Ouyang et al., 2022). We find that only the largest GPT-3-based models (175B parameters) accurately refine outputs. Using the above insight, we use the largest GPT-3 models to test our algorithm on text summarization, following Stiennon et al. (2020). A model trained with our algorithm generates summaries that human evaluators prefer to human reference summaries ∼51% of the time. We obtain these results when learning from only 100 samples of natural language feedback. Our analysis shows that LM-generated refinements typically do incorporate the feedback, especially when choosing the refinement with the highest similarity with the feedback. Overall, our results suggest that natural language feedback is a promising avenue for learning from human preferences.

2 Method

Here, we define our problem formulation more formally. Given an input x, we seek to generate an output y that is high quality according to human preference judgments. We assume access to natural language feedback f on an initial model-generated output y given the input x.

To tackle the above problem, we leverage the ability of pretrained LMs to follow instructions (Radford et al., 2019; Sanh et al., 2021; Wei et al., 2022; Ouyang et al., 2022). We assume access to an LM that takes an input (e.g., a task instruction) and produces a distribution over text completions (e.g., a task output). We instruct the LM to refine the initial output y given the input x and feedback f. We then sample N refinements $y'_1, \ldots, y'_N$ from the LM. Refinements may vary in quality, so we introduce a function S that scores refinements for how effectively they incorporate feedback. We choose the refinement with the highest score from S and finetune a model on all chosen y' given x. We use the resulting model to generate outputs at test time.

3 Experiments

3.1 Can Language Models Use Feedback?

For our algorithm to work, LMs must be able to accurately incorporate feedback to generate refinements. Thus, we first validate our algorithm on a carefully-controlled synthetic task of removing specific offensive words from a given sentence. We examine how effective various models are at incorporating feedback, to determine what model to use for refining outputs.

Experimental Setup We instruct an LM to refine an automatically-generated sentence with ≤ 10 offensive words by removing ≤ 3 specific words (see Appendix B for a detailed explanation and examples). We evaluate how often the generated refinement exactly matches the target sentence, which we also automatically generate. For our LMs, we use differently-sized GPT-3 models (Brown et al., 2020) and their finetuned, InstructGPT counterparts (Ouyang et al., 2022).[1] We report all hyperparameters used in Appendix E. We report mean and std. error for all results in our work.

Results Fig. 2 shows the results. Only the largest GPT-3 and InstructGPT models (175B parameters) incorporate feedback a non-trivial amount of time. Using this insight, we only use the 175B parameter models in the rest of our experiments.

3.2 Text Summarization

3.2.1 Experimental Setup

Generating Refinements We now evaluate our algorithm on the real-world task of text summarization. We follow prior work on learning from human preferences (Stiennon et al., 2020) and learn to summarize Reddit posts from Völske et al. (2017). We take 100 samples from the Reddit data subset used in Stiennon et al. (2020). We use InstructGPT

[1] Via the OpenAI API. OpenAI does not disclose the size of the provided models, so we use estimates from Eleuther.

(175B) to generate initial summaries and refinements, using the instructions in Appendix F. We then write feedback f on the initial summary y given the Reddit post x, and we generate possible refinements $y'_1, \ldots, y'_{20}$.

Scoring Refinements We choose a refinement with a scoring function S that scores refinements for how effectively they incorporate feedback. For S, we use contrastive pre-trained text embedding function $\mathcal{E}$ (Neelakantan et al., 2022) to embed the feedback f and refinements $y'_1, \ldots, y'_{20}$[2]. We then choose the refinement with the highest cosine similarity score with the feedback. We opted for high similarity with the feedback because feedback often describes what the ideal or improved text would look like. We refer to refinements generated with the above algorithm as REFINEMENT WITH FEEDBACK + BEST OF N.

Finetuning We finetune GPT-3 (175B; Brown et al., 2020)[3] on refinements generated by REFINEMENT WITH FEEDBACK + BEST OF N. We compare against finetuning on INITIAL SUMMARIES generated with InstructGPT. We also compare against summaries generated directly by InstructGPT and GPT-3 (175B). We use the same instructions as for INITIAL SUMMARIES (in Appendix F) and provide the post and its title.

