Fine-Tuning Language Models from Human Preferences

Daniel M. Ziegler[*]   **Nisan Stiennon**[*]   **Jeffrey Wu**   **Tom B. Brown**
Alec Radford   **Dario Amodei**   **Paul Christiano**   **Geoffrey Irving**
OpenAI
{dmz,nisan,jeffwu,tom,alec,damodei,paul,irving}@openai.com

Abstract

Reward learning enables the application of reinforcement learning (RL) to tasks where reward is defined by human judgment, building a model of reward by asking humans questions. Most work on reward learning has used simulated environments, but complex information about values is often expressed in natural language, and we believe reward learning for language is a key to making RL practical and safe for real-world tasks. In this paper, we build on advances in generative pretraining of language models to apply reward learning to four natural language tasks: continuing text with positive sentiment or physically descriptive language, and summarization tasks on the TL;DR and CNN/Daily Mail datasets. For stylistic continuation we achieve good results with only 5,000 comparisons evaluated by humans. For summarization, models trained with 60,000 comparisons copy whole sentences from the input but skip irrelevant preamble; this leads to reasonable ROUGE scores and very good performance according to our human labelers, but may be exploiting the fact that labelers rely on simple heuristics.

1. Introduction

We would like to apply reinforcement learning to complex tasks defined only by human judgment, where we can only tell whether a result is good or bad by asking humans. To do this, we can first use human labels to train a model of reward, and then optimize that model. While there is a long history of work learning such models from humans through interaction, this work has only recently been applied to modern deep learning, and even then has only been applied to relatively simple simulated environments (Christiano et al., 2017; Ibarz et al., 2018; Bahdanau et al., 2018). By contrast, real world settings in which humans need to specify com-

plex goals to AI agents are likely to both involve and require natural language, which is a rich medium for expressing value-laden concepts. Natural language is particularly important when an agent must communicate back to a human to help provide a more accurate supervisory signal (Irving et al., 2018; Christiano et al., 2018; Leike et al., 2018).

Natural language processing has seen substantial recent advances. One successful method has been to pretrain a large generative language model on a corpus of unsupervised data, then fine-tune the model for supervised NLP tasks (Dai and Le, 2015; Peters et al., 2018; Radford et al., 2018; Khandelwal et al., 2019). This method often substantially outperforms training on the supervised datasets from scratch, and a single pretrained language model often can be fine-tuned for state of the art performance on many different supervised datasets (Howard and Ruder, 2018). In some cases, fine-tuning is not required: Radford et al. (2019) find that generatively trained models show reasonable performance on NLP tasks with no additional training (zero-shot).

There is a long literature applying reinforcement learning to natural language tasks. Much of this work uses algorithmically defined reward functions such as BLEU for translation (Ranzato et al., 2015; Wu et al., 2016), ROUGE for summarization (Ranzato et al., 2015; Paulus et al., 2017; Wu and Hu, 2018; Gao et al., 2019b), music theory-based rewards (Jaques et al., 2017), or event detectors for story generation (Tambwekar et al., 2018). Nguyen et al. (2017) used RL on BLEU but applied several error models to approximate human behavior. Wu and Hu (2018) and Cho et al. (2019) learned models of coherence from existing text and used them as RL rewards for summarization and long-form generation, respectively. Gao et al. (2019a) built an interactive summarization tool by applying reward learning to one article at a time. Experiments using human evaluations as rewards include Kreutzer et al. (2018) which used off-policy reward learning for translation, and Jaques et al. (2019) which applied the modified Q-learning methods of Jaques et al. (2017) to implicit human preferences in dialog. Yi et al. (2019) learned rewards from humans to fine-tune dialog models, but smoothed the rewards to allow supervised learning. We refer to Luketina et al. (2019) for a survey of

---

[*]Equal contribution. Correspondence to paul@openai.com.

RL tasks involving language as a component, and for RL results using transfer learning from language. RL is not the only way to incorporate ongoing human feedback: Hancock et al. (2019) ask humans what a dialogue system should have said instead, then continue supervised training.

In this paper, we combine the pretraining advances in natural language processing with human preference learning. We fine-tune pretrained language models with reinforcement learning rather than supervised learning, using a reward model trained from human preferences on text continuations. Following Jaques et al. (2017; 2019), we use a KL constraint to prevent the fine-tuned model from drifting too far from the pretrained model. We apply our method to two types of tasks: continuing text in a way that matches a target style, either positive sentiment or vividly descriptive, and summarizing text from the CNN/Daily Mail or TL;DR datasets (Hermann et al., 2015; Völske et al., 2017). Our motivation is NLP tasks where supervised data sets are unavailable or insufficient, and where programmatic reward functions are poor proxies for our true goals.

For stylistic continuation, 5,000 human comparisons (each choosing the best of 4 continuations) result in the fine-tuned model being preferred by humans 86% of the time vs. zero-shot and 77% vs. fine-tuning to a supervised sentiment network. For summarization, we use 60,000 human samples to train models that can roughly be described as "smart copiers": they typically copy whole sentences from the input, but vary what they copy to skip irrelevant initial text. This copying behavior emerged naturally from the data collection and training process; we did not use any explicit architectural mechanism for copying as in See et al. (2017); Gehrmann et al. (2018). One explanation is that copying is an easy way to be accurate, given that we did not instruct labelers to penalize copying but do instruct them to penalize inaccuracy. It may also reflect the fact that some labelers check for copying as a fast heuristic to ensure a summary is accurate. Indeed, human labelers significantly prefer our models to supervised fine-tuning baselines and even to human-written reference summaries, but not to a lead-3 baseline which copies the first three sentences.

For summarization, we continue to collect additional data and retrain our reward model as the policy improves (*online* data collection). We also test *offline* data collection where we train the reward model using data from the original language model only; offline data collection significantly reduces the complexity of the training process. For the TL;DR dataset, human labelers preferred the policy trained with online data collection 71% of the time, and in qualitative evaluations the offline model often provides inaccurate summaries. In contrast, for stylistic continuation we found that offline data collection worked similarly well. This may be related to the style tasks requiring very little data; Radford
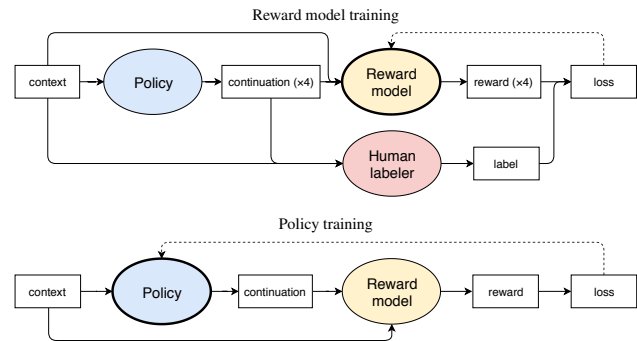

Figure 1: Our training processes for reward model and policy. In the online case, the processes are interleaved.

et al. (2017) show that generatively trained models can learn to classify sentiment from very few labeled examples.

In concurrent work, Böhm et al. (2019) also use human evaluations to learn a reward function for summarization, and optimize that reward function with RL. Their work provides a more detailed investigation of the learned policy and reward function on the CNN/Daily Mail dataset, while we are interested in exploring learning from human feedback more generally and at larger computational scale. So we consider several additional tasks, explore the effects of on-policy reward model training and more data, and fine-tune large language models for both reward modeling and RL.