Evaluation We test on 100 unseen Reddit posts from the same dataset and conduct human evaluations for all experiments.[4] Evaluators rank the summaries according to the rubric in Appendix C, with ties allowed. We show the win rate of an algorithm, counting ties as a half win, similar to Kendall rank correlation.[5]. We refer to Appendix C for a description of all human evaluation and feedback annotation procedures and Appendix D for more details about the ranking scheme.

3.2.2 Main Results

Fig. 3 reports the win rate of our learning algorithm over HUMAN SUMMARIES and Appendix Fig. 6 reports the win rate over InstructGPT. Finetuning on REFINEMENT WITH FEEDBACK + BEST OF N generates summaries on par with human summaries, with a win rate of $51 \pm 5.0\%$ over human summaries. In contrast, all baselines underperform

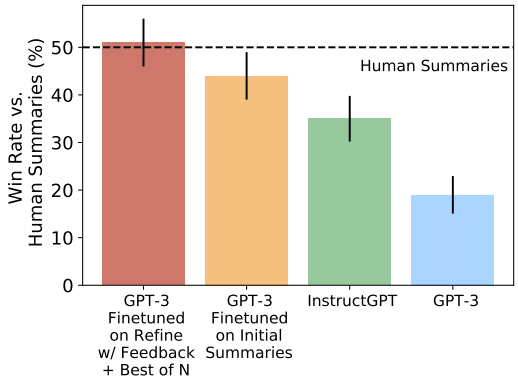

Figure 3: How often human evaluators prefer summaries from our learning algorithm and baselines to HUMAN SUMMARIES. Our proposed algorithm (leftmost bar) generates summaries of a similar quality to human summaries.

human summaries, with win rates of $19 \pm 3.9\%$ (GPT-3), $35 \pm 4.8\%$ (InstructGPT), $44 \pm 5.0\%$ (finetuning on INITIAL SUMMARIES)). In particular, our approach achieves a win rate of $57 \pm 5.0\%$ over the strongest baseline, finetuning on INITIAL SUMMARIES. Our result suggests that our learning algorithm produces higher-quality summaries by finetuning on the higher-quality targets (our refinements). Overall, we achieve strong results on summarization while learning from only 100 samples of human-written feedback.

3.2.3 Analysis

We now aim to examine the importance of various aspects of our algorithm for generating high-quality refinements (before finetuning). We evaluate REFINEMENT WITH FEEDBACK, which *randomly* chooses a refinement $\in y'_1, \ldots, y'_{20}$. This ablation helps to evaluate the importance of choosing a refinement with our scoring function S. We also evaluate REFINEMENT WITHOUT FEEDBACK, which instructs the LM to refine the initial summary but without feedback. This ablation helps to evaluate the importance of using the feedback. Lastly, we evaluate HUMAN SUMMARIES, i.e., summaries written by Reddit users on their own posts, and INITIAL SUMMARIES, i.e., the initial summary y generated by the LM. See Appendix F for concrete examples of the instructions that we use.

Fig. 4 (left) shows the win rates of refinements from various methods against INITIAL SUMMARIES. REFINEMENT WITH FEEDBACK + BEST OF N improves over the INITIAL SUMMARIES, with our algorithm being preferred $67 \pm 3.1\%$ of the time. Our algorithm is preferred $54 \pm 3.5\%$ of the time to human summaries, while INITIAL SUMMARIES are significantly worse than human summaries, preferred only $39 \pm 3.4\%$ of the time. Ap-

[2]We use OpenAI's API to access the embeddings.

[3]InstructGPT cannot yet be finetuning via OpenAI's API.

[4]We plan to conduct larger-scale human evaluations in the future, to confirm our initial findings.

[5]Kendall Rank correlation

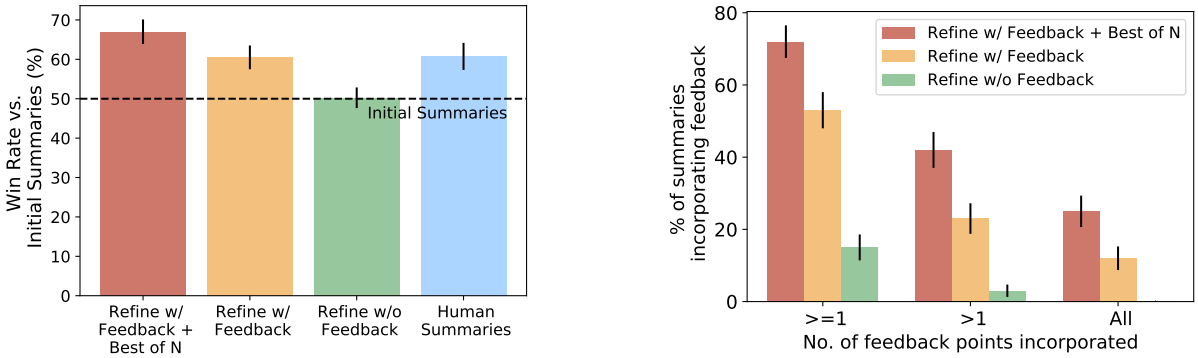

Figure 4: **Left**: How often human evaluators prefer summaries from each refinement method to the INITIAL SUMMARIES (from InstructGPT). REFINEMENT WITH FEEDBACK improves on the INITIAL SUMMARIES and outperforms human summaries with BEST OF N sampling. **Right**: Refining with feedback generally does incorporate specific point(s) mentioned in the feedback.

pendix Fig. 5 shows win rates of refinements generated with various methods against HUMAN SUMMARIES and Appendix Fig. 7 shows that refinements are more helpful when the initial summary is of lower quality. We also refer to Appendix G for 10 random examples of INITIAL SUMMARIES, feedback, and refinements from various methods. Overall, using feedback and scoring refinements are both important steps for generating high-quality refinements of the initial output.

Here, we examine whether refinements are of higher quality because they incorporate the feedback, rather than by improving the summary in other ways. To do so, we evaluate how often the refinements incorporate the human-written feedback. We evaluate (1) how often ≥ 1 point mentioned in the feedback is incorporated in the refinement, (2) how often > 1 point is incorporated, and (3) how often all of the feedback is incorporated. In Fig. 4 (right), we see that our algorithm incorporates ≥ 1 feedback point $72 \pm 4.5\%$ of the time, showing that LMs are able to incorporate feedback with high accuracy. REFINEMENTS WITHOUT FEEDBACK only incorporates at least one feedback point $15 \pm 3.6\%$ of the time. Our results suggest that refinements are high-quality because they incorporate specific points in the feedback.

4 Additional Related Work

Existing work in NLP primarily investigates using explanations for *labeled outputs* to *classification tasks*. In contrast, we do not assume access to gold-labeled outputs, and we study the more general text generation setting, which classification tasks can be formulated as (Radford et al., 2019; Raffel et al., 2020; Brown et al., 2020). Explanations describe why a labeled output is correct, while feedback describes how to improve a candidate output. Prior work explores ways of using explanations to train text classification models, with mixed results (Camburu et al., 2018; Stacey et al., 2021; Pruthi et al., 2021; Wiegreffe et al., 2021; Hase and Bansal, 2021, *inter alia*).

Outside of text domains, there is abundant work in reinforcement learning that leverages language in various ways (see Luketina et al., 2019, for an overview). This line of work generally focuses on using language to specify the task, as in instruction following (Chaplot et al., 2017; Mahmoudieh et al., 2022; Ouyang et al., 2022, *inter alia*), rather than to correct a model's behavior as in our work. Other work uses language to train the model via strong supervision (Andreas et al., 2017; Kaplan et al., 2017) or reward shaping (Goyal et al., 2019). Instead, we aim to *correct* a model's behavior using language feedback on the behavior.

5 Conclusion

In this work, we proposed an algorithm for training LMs to behave in line with human preferences, by learning from natural language feedback. We validated our approach on a carefully-controlled word-removal task, showing that only large LMs (175B parameters) accurately incorporate feedback. Using this insight, we then tested our algorithm on the real-world task for text summarization. Our finetuning algorithm brought a GPT-3 model to roughly human-level summarization ability, using only 100 samples of human feedback. Language feedback is a natural form of communicating with models which may make it easier for many people to provide informative, high-quality feedback. In the long run, our work suggests many exciting avenues for future work, e.g., in guiding models with language feedback in other domains from code generation to conversational assistance.