2. Methods

We begin with a vocabulary Σ and a language model ρ which defines a probability distribution over sequences of tokens Σ^n via

$$\rho(x_0 \cdots x_{n-1}) = \prod_{0 \leq k < n} \rho(x_k | x_0 \cdots x_{k-1})$$

We will apply this model to a task with input space $X = \Sigma^{\leq m}$, data distribution $\mathcal{D}$ over X, and output space $Y = \Sigma^n$. For example, $x \in X$ could be an article of up to 1000 words and $y \in Y$ could be a 100-word summary. ρ defines a probabilistic policy for this task via $\rho(y|x) = \rho(xy)/\rho(x)$: fixing the beginning of the sample to x and generating subsequent tokens using ρ.

We initialize a policy $\pi = \rho$, and then fine-tune π to perform the task well using RL. If the task was defined by a reward function $r : X \times Y \to \mathbb{R}$, then we could use RL to directly optimize the expected reward:

$$\mathbb{E}_\pi [r] = \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi(\cdot | x)} [r(x, y)]$$

However, we want to perform tasks defined by human judgments, where we can only learn about the reward by asking humans. To do this, we will first use human labels to train a reward model, and then optimize that reward model.

Following Christiano et al. (2017), we ask human labelers to pick which of several values of y_i is the best response to a given input x.[1] We ask humans to choose between four options (y_0, y_1, y_2, y_3); considering more options allows a human to amortize the cost of reading and understanding the prompt x. Let $b \in \{0, 1, 2, 3\}$ be the option they select. Having collected a dataset S of $(x, y_0, y_1, y_2, y_3, b)$ tuples, we fit a reward model $r : X \times Y \to \mathbb{R}$ using the loss

$$\text{loss}(r) = \mathbb{E}_{(x,\{y_i\}_i,b)\sim S} \left[\log \frac{e^{r(x,y_b)}}{\sum_i e^{r(x,y_i)}} \right] \quad (1)$$

Since the reward model needs to understand language, we initialize it as a random linear function of the final embedding output of the language model policy ρ following Radford et al. (2018) (see section 4.2 for why we initialize from ρ rather than π). To keep the scale of the reward model consistent across training, we normalize it so that it has mean 0 and variance 1 for $x \sim \mathcal{D}, y \sim \rho(\cdot|x)$.

Now we fine-tune π to optimize the reward model r. To keep π from moving too far from ρ, we add a penalty with expectation $\beta \, \text{KL}(\pi, \rho)$ (see table 10 for what happens without this). That is, we perform RL on the modified reward

$$R(x, y) = r(x, y) - \beta \log \frac{\pi(y|x)}{\rho(y|x)}. \quad (2)$$

We either choose a constant β or vary it dynamically to achieve a particular value of $\text{KL}(\pi, \rho)$; see section 2.2. This term has several purposes: it plays the role of an entropy bonus, it prevents the policy from moving too far from the range where r is valid, and in the case of our style continuation tasks it also is an important part of the task definition: we ask humans to evaluate style, but rely on the KL term to encourage coherence and topicality.

Our overall training process is:

1. Gather samples (x, y_0, y_1, y_2, y_3) via $x \sim \mathcal{D}, y_i \sim \rho(\cdot|x)$. Ask humans to pick the best y_i from each.

2. Initialize r to ρ, using random initialization for the final linear layer of r. Train r on the human samples using loss (1).

3. Train π via Proximal Policy Optimization (PPO, Schulman et al. (2017)) with reward R from (2) on $x \sim \mathcal{D}$.

4. In the online data collection case, continue to collect additional samples, and periodically retrain the reward model r. This is described in section 2.3.

---

[1] In early experiments we found that it was hard for humans to provide consistent fine-grained quantitative distinctions when asked for an absolute number, and experiments on synthetic tasks confirmed that comparisons were almost as useful.

2.1. Pretraining details

We use a 774M parameter version of the GPT-2 language model in Radford et al. (2019) trained on their WebText dataset and their 50,257 token invertible byte pair encoding to preserve capitalization and punctuation (Sennrich et al., 2015). The model is a Transformer with 36 layers, 20 heads, and embedding size 1280 (Vaswani et al., 2017).

For stylistic continuation tasks we perform supervised fine-tuning of the language model to the BookCorpus dataset of Zhu et al. (2015) prior to RL fine-tuning; we train from scratch on WebText, supervised fine-tune on BookCorpus, then RL fine-tune to our final task. To improve sample quality, we use a temperature of $T < 1$ for all experiments; we modify the initial language model by dividing logits by T, so that future sampling and RL with $T = 1$ corresponds to a lower temperature for the unmodified pretrained model.

2.2. Fine-tuning details

Starting with the pretrained language model, the reward model is trained using the Adam optimizer (Kingma and Ba, 2014) with loss (1). The batch size is 8 for style tasks and 32 for summarization, and the learning rate is 1.77×10^{-5} for both. We use a single epoch to avoid overfitting to the small amount of human data, and turn off dropout.

For training the policy π, we use the PPO2 version of Proximal Policy Optimization from Dhariwal et al. (2017). We use 2M episodes $(x, y$ pairs$)$, $\gamma = 1$, four PPO epochs per batch with one minibatch each, and default values for the other parameters. We use batch size 1024 for style tasks and 512 for summarization. We do not use dropout for policy training. The learning rate was 1.41×10^{-5} for style tasks and 7.07×10^{-6} for summarization.

Models trained with different seeds and the same KL penalty β sometimes end up with quite different values of $\text{KL}(\pi, \rho)$, making them hard to compare. To fix this, for some experiments we dynamically vary β to target a particular value of $\text{KL}(\pi, \rho)$ using the log-space proportional controller

$$e_t = \text{clip}\left(\frac{\text{KL}(\pi_t, \rho) - \text{KL}_{\text{target}}}{\text{KL}_{\text{target}}}, -0.2, 0.2 \right)$$
$$\beta_{t+1} = \beta_t (1 + K_\beta e_t)$$

We used $K_\beta = 0.1$.

For supervised fine-tuning baselines, we fine-tune for 1 epoch on the CNN/Daily Mail and TL;DR training sets (for TL;DR we removed 30K examples to serve as a validation set). We decayed the learning rate to 0 with a cosine schedule; for the initial value, we swept over 8 log-linearly spaced options between 10^{-4} and 3×10^{-4}. We also experimented with different dropout rates, and found a rate of 0.1 to work best. We then chose the model with the best validation loss.

2.3. Online data collection

If the trained policy π is very different from the zero-shot policy ρ, the reward model will suffer a large distributional shift from training on samples from ρ to evaluation on samples from π. To prevent this, we can collect human data throughout RL fine-tuning, continuously gathering new data by sampling from π and retraining the reward model. As section 3 shows, online data collection was important for summarization but not for the simpler style tasks.

In the online case, we will choose a function $l(n)$ describing how many labels we want before beginning the nth PPO episode. Let $N_\pi = 2 \times 10^6$ be the total number of PPO episodes, $N_r^0 = l(0)$ be an initial number of human labels, and N_r be the total number of human labels. We take

$$l(n) = N_r^0 + (N_r - N_r^0)\left(1 - (1 - n/N_\pi)^2\right)$$

We pause before the nth PPO episode if we have fewer than $l(n)$ labels. We send another batch of requests to the labelers if the total requests so far is less than $l(n) + 1000$, to ensure they have at least 1000 outstanding queries at any time. We train the reward model before the first PPO episode, and then retrain it 19 more times at evenly spaced values of $l(n)$. Each time we retrain we reinitialize r to a random linear layer on top of ρ and do a single epoch through the labels collected so far. The offline case is the limit $N_r = N_r^0$.

To estimate overall progress, we gather validation samples consisting of $x \sim \mathcal{D}; y_0, y_1 \sim \rho(\cdot|x); y_2, y_3 \sim \pi(\cdot|x)$ at a constant rate; human labels on these give how often π beats ρ. Since validation samples are only used to evaluate the current π, we can add them to the training set for r. In order to estimate inter-labeler agreement, 5% of queries are answered 5 times by different labelers. Label counts in section 3 include validation samples and repeated labels.

2.4. Human labeling

We use Scale AI to collect labels. The Scale API accepts requests of the form (x, y_0, y_1, y_2, y_3) and returns selections $b \in \{0, 1, 2, 3\}$. We describe the task to Scale through a combination of instructions (appendix A) and a dataset of about 100 example comparisons from the authors.

Unlike many tasks in ML, our queries do not have unambiguous ground truth, especially for pairs of similar outputs (which play a large role in our training process, since we train r on pairs of labels sampled from a single policy π). This means that there is significant disagreement even between labelers who have a similar understanding of the task and are trying to rate consistently. On 4-way comparisons for sentiment and TL;DR summarization, authors of this paper agree about 60% of the time (vs. 25% for random guessing). This low rate of agreement complicates the quality control process for Scale; the authors agree with Scale

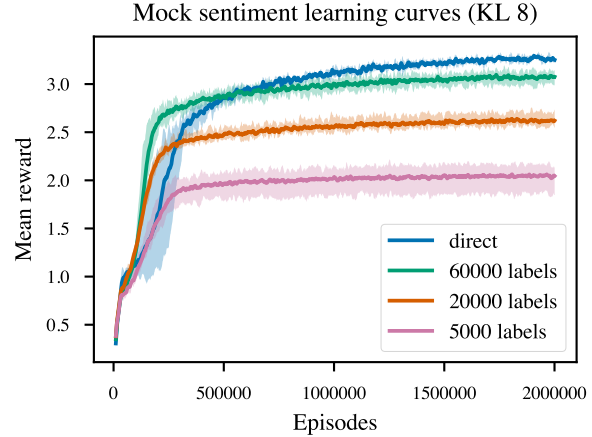Mock sentiment learning curves (KL 8)

Figure 2: Learning curves for a 124M-parameter model with mock sentiment reward, targeting a KL of 8 nats. Lines and shaded areas show mean and range for 5 seeds. Early on the reward model sometimes speeds up training, a phenomenon also observed by Christiano et al. (2017).

labelers 38% of the time on sentiment and 46% of the time on TL;DR summarization. We give further details of the human data collection and quality evaluation in appendix B.

For final evaluation of two models A and B, we generate either 2-way comparisons between pairs ($a \sim A, b \sim B$) or 4-way comparisons with quadruples ($a_0, a_1 \sim A, b_0, b_1 \sim B$), randomize the order in which samples are presented, and present these comparisons to Scale. Evaluating the quality of a model trained by Scale using the same set of humans from Scale is perilous: it demonstrates that r and π have succeeded in fitting to the human reward, but does not show that those human evaluations capture what we really care about, and our models are incentivized to exploit idiosyncrasies of the labeling process. We include samples from our models so that readers can judge for themselves.

3. Experiments

In section 3.1.1, we test our approach to RL fine-tuning of language models by using a mock labeler (a sentiment model trained on a review classification problem) as a stand-in for human labels. We show that RL fine-tuning is effective at optimizing this complex but somewhat artificial reward. In section 3.1.2, we show that we can optimize language models from human preferences on stylistic continuation tasks (sentiment and physical descriptiveness) with very little data, and that in the sentiment case the results are preferred to optimizing the review sentiment model. In section 3.2 we apply RL fine-tuning to summarization on the CNN/Daily Mail and TL;DR datasets, show that the resulting models are essentially "smart copiers", and discuss these results in the context of other summarization work.
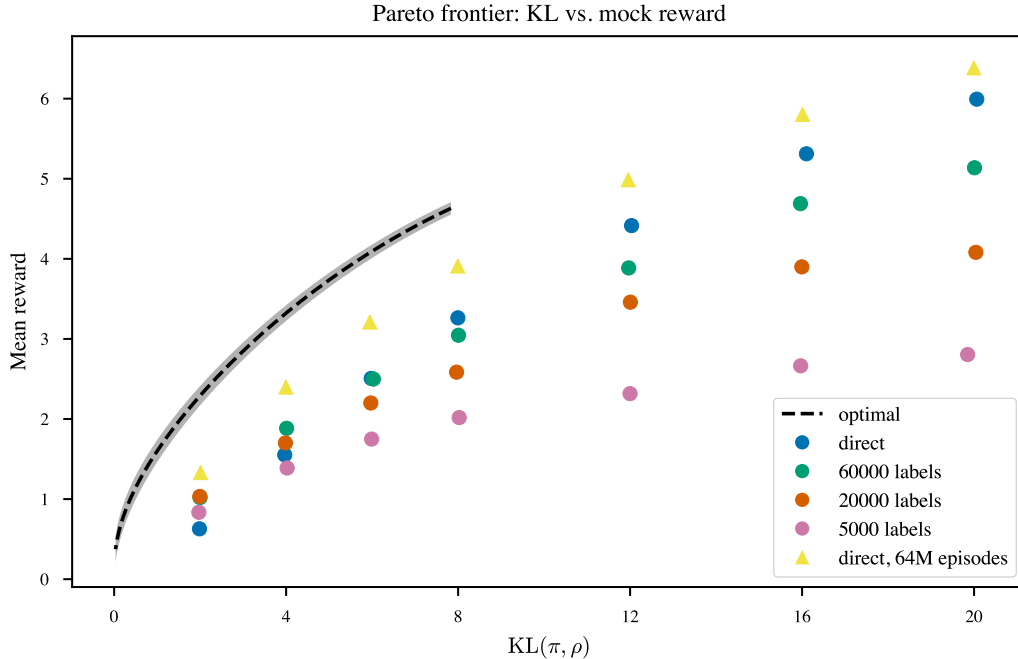
Pareto frontier: KL vs. mock reward

Figure 3: Allowing the policy π to move further from the initial policy ρ as measured by $\mathrm{KL}(\pi, \rho)$ achieves higher reward at the cost of less natural samples. Here we show the optimal KL vs. reward for 124M-parameter mock sentiment (as estimated by sampling), together with results using PPO. Runs used 2M episodes, except for the top series.

We [release code](#)[2] for reward modeling and fine-tuning in the offline data case. Our public version of the code only works with a smaller 124M parameter model with 12 layers, 12 heads, and embedding size 768. We include fine-tuned versions of this smaller model, as well as some of the human labels we collected for our main experiments (note that these labels were collected from runs using the larger model).

3.1. Stylistic continuation tasks

We first apply our method to stylistic text continuation tasks, where the policy is presented with an excerpt from the Book-Corpus dataset ([Zhu et al., 2015](#)) and generates a continuation of the text. The reward function evaluates the style of the concatenated text, either automatically or based on human judgments. We sample excerpts with lengths of 32 to 64 tokens, and the policy generates 24 additional tokens. We set the temperature of the pretrained model to $T = 0.7$ as described in [section 2.1](#).