6 Acknowledgements

We are grateful to Nat McAleese, Geoffrey Irving, Jeff Wu, Sam Bowman, Daniel Ziegler, Seraphina Nix, and Lennart Heim for for helpful conversations and feedback. Jérémy Scheurer and Jun Shern Chan thank Open Philanthropy for funding that enabled this research. Ethan Perez thanks the National Science Foundation and Open Philanthropy for fellowship support. Jon Ander Campos is supported by a doctoral grant from the Spanish MECD. Angelica Chen and Kyunghyun Cho are supported by the NYU Center for Data Science National Science Foundation (Award 1922658). Kyunghyun Cho is also supported by Samsung Advanced Institute of Technology (Next Generation Deep Learning: from pattern recognition to AI). We also thank OpenAI for providing access and credits to their models via the API Academic Access Program.

References

Jacob Andreas, Dan Klein, and Sergey Levine. 2017. Modular multitask reinforcement learning with policy sketches. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 166–175. PMLR.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. *Advances in Neural Information Processing Systems*, 31.

Devendra Singh Chaplot, Kanthashree Mysore Sathyendra, Rama Kumar Pasumarthi, Dheeraj Rajagopal, and Ruslan Salakhutdinov. 2017. Gated-Attention Architectures for Task-Oriented Language Grounding. *arXiv preprint arXiv:1706.07230*.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A Smith. 2020. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. *arXiv preprint arXiv:2009.11462*.

Prasoon Goyal, Scott Niekum, and Raymond J. Mooney. 2019. Using Natural Language for Reward Shaping in Reinforcement Learning.

Peter Hase and Mohit Bansal. 2021. When can models learn from explanations? a formal framework for understanding the roles of explanation data. *arXiv preprint arXiv:2102.02201*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

Russell Kaplan, Christopher Sauer, and Alexander Sosa. 2017. Beating Atari with Natural Language Guided Reinforcement Learning.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2021. TruthfulQA: Measuring How Models Mimic Human Falsehoods.

Jelena Luketina, Nantas Nardelli, Gregory Farquhar, Jakob Foerster, Jacob Andreas, Edward Grefenstette, Shimon Whiteson, and Tim Rocktäschel. 2019. A survey of reinforcement learning informed by natural language. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 6309–6317. International Joint Conferences on Artificial Intelligence Organization.

Parsa Mahmoudieh, Sayna Ebrahimi, Deepak Pathak, and Trevor Darrell. 2022. Zero-Shot Reward Specification via Grounded Natural Language.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. 2021. WebGPT: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.

Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, Jerry Tworek, Qiming Yuan, Nikolas Tezak, Jong Wook Kim, Chris Hallacy, Johannes Heidecke, Pranav Shyam, Boris Power, Tyna Eloundou Nekoul, Girish Sastry, Gretchen Krueger, David Schnurr, Felipe Petroski Such, Kenny Hsu, Madeleine Thompson, Tabarak Khan, Toki Sherbakov, Joanne Jang, Peter Welinder, and Lilian Weng. 2022. Text and Code Embeddings by Contrastive Pre-Training.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Preprint*.

Romain Paulus, Caiming Xiong, and Richard Socher. 2018. A Deep Reinforced Model for Abstractive Summarization. In *International Conference on Learning Representations*.

Danish Pruthi, Rachit Bansal, Bhuwan Dhingra, Livio Baldini Soares, Michael Collins, Zachary C. Lipton, Graham Neubig, and William W. Cohen. 2021. Evaluating Explanations: How much do explanations from the teacher aid students?

Alec Radford and Karthik Narasimhan. 2018. Improving Language Understanding by Generative Pre-Training.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners.

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. 2021. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2021. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*.

Joe Stacey, Yonatan Belinkov, and Marek Rei. 2021. Supervising Model Attention with Human Explanations for Robust Natural Language Inference. *arXiv preprint arXiv:2104.08142*.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021.

Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. 2017. TL;DR: Mining Reddit to learn automatic summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 59–63, Copenhagen, Denmark. Association for Computational Linguistics.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022. Finetuned Language Models are Zero-Shot Learners. In *International Conference on Learning Representations*.