3.1.1. MOCK SENTIMENT TASK

To study our method in a controlled setting, we first apply it to optimize a known reward function r_s designed to reflect some of the complexity of human judgments. We construct

r_s by training a classifier[3] on a binarized, balanced subsample of the Amazon review dataset of [McAuley et al.](#) ([2015](#)). The classifier predicts whether a review is positive or negative, and we define $r_s(x, y)$ as the classifier's log odds that a review is positive (the input to the final sigmoid layer).

Optimizing r_s without constraints would lead the policy to produce incoherent continuations, but as described in [section 2.2](#) we include a KL constraint that forces it to stay close to a language model ρ trained on BookCorpus.

The goal of our method is to optimize a reward function using only a small number of queries to a human. In this mock sentiment experiment, we simulate human judgments by assuming that the "human" always selects the continuation with the higher reward according to r_s, and ask how many queries we need to optimize r_s.

[Figure 2](#) shows how r_s evolves during training, using either direct RL access to r_s or a limited number of queries to train a reward model. 20k to 60k queries allow us to optimize r_s nearly as well as using RL to directly optimize r_s.

Because we know the reward function, we can also analytically compute the optimal policy and compare it to our learned policies. With a constraint on the KL divergence

---

[3]The model is a Transformer with 6 layers, 8 attention heads, and embedding size 512.
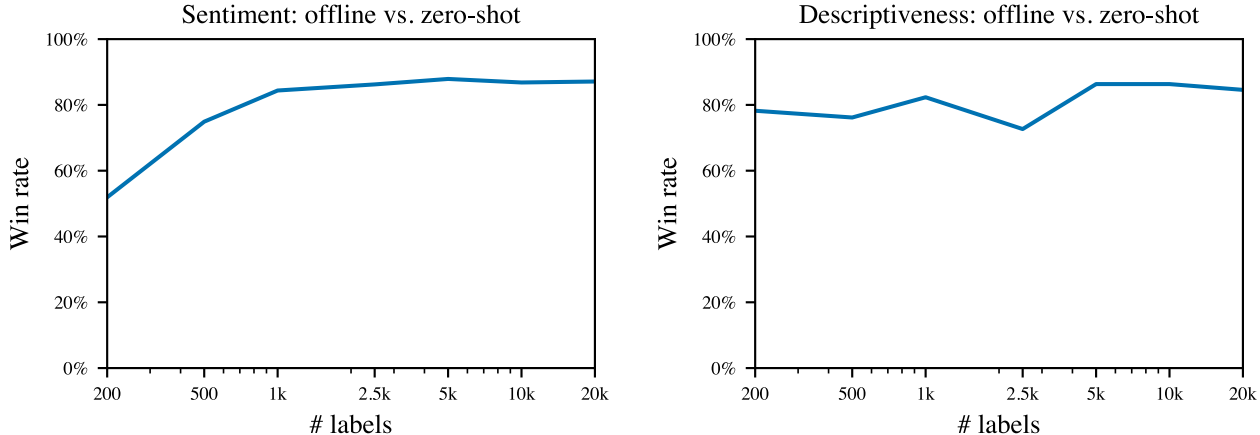
Figure 4: Human evaluations comparing the zero-shot model with offline fine-tuned models using varying amounts of human data. We report how often the fine-tuned model is preferred by a majority of 3 labelers. We omit error bars because we lack an estimate of the largest source of variance (randomness across training runs).

	Sentiment		Descriptiveness	
5k offline vs. zero-shot	**88%**	12%	**86%**	14%
5k offline vs. mock	**77%**	23%	—	
5k offline vs. 20k offline	48%	52%	47%	53%
5k offline vs. 5k online	50%	50%	48%	52%

Table 1: Human evaluations for the sentiment and descriptiveness tasks. We sample 1024 excerpts from the BookCorpus test set and report how often each model's continuations were preferred, as judged by a majority of 3 labelers.

KL(π, ρ) between the learned policy π and the language model ρ, the optimal policy has the form:

$$\pi_{\text{opt}}(y|x) \propto \rho(y|x)e^{r_s(x,y)/\beta}$$

We approximate the reward of this policy for given x and β by sampling a large number of continuations from $\rho(y|x)$ and reweighting them by $e^{r_s(x,y)/\beta}$. Figure 3 compares the reward obtained by our policies to the estimated optimal reward across a range of KL values. There is a significant gap from optimality after training the policy on 2M continuations—the number used in our main experiments—though it is largely closed with more training. Our policies continue to receive higher rewards for larger KL divergences, where we cannot afford to approximate π_{opt} by sampling.

3.1.2. HUMAN EVALUATIONS OF CONTINUATIONS

We apply our method to two continuation tasks defined by human judgments:

Sentiment: Humans are asked to reward "positive and happy" continuations.

Descriptiveness: Humans are asked to reward "vividly descriptive" continuations.

The human labelers are presented with a BookCorpus excerpt and four possible continuations; they are asked to select the best continuation. Full instructions for labelers are provided in appendix A (although labelers also learned from ~ 50 example comparisons labeled by the authors and so the instructions do not completely define the task).

To make the labeling task more natural, we select excerpts that start and end with a period. When sampling continuations that will be presented to humans, we use rejection sampling to ensure there is a period between tokens 16 and 24 and then truncate at that period.[4] During the RL fine-tuning, we penalize continuations that don't have such a period by giving them a fixed reward of -1.

We dynamically adjusted β to obtain a KL divergence of 6 nats for descriptiveness and 10 nats for sentiment (section 2.2).

We trained a range of models using different amounts of feedback, testing both offline data collection where humans

---

[4]This is a crude approximation for "end of sentence." We chose it because it is easy to integrate into the RL loop, and even a crude approximation is sufficient for the intended purpose of making the human evaluation task somewhat easier.

context	Pearl thought to herself that what they were about to do was exactly the sort of thing that they could do to help the villagers. They were all terrified of these guys. At the police station the three walked up to the counter behind which was a senior constable studying some papers.		
	Continuation 1	*Continuation 2*	*Continuation 3*
zero-shot	"Hello, I'm Pearl and this is my friend, Mike," said Pearl.	"May we speak to the police officer, sir?" asked the one in charge.	'Hello, can I help you?' 'Yes, we're the same people that the people were talking about.
5k offline fine-tune	He turned to them with a smile. "Good afternoon, ladies. I'm Detective Inspector Jones.	The constable stood up and smiled as he saw them, obviously pleased to see them.	He smiled at them and waved them in, his eyes twinkling as he listened to their tales.

Table 2: Three random ($T = 0.7$) continuations for our sentiment continuation task. Chosen from appendix table 11; see appendix for more.

context	"I do not know if it was Viking related, but it could have been." "Really?" Ailia said. Is it safe to be traveling here then? Ailia looked behind her to make sure they weren't being followed.		
	Continuation 1	*Continuation 2*	*Continuation 3*
zero-shot	There were no signs of anyone. "It is safe enough," Ailios said.	"Because I have a friend that is in the area and he will be coming with us.	It was hard to see that far. "I do not like that word.
5k offline fine-tune	Kaya crouched low, her eyes wide in the moonlight. Her body was tense.	She put her hand on the sword strapped to her back, and then pulled it out.	She strode out the door and walked down the street, her nose wrinkled in disapproval.

Table 3: Three random ($T = 0.7$) continuations for our descriptiveness continuation task. Chosen from appendix table 12; see appendix for more.

rate only the initial language model's continuation, and online data collection where humans continuously rate the current policy's continuations (section 2.3). We then compared these different policies to each other and to the zero-shot performance of the original language model. The results are shown in fig. 4 and table 1. Each model comparison is based on 1024 four-way continuation comparisons, two from each of the models being compared, each rated by 3 humans.

For these continuation tasks, offline and online data collection give similar performance. We find that very little human data is required for fine-tuning: performance with 5k, 10k, and 20k reward model training samples is similar, degrading only for less than 5k samples.[5] The model trained using the review sentiment classifier from section 3.1.1 does poorly relative to models optimized using human preference: in 77% of contexts, labelers preferred the output of the model trained with real human feedback.

---

[5]The descriptiveness policy trained with 2.5k samples performed poorly, but we believe this is due to randomness in RL.

3.2. Summarization

We also applied our method to two summarization tasks: the CNN/Daily Mail dataset of Hermann et al. (2015) and the TL;DR dataset of Völske et al. (2017). We sample articles or Reddit posts, truncate to 500 tokens, add a `"\n\nTL;DR:"` suffix (and for CNN/Daily Mail, a `"Article:\n\n"` prefix) and let the policy respond with up to 75 tokens. We set the temperature of the pretrained model to $T = 0.5$ for CNN/Daily Mail and $T = 0.7$ for TL;DR. To make the task more natural for humans, we ensure articles consist of whole sentences by truncating to the last newline character. When sampling summaries that will be shown to a human, we use rejection sampling to ensure there is a newline between tokens 55 and 75 and truncate at that newline. During RL fine-tuning, we penalize summaries that don't have such a newline by giving them a fixed score of -1. For CNN/Daily Mail we used a fixed KL coefficient $\beta = 0.1$; for TL;DR we used $\beta = 0.03$.

For RL fine-tuning, we trained online data collection mod-

	TL;DR				CNN/Daily Mail			
	R-1	R-2	R-L	R-AVG	R-1	R-2	R-L	R-AVG
SOTA	22*	5*	17*	14.7*	41.22	18.68	38.34	32.75
lead-3	17.435	3.243	14.575	11.751	**40.379**	**17.658**	36.618	31.552
zero-shot	15.862	2.325	13.518	10.568	28.406	8.321	25.175	20.634
supervised baseline	17.535	3.124	14.969	11.877	39.525	16.992	36.728	31.082
supervised + 60k fine-tune	**18.434**	**3.542**	**15.457**	**12.478**	40.093	17.611	**37.104**	**31.603**
60k fine-tune	16.800	2.884	14.011	11.232	37.385	15.478	33.330	28.731
30k fine-tune	16.410	2.920	13.653	10.994	35.581	13.662	31.734	26.992
15k fine-tune	15.275	2.240	12.872	10.129	38.466	15.960	34.468	29.631
60k offline fine-tune	16.632	2.699	13.984	11.105	33.860	12.850	30.018	25.576

Table 4: ROUGE evaluations of summarization models. For all models (excluding the lead-3 baselines), we sample with temperature 0.7 for TL;DR and 0.5 for CNN/Daily Mail. We use the CNN/DM test set, but our own validation set for TL;DR. CNN/Daily Mail SOTA is from Gehrmann et al. (2018). * TL;DR SOTA is from Gehrmann et al. (2019), but the numbers are not comparable as we lack test set access and the TL;DR leaderboard uses an unofficial implementation of ROUGE.

	TL;DR		CNN/Daily Mail	
60k fine-tuned vs. zero-shot	**96%**	4%	**91%**	9%
60k fine-tuned vs. supervised	**97%**	3%	**80%**	20%
60k fine-tuned vs. lead-3	45%	**55%**	40%	**60%**
60k fine-tuned vs. supervised + 60k fine-tuned	**80%**	20%	**74%**	26%
60k fine-tuned vs. 30k fine-tuned	40%	**60%**	**62%**	38%
60k fine-tuned vs. 15k fine-tuned	**79%**	21%	47%	**53%**
60k fine-tuned vs. 60k offline fine-tuned	**64%**	36%	**65%**	35%
60k fine-tuned vs. reference summaries	**96%**	4%	**84%**	16%
lead-3 vs. supervised	**97%**	3%	**89%**	11%
lead-3 vs. reference summaries	**97%**	3%	**89%**	11%
lead-3 vs. supervised + 60k fine-tuned	**75%**	25%	**85%**	15%

Table 5: Human evaluation of summarization models. For each pair of models and each dataset, we sample 1024 articles from the test set, generate a summary from each model, and ask 3 humans to pick the best summary using the same instructions as in training. The model chosen by a majority of the humans wins on that article. We report the fraction of articles that each model wins. For all models, we sample with temperature 0.7 for TL;DR and 0.5 for CNN/DM.

els with 15k, 30k, and 60k human labels, and an offline data collection ablation with 60k labels. We also show zero-shot performance of the pretrained model, a supervised fine-tuned baseline using the same pretrained model as starting point (section 2.2), and a lead-3 baseline which copies the first three sentences of the context. We truncate lead-3 at a period in the same way we truncate generated summaries, so occasionally it is 2 sentences. Finally, we combine supervised and RL fine-tuning: performing human RL fine-tuning starting with the supervised fine-tuned model. The purely RL fine-tuned models use contexts from the datasets during training but ignore the reference summaries; the supervised and supervised+RL models use both contexts and summaries.

We report two sets of numerical results: human evaluations between pairs of models (table 5) and ROUGE results on the test set of CNN/Daily Mail and our validation set of TL;DR (table 4). ROUGE results suggest that online data collection is important for best performance, in contrast to our stylistic continuation tasks. At a fixed number of labels, online tends to be better than offline, with a 3 point R-AVG gain on CNN/DM at 60k labels.[6] On both datasets we see significant returns to data volume up to 60k human labels (though the trend is less clear for human evaluation). On both datasets, supervised + RL fine-tuning is best, and indeed pure RL fine-tuning is worse than the supervised baseline according to

---

[6]That said, different training runs have considerable variation and it is expensive to run multiple seeds with humans, so it is possible that this gap is largely noise.

reference summary	Solar plane attempting to be first to circumnavigate world without using fuel is stuck in China. Solar Impulse 2 attempts to prove the power of renewable energy.
zero-shot	The plane has been grounded in China for two weeks because of bad weather.
60k fine-tune	The Solar Impulse 2, the experimental plane attempting to fly around the world without using a drop of fuel, has been grounded by the weather in China. What was supposed to be an overnight pit stop in the southwestern city of Chongqing has now stretched into a two-and-a-half week stay.
supervised	Solar Impulse 2 has been grounded by the weather in China. The plane took off from Abu Dhabi on March 9. The plane is trying to prove the power of renewable energy.
supervised + 60k fine-tune	Solar Impulse 2 has been grounded in China for two-and-a-half weeks. The plane is attempting to fly around the world without using a drop of fuel. The team, which includes Bertrand Piccard, is taking turns flying the single-seater. The plane took off from Abu Dhabi on March 9 and has successfully flown through Oman.

Table 6: Random ($T = 0.5$) summaries for our CNN/DM summarization task, on the same context. Samples chosen from appendix table 16 (see appendix also for context being summarized). The 60k fine-tune model copies from the source article.

ROUGE in all cases (though the supervised baseline uses the full supervised training dataset, which is much larger than 60k samples). Lead-3 is hard to beat: it is the best model for R-1 and R-2 on CNN/Daily Mail, and only supervised + RL fine-tuning beats it otherwise.