Sarah Wiegreffe, Ana Marasović, and Noah A. Smith. 2021. Measuring association between labels and free-text rationales. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10266–10284, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

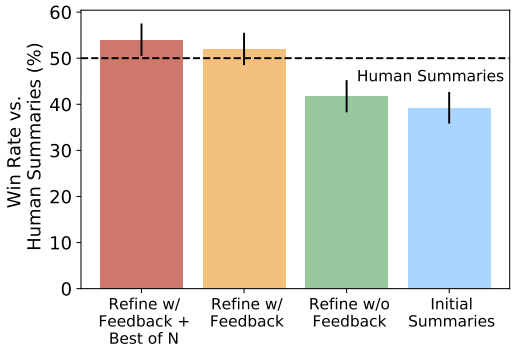

Figure 5: Win rate of various refinement methods.

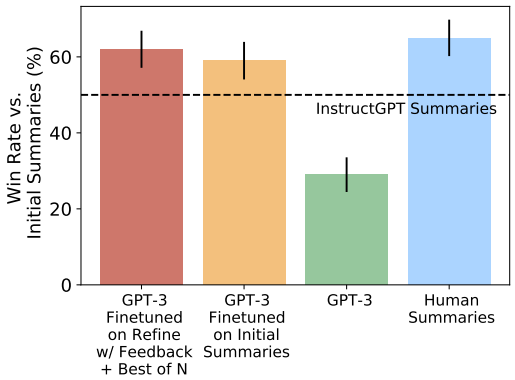

Figure 6: Win rate of our learning algorithm over InstructGPT (used to generate INITIAL SUMMARIES)

A Additional Results

Here, we report additional results. In Fig. 5, we report the win rate of our various refinement methods over HUMAN SUMMARIES. In Fig. 6, we report the win rate of our learning algorithm over InstructGPT (used to generate the INITIAL SUMMARIES).

Fig. 7 shows the win rate of various refinement methods by the rank of the INITIAL SUMMARIES relative to summaries from other methods. REFINEMENT WITH FEEDBACK (+ BEST OF N) have high win rates relative to other methods when theINITIAL SUMMARY is poorly ranked. When the INITIAL SUMMARY rank is 4, REFINEMENT WITH FEEDBACK has a win rate of $83 \pm 3.9\%$ vs. $49 \pm 5.4\%$ for REFINEMENT WITHOUT FEEDBACK. On the other hand, when the INITIAL SUMMARY is higher quality (rank 2), the win rate of REFINEMENT WITH FEEDBACK is only $8 \pm 4.0\%$ vs. $31 \pm 5.8\%$ for REFINEMENTS WITHOUT FEEDBACK. The result is intuitive, as feedback on a bad summary should be more helpful than feedback on a good summary since there is more room for improvement on a bad summary.

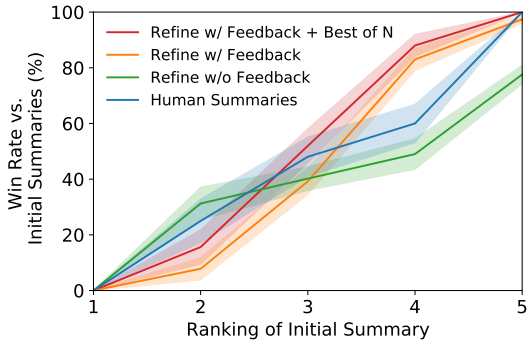

Figure 7: This plot shows the win rate of various methods against the INITIAL SUMMARIES (y-axis) given various rankings of the initial summaries (x-axis). We observe that the worse the initial summaries, the better the refinements are.

B Targeted Word Removal Details

Below is an example of how we instruct or "prompt" an LM to remove specific, offensive words from a sentence.

> *"In this text, many toxic and offensive words are used: You are such a jerk, and a nice person, and an idiot. The ideal text should remove the word jerk, but otherwise be unchanged: You are"*

Here, the target completion is *" such a nice person and an idiot."* More formally, we sample offensive sentences by using k offensive words from a fixed set of 25 offensive words, drawn uniformly at random (without replacement). Each offensive sentence also includes the words "nice person" in addition to all the offensive words. For each $k \in \{1, \ldots, 10\}$, we sample 50 offensive sentences. The task is then to remove $l \in [1, 2, 3]$ offensive words from a given sentence, with $k \geq l$. Since we include the words "nice person" in the offensive sentence, we can remove $l = k$ offensive words and still have a target sentence that intuitively makes sense.

C Human Feedback and Evaluation

Automated metrics such as ROUGE (Lin, 2004) do not correlate well with human preferences for summarization (Paulus et al., 2018; Stiennon et al., 2020), so we conduct human evaluations to evaluate various methods. We show all instructions given to annotators and evaluators here. Two of our authors wrote feedback on the INITIAL SUMMARIES. To annotate feedback, they had access to the title, post, and the INITIAL SUMMARY. One author conducted the human evaluation for how often refinement incorporates feedback (Fig. 4; right).

For this evaluation, we provided the title, post, INITIAL SUMMARY, feedback, and summaries generated by various methods. Lastly, two authors not involved in feedback annotation conducted the human evaluation of generated refinements (Fig. 4 left, Fig. 5, and Fig. 7). The annotators had access to title, post, and summaries generated with various methods. See Appendix D for more detail about our ranking procedure. One author conducted the human evaluation for Figs. 3 and 6.

D Details about Ranking Procedure

We use a standard ranking scheme where each of 5 summaries is given a rank between 1 and 5 (inclusive). The method REFINEMENT WITHOUT FEEDBACK often generates refinements that are exact copies of the INITIAL SUMMARIES, so we allow for ties in our ranking scheme. We assign the rank r' to all summaries ranked in a tie, where $r' = \frac{r+(r+n-1)}{2}$, r is the rank of the tied elements, and n is the number of ties at the rank. For example, we map a ranking of $(1, 2, 2, 4, 5) \rightarrow (1, 2.5, 2.5, 4, 5)$ and a ranking of $(1, 2, 3, 3, 3) \rightarrow (1, 2, 4, 4, 4)$.

E Hyperparameters

E.1 Generating Refinements

For the targeted word removal experiments (§3.1), we use greedy decoding until 200 tokens or */ n* is generated. For all summarization experiments (§3.2), we sample up to 48 tokens (as in Stiennon et al., 2020) with nucleus sampling (Holtzman et al., 2019) with $p = 0.9$. We strip non-alphanumeric characters (e.g., newlines) from the beginning of sampled summaries. Due to the maximum token length, sampled summaries sometimes end with an incomplete sentence. Thus, we remove ending sentences that do not end in ".", "!", or "?".

E.2 Finetuning

We finetune GPT-3 with 175B parameters on REFINEMENTS WITH FEEDBACK + BEST OF N and INITIAL SUMMARIES. We use a maximum batch size of 256 and the default of 4 epochs, as recommended by the OpenAI API. For other hyperparameters, we conduct a hyperparameter search using 5-fold cross-validation on our train dataset of 100 examples. We use OpenAI's default hyperparameter settings from the OpenAI API and search over the *learning rate multiplier*, the multiplier on the pretraining learning rate used to obtain the fine-tuning learning rate. We sweep over $[0.005, 0.01, 0.025, 0.05, 0.1, 0.2]$ and choose 0.05. We also use OpenAI's default hyperparameter settings and sweep over the *prompt loss weight* or the weight used for language modeling loss on the prompt (Radford and Narasimhan, 2018). We sweep over $[0.01, 0.025, 0.05, 0.1, 0.2]$ and choose 0.01. We then finetune a new model on the full 100 examples with the chosen hyperparameters. We refer to the API documentation for a more detailed description of the adjustable hyperparameters provided by OpenAI.

F Prompt Templates

Table 1 shows the prompts used in our experiments.

G Examples

Table 2 shows summaries of 10 random Reddit posts from various methods: initial model-generated summaries and refinement methods.