But our goal is optimizing reward defined by humans, not ROUGE. Table 5 shows pairwise comparisons between different model pairs according to human labelers, using 1024 samples with majority vote of 3 labelers per sample. Here the picture is different, though also significantly noisier. Our online trained, 60k label model reliably beats both the zero-shot and supervised baselines, and even beats the combined supervised + RL fine-tuned model. Online training remains important, but the situation w.r.t. data volume is less clear and likely contaminated by noise: the 60k TL;DR model beats the 30k model only 40% of the time, for example. More worrisome, the 60k online model beats the human ground truth 96% of the time for TL;DR and 84% of the time for CNN/Daily Mail.

What is going on? As we show in the next section, our 60k RL fine-tuned model is almost entirely extractive (despite lacking any explicit extractive architectural component): it mostly copies whole sentences from the context, but varies which sentences are copied.

3.2.1. WHAT OUR MODELS COPY

Much previous work in summarization has focused on explicit copying mechanisms, including the pointer network-based architecture of See et al. (2017) and the two-phase mask and paraphrase approach of Gehrmann et al. (2018). The goal is to take advantage of copying (which is of fundamental importance to the task of summarization) without only copying—to be abstractive rather than extractive.

Figures 5 and 6 show the fractions of n-grams and sentences generated by our models which are novel and repeated, respectively. From the novelty stats, we see that our RL fine-tuning consistently causes models to copy more. In particular, our 60k RL fine-tuned models are almost entirely extractive: they copy whole sentences 71% of the time for TL;DR and 98% of the time for CNN/Daily Mail. Applying RL fine-tuning starting from the supervised fine-tuned model copies much less: 6% and 30% for TL;DR and CNN/Daily Mail. Although we do not use explicit coverage metrics as in See et al. (2017); Gehrmann et al. (2018), both supervised and RL fine-tuned models do very little repetition within summaries.

While the purely RL fine-tuned models mostly copy, they vary where they copy from. Figure 7 illustrates this via the position of the longest common subsequence between context and summary. To understand when the model chooses to copy from the exact beginning, we identify common preambles in articles such that we would expect copying to be a poor strategy. Table 7 shows that these preambles are copied much less often than in the immediate beginnings of other articles, giving evidence that our models are smart about when to copy. However, we cannot establish that our reward model is smart beyond rewarding copying, as the zero-shot model also skips preambles.

Since combining supervised fine-tuning and RL fine-tuning gives the best ROUGE scores and and is also more abstractive, why not use it? Unfortunately there is an advantage to pure copying shown in table 8: it makes it easy for the model to tell the truth. The models that copy the most, 60k RL fine-tuned, is 90% and 95% accurate on TL;DR and CNN/Daily Mail; lifting whole sentences from the article usually leaves them true. The supervised fine-tuned and combined supervised+RL fine-tuned models are accurate
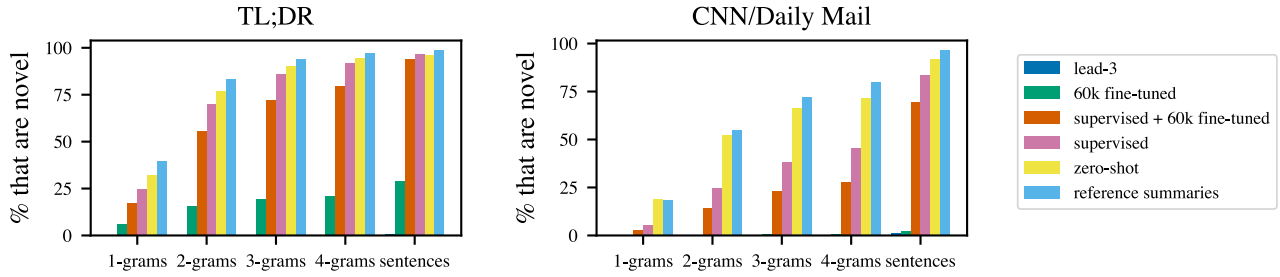
Figure 5: Percent of n-grams and sentences in summaries that do not appear in the source (compare to figure 6 in See et al. (2017)). n-grams are consecutive sequences of words in a single sentence in a summary, and they count as novel if they do not appear consecutively in the article. We ignore punctuation and capitalization.
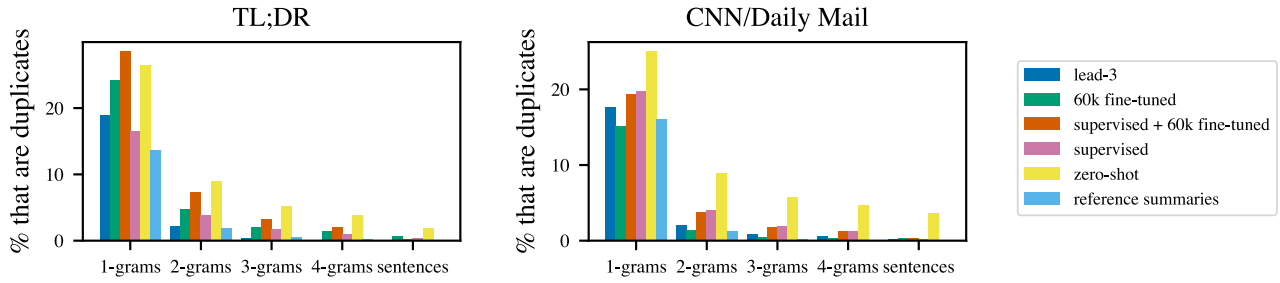
Figure 6: Percent of n-grams and sentences in summaries that appear multiple times in the summary (compare to figure 4 in See et al. (2017)).

Figure 7: Variation in where the models copy from, illustrated by the location of the longest common subsequence of bigrams between context article/post (left) and summary (right) for 256 randomly chosen contexts. Document lengths are shown in gray, with bigrams highlighted (with color depending on positions in contexts). Here, we picture the 60k fine-tuned models, which do the most copying.

	TL;DR		CNN/Daily Mail	
	all	preamble	all	preamble
zero-shot	1.3%	0%	10.4%	1.0%
60k fine-tuned	28.3%	0.2%	77.6%	1.4%
supervised	1.5%	0%	9.4%	0%
supervised + 60k fine-tuned	7.9%	0%	16.6%	0%
reference summaries	0.6%	0%	5.1%	0%
total articles	30000	3762	13368	297

Table 7: How often different models copy the first 3 words of the article as the first 3 words of the summary, on the validation sets. We additionally consider the subset of posts/articles with preambles. On TL;DR, we used posts which begin with one of 'hi', 'hello', 'hey', 'ok', 'okay', or 'so'. For CNN/Daily Mail, we used articles with a colon within the first 3 words, such as "Winner: Simon Wood took home the TV crown [...]" and "Fully charged: The new scheme will let EE customers pick up free portable chargers [...]".

	TL;DR	CNN/Daily Mail
zero-shot	6/30	6/30
60k fine-tuned	26/30	29/30
supervised	8/30	19/30
supervised + 60k fine-tuned	11/30	20/30

Table 8: Frequency with which generated summaries are accurate, in the sense of only making statements supported by the context, as judged by the authors on 30 articles from each dataset. The 60k fine-tuned model achieves high accuracy via copying; the supervised and supervised + 60k fine-tuned models are more abstractive but at significant cost to accuracy.

at most 70% of the time: they paraphrase but paraphrase badly, often swapping names from the context or mixing together multiple sentences in invalid ways. Zero-shot is the most novel, but is accurate only 20% of the time. Similarly, Kryściński et al. (2019) found that 30% of samples from the supervised summarization models they tested contained inconsistencies, and Khandelwal et al. (2019) found that their pretrained encoder-decoder model "hallucinates facts...which are topical but never appear in the source".

There are at least two ways of interpreting these results. The first is that copying is the easiest way to be accurate. The labelers were told to penalize inaccuracy and redundancy, but were not told to penalize copying. The zero-shot model copies some of the time, and when it copied it was accurate, so this behavior was reinforced. The result is a model that "degenerated to copying", but at least does not lie.

However, this does not explain why both our model and lead-3 are strongly preferred by the labelers to the human reference summaries (table 5). This reveals a mismatch between the notion of quality we wanted our model to learn, and what the humans labelers actually evaluated. Checking for copying is very easy, so labelers who check primarily for copying can work quickly. Since the online data collection setting made quality control more difficult, we failed to detect and penalize this behavior.

4. Challenges

We conclude with a few lessons and directions we plan to consider in future reward learning work.

4.1. Online data collection is hard

Online data collection was necessary to achieve the best results on summarization. However, fully online data collection—where each label comes from an up-to-date version of the policy which has already learned from almost all previous labels—had major disadvantages:

1. **Software complexity:** Our online system interleaves data gathering, reward model training, and RL fine-tuning. The resulting distributed system was significantly more complicated than if each task was kept separate, slowing the software development process. Moreover, a bug in any one of the tasks would break the entire training process.

2. **Machine learning complexity:** Online experiments were difficult to debug, as it was hard to iterate on one piece of the ML system at a time. We could often debug an online job only by switching briefly to offline, such as by launching a standalone reward model training run, but then would switch back to online once debugging was complete (until the next cycle).

3. **Quality control issues**: Significant work was required on Scale's part to make their data quality mechanisms work in the low latency, online setting. However, even after this work it was difficult to maintain high data quality over a long period of time, and regressions were often not detected until after (or well after) training runs were complete. Since evaluation of labeler performance was online, by the time a worker was detected as poor some of their data might already been reported back and used for reward model training.

We believe the right middle ground between offline and online data collection is batched data collection, and plan to use this setting in future work. Collect a batch of data from the pretrained policy ρ, train the reward model r on this batch, then fine-tune the policy π with r frozen. Once complete, collect another batch of data sampled from π, and iterate. The latency for each batch can be far longer than the online case, simplifying quality control. As in the fully online setting, we can always retrain the reward model from scratch on all data collected so far; human data is expensive so the total volume will be low. Removing the interleaved training of r and π simplifies software architecture and diagnosis of ML issues, and allows iteration on just one component (say r in isolation) if problems occur. Li et al. (2016) reached similar conclusions in a restricted dialogue setting after validating in simulation that online and batched trained performed similarly.

Batched data collection is also a well-studied setting for active learning techniques. Although we use RL to fine-tune the policy π, the human data is used only for supervised training of the reward model r. Thus, any method for batch mode active learning of supervised models applies, using π as the unlabeled data distribution for r. Examples of such techniques include selecting batches based on entropy considerations (Guo and Schuurmans, 2008), gradient-based metrics (Huang et al., 2016; Ash et al., 2019), or by attempting to distinguish labeled and unlabeled examples (Gissin and Shalev-Shwartz, 2019).

4.2. Sharing parameters between reward model and policy causes overfitting

Although the reward model and policy are both initialized to ρ, we train them as separate networks rather than a single shared network with multiple heads. We might expect joint training to be helpful, effectively using RL as an auxiliary task to improve the reward model's performance. Joint training is particularly appealing because it could help the reward model stay strong enough that the policy cannot exploit it. Sharing could also improve computational efficiency, by allowing the models to share activations rather than requiring two separate forward passes.

Despite several attempts, we were not able to make this idea work. The problem comes from the massive imbalance of data: we have at most 60k samples for the reward model, but 2M episodes for the policy. This makes it challenging to maintain performance on both tasks without performing many epochs for the reward model and overfitting. We hope that future work will overcome this challenge.

4.3. Ambiguous tasks make labeling hard

Evaluation of a summary is both subjective and multidimensional. A single human labeler may have a clear notion of whether a given sample is separately accurate, grammatical, nonredundant, or covers all important topics; but in our experiments a labeler will often be asked to choose between samples each of which has some deficiencies. In choosing which of four samples is the best, a labeler must trade off between different desiderata. This makes consistent labeling difficult for honest labelers (including the authors!), and makes it difficult to quickly detect problematic labelers. It also makes the research more difficult to present and interpret: during our experiments we routinely checked the performance of models by having authors label results, since we knew the authors would attempt to do the task honestly, but were epistemically uneasy about reporting these numbers in the paper (table 8 is the one exception).

One could hope to cope with such "noise" by simply getting more labels and averaging them, but this does not resolve all the practical difficulties with ambiguity. When possible, it seems better to design less ambiguous labeling tasks that get at the same information. For example, rather than asking a person to rate or compare summaries, we could ask for a verbal description of the problems with a summary, or a suggested correction. If problems don't exist we are done; otherwise describing a problem does not require consistently picking the same most important problem. Even if two people disagree on the most important problem, they may be more likely to agree that the other picked *some* problem, and more agreement eases data quality control and the overall experimental process.

4.4. Bugs can optimize for bad behavior

One of our code refactors introduced a bug which flipped the sign of the reward. Flipping the reward would usually produce incoherent text, but the same bug also flipped the sign of the KL penalty. The result was a model which optimized for negative sentiment while still regularizing towards natural language. Since our instructions told humans to give very low ratings to continuations with sexually explicit text, the model quickly learned to output only content of this form, regardless of how innocuous the starting point was. This bug was remarkable since the result was not gibberish but maximally bad output. The authors were asleep during

the training process, so the problem was noticed only once training had finished. A mechanism such as Toyota's Andon cord could have prevented this, by allowing any labeler to stop a problematic training process.

5. Conclusion

We have demonstrated RL fine-tuning of language models to four NLP tasks: stylistic continuation with high sentiment or physically descriptive language, and summarization on the CNN/Daily Mail and TL;DR datasets. Rather than building task-specific techniques, we achieve our results by straight-forwardly applying reward learning to language generation. We extend previous reward learning work with pretrained models and KL regularization to prevent the policy from diverging too far from natural language.

Our results are mixed. On the continuation tasks we achieve good results vs. the zero-shot baseline as evaluated by humans with very few samples: 2.5k for sentiment and 5k for descriptiveness. However, for both summarization tasks our policies are only "smart copiers" (extractive rather than abstractive): they copy from the input text but skip over irrelevant preamble. The advantage of copying is truthfulness: by comparison the zero-shot and supervised models produce natural, plausible-looking summaries that are often lies. We believe the limiting factor in our experiments is data quality, in particular exacerbated by the online data collection setting, and plan to ameliorate this with batched data collection in future.

We believe the application of human reward learning to natural language tasks is important both from a capability and safety perspective. On the capability side, purely supervised training is insufficient to correct mistakes that arise when sampling from trained policies, and RL training to programmatic reward functions such as BLEU or ROUGE is insufficient: Paulus et al. (2017) conclude that "optimizing for single discrete evaluation metric[s] such as ROUGE with RL can be detrimental to the model quality." Interactive tasks such as dialogue are particularly relevant: it is difficult to define the goal of a dialogue without the human participant, and the length of dialogue makes it more likely that supervised learned models will go off distribution. In the supervised case NLP models are trained using human data; if we want RL fine-tuning we need human data too.