Methods	Format
INITIAL SUMMARY	Write an excellent summary of a given text. An excellent summary is coherent, accurate, concise, and detailed, and it follows human preferences about summaries. TITLE: {`title`} Text: {`text`} TL;DR:
REFINEMENT WITH FEEDBACK	Given a text, an initial summary, and feedback on the initial summary, write an improved summary that incorporates the feedback on the initial summary and is better than the initial summary. The improved summary is coherent, accurate, concise, and detailed, and it follows human preferences about summaries. Most importantly, the improved summary incorporates the feedback. TITLE: {`title`} Text: {`text`} Summary: {`summary`} Feedback: {`feedback`} TL;DR:
REFINEMENT WITHOUT FEEDBACK	Given a text and an initial summary, write an improved summary that is better than the initial summary. The improved summary is coherent, accurate, concise, and detailed, and it follows human preferences about summaries. TITLE: {`title`} Text: {`text`} Summary: {`summary`} TL;DR:

Table 1: Prompt templates used for various methods.

Example 1

Initial Summary
Should the author ask her committed boyfriend if he lied about visiting an old flame?

Feedback
The summary is good but should mention more details. Concretely, it should remove the word "committed", since that isn't emphasized in the text. The summary should also mention that the author believes her boyfriend visited an old flame very early on in their relationship, i.e. after 4 months. Further the summary should mention that her boyfriend has already denied that this happened when the author asked him when they were both drunk. It should also convey that the boyfriend has lied before. The author is asking if she should mention this issue or not, since it would bring peace but she already had brougth it up once.

Refinement with Feedback + Best of N
The author is considering asking her boyfriend if he lied about visiting an old flame early on in their relationship, as this has been bothering her. He already denied it when asked drunk, and he has lied to her before.

Refinement with Feedback
The author is considering asking her boyfriend if he lied about visiting an old flame early on in their relationship, after he denied it when asked while drunk. She is worried about how this would add to the list of his past lies.

Refinement without Feedback
The author is considering asking her committed boyfriend if he lied about visiting an old flame, and she is torn about whether or not to do so.

Human Summary
Asked my bf once if he went to go meet a girl, but i think he lied. Should I ask if he lied?

Example 2

Initial Summary
my boyfriend invited me to spend Mother's Day with his parents but I feel weird about it because I don't want to imply that I will become their daughter one day. Is this too soon or am I overthinking it?

Feedback
The summary is mostly correct but it should mention that one of the reason why the original poster feels weird about all this situation is because their relationship is just starting and they haven't talked about a future together yet. It is also important that the original poster can't spend they Mother's day with her mom and this is one of the reasons why her boyfriend has invited her.

Refinement with Feedback + Best of N

The original poster is feeling weird about spending Mother's Day with their boyfriend's parents because they haven't talked about a future together yet. They also can't spend Mother's Day with their own mom.

Refinement with Feedback
The original poster feels weird about spending Mother's day with their boyfriend's parents because they haven't talked about their relationship yet, but they should go because it will be a good opportunity to get to know them better.

Refinement without Feedback
My boyfriend invited me to spend Mother's Day with his parents, but I feel weird about it because I don't want to imply that I will become their daughter one day. Is this too soon or am I overthinking it?

Human Summary
my boyfriend and I have been together 8 months and he invited me to spend Mother's Day with his mom and dad, but I feel uncomfortable too soon?

Example 3

Initial Summary
Girl (F/19) doesn't know what to do with guy (M/21) who is good and polite, but not romantic and doesn't want anything serious.

Feedback
The summary should make clear that the girl is already together with the guy for 7 months. The summary should also point out that the guy is not passionate and doesn't want sex. Lastly the summary should convey that the author is frustrated by the fact that the guy doesn't want anything serious and says he doesn't want to go fast, but that she also thinks she's in love.

Refinement with Feedback + Best of N
Girl (F/19) is dating guy (M/21) who is good and polite but not romantic and doesn't want anything serious. Girl is frustrated because she wants more than what the guy is willing to give.

Refinement with Feedback
Girl (F/19) is frustrated with guy (M/21) because he is good and polite, but not romantic and doesn't want anything serious.

Refinement without Feedback
Girl (F/19) doesn't know what to do with guy (M/21) who is good and polite, but not romantic and doesn't want anything serious.

Human Summary
He (21) is a good guy, but I'm afraid he doesn't want anything serious with me (19). How should I react?

Table 2: Examples of summaries, human-written feedback, and generated refinements.