On the AI safety side, interactive communication between humans and ML models is a requirement for scalable reward learning methods such as amplification, debate, and recursive reward modeling (Christiano et al., 2018; Irving et al., 2018; Leike et al., 2018), and natural language is how humans communicate complex ideas. Although language models are unlikely to be ready for these tasks in their full generality, Perez et al. (2019) demonstrates that debate al-

ready improves generalization for question-answering when debaters quote from a source text. Using direct human preferences for language tasks is a step in the direction of scalable reward learning for language, and we believe further steps are possible.

Acknowledgments

References

Jordan T Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. *arXiv preprint arXiv:1906.03671*, 2019.

Dzmitry Bahdanau, Felix Hill, Jan Leike, Edward Hughes, Arian Hosseini, Pushmeet Kohli, and Edward Grefenstette. Learning to understand goal specifications by modelling reward. *arXiv preprint arXiv:1806.01946*, 2018.

Forian Böhm, Yang Gao, Christian Meyer, Ori Shapira, Ido Dagan, and Iryna Gurevych. Better rewards yield better summaries: Learning to summarise without references. In *Conference on Empirical Methods in Natural Language Processing*, 2019.

Woon Sang Cho, Pengchuan Zhang, Yizhe Zhang, Xiujun Li, Michel Galley, Chris Brockett, Mengdi Wang, and Jianfeng Gao. Towards coherent and cohesive long-form text generation. In *Proceedings of the First Workshop on Narrative Understanding*, pages 1–11, 2019.

Paul Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, pages 4302–4310, 2017.

Paul Christiano, Buck Shlegeris, and Dario Amodei. Supervising strong learners by amplifying weak experts. *arXiv preprint arXiv:1810.08575*, 2018.

Andrew M Dai and Quoc V Le. Semi-supervised sequence learning. In *Advances in neural information processing systems*, pages 3079–3087, 2015.

Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman,

Szymon Sidor, Yuhuai Wu, and Peter Zhokhov. Openai baselines. https://github.com/openai/baselines, 2017.

Yang Gao, Christian M Meyer, and Iryna Gurevych. Preference-based interactive multi-document summarisation. *arXiv preprint arXiv:1906.02923*, 2019a.

Yang Gao, Christian M Meyer, Mohsen Mesgar, and Iryna Gurevych. Reward learning for efficient reinforcement learning in extractive document summarisation. *arXiv preprint arXiv:1907.12894*, 2019b.

Sebastian Gehrmann, Yuntian Deng, and Alexander M Rush. Bottom-up abstractive summarization. *arXiv preprint arXiv:1808.10792*, 2018.

Sebastian Gehrmann, Zachary Ziegler, and Alexander Rush. Generating abstractive summaries with finetuned language models. In *TL;DR Challenge System Descriptions*, 2019.

Daniel Gissin and Shai Shalev-Shwartz. Discriminative active learning. *arXiv preprint arXiv:1907.06347*, 2019.

Yuhong Guo and Dale Schuurmans. Discriminative batch mode active learning. In *Advances in neural information processing systems*, pages 593–600, 2008.

Braden Hancock, Antoine Bordes, Pierre-Emmanuel Mazare, and Jason Weston. Learning from dialogue after deployment: Feed yourself, chatbot! *arXiv preprint arXiv:1901.05415*, 2019.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *Advances in neural information processing systems*, pages 1693–1701, 2015.

Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

Jiaji Huang, Rewon Child, Vinay Rao, Hairong Liu, Sanjeev Satheesh, and Adam Coates. Active learning for speech recognition: the power of gradients. *arXiv preprint arXiv:1612.03226*, 2016.

Borja Ibarz, Jan Leike, Tobias Pohlen, Geoffrey Irving, Shane Legg, and Dario Amodei. Reward learning from human preferences and demonstrations in Atari. In *Advances in Neural Information Processing Systems*, 2018. URL https://arxiv.org/abs/1811.06521.

Geoffrey Irving, Paul Christiano, and Dario Amodei. AI safety via debate. *arXiv preprint arXiv:1805.00899*, 2018. URL https://arxiv.org/abs/1805.00899.

Natasha Jaques, Shixiang Gu, Dzmitry Bahdanau, José Miguel Hernández-Lobato, Richard E Turner, and Douglas Eck. Sequence tutor: Conservative fine-tuning of sequence generation models with kl-control. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1645–1654. JMLR. org, 2017.

Natasha Jaques, Asma Ghandeharioun, Judy Hanwen Shen, Craig Ferguson, Agata Lapedriza, Noah Jones, Shixiang Gu, and Rosalind Picard. Way off-policy batch deep reinforcement learning of implicit human preferences in dialog. *arXiv preprint arXiv:1907.00456*, 2019.

Urvashi Khandelwal, Kevin Clark, Dan Jurafsky, and Lukasz Kaiser. Sample efficient text summarization using a single pre-trained transformer. *arXiv preprint arXiv:1905.08836*, 2019.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Julia Kreutzer, Joshua Uyheng, and Stefan Riezler. Reliability and learnability of human bandit feedback for sequence-to-sequence reinforcement learning. *arXiv preprint arXiv:1805.10627*, 2018.

Wojciech Kryściński, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. Neural text summarization: A critical evaluation. *arXiv preprint arXiv:1908.08960*, 2019.

Jan Leike, David Krueger, Tom Everitt, Miljan Martic, Vishal Maini, and Shane Legg. Scalable agent alignment via reward modeling: a research direction. *arXiv preprint arXiv:1811.07871*, 2018.

Jiwei Li, Alexander H Miller, Sumit Chopra, Marc'Aurelio Ranzato, and Jason Weston. Dialogue learning with human-in-the-loop. *arXiv preprint arXiv:1611.09823*, 2016.

Jelena Luketina, Nantas Nardelli, Gregory Farquhar, Jakob Foerster, Jacob Andreas, Edward Grefenstette, Shimon Whiteson, and Tim Rocktäschel. A survey of reinforcement learning informed by natural language. In *Proceedings of the Twenty-Eight International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, 8 2019.

Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 43–52. ACM, 2015.

Khanh Nguyen, Hal Daumé III, and Jordan Boyd-Graber. Reinforcement learning for bandit neural machine translation with simulated human feedback. *arXiv preprint arXiv:1707.07402*, 2017.

Romain Paulus, Caiming Xiong, and Richard Socher. A deep reinforced model for abstractive summarization. *arXiv preprint arXiv:1705.04304*, 2017.

Ethan Perez, Siddharth Karamcheti, Rob Fergus, Jason Weston, Douwe Kiela, and Kyunghyun Cho. Finding generalizable evidence by learning to convince Q&A models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, Hong Kong, China, November 2019. Association for Computational Linguistics.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

Alec Radford, Rafal Jozefowicz, and Ilya Sutskever. Learning to generate reviews and discovering sentiment. *arXiv preprint arXiv:1704.01444*, 2017.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training, 2018. URL https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners, 2019. URL https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Abigail See, Peter J Liu, and Christopher D Manning. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*, 2017.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015.

Pradyumna Tambwekar, Murtaza Dhuliawala, Animesh Mehta, Lara J Martin, Brent Harrison, and Mark O Riedl. Controllable neural story generation via reinforcement learning. *arXiv preprint arXiv:1809.10736*, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

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

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

Yuxiang Wu and Baotian Hu. Learning to extract coherent summary via deep reinforcement learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Sanghyun Yi, Rahul Goel, Chandra Khatri, Tagyoung Chung, Behnam Hedayatnia, Anu Venkatesh, Raefer Gabriel, and Dilek Hakkani-Tur. Towards coherent and engaging spoken dialog response generation using automatic conversation evaluators. *arXiv preprint arXiv:1904.13015*, 2019.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27, 2015.

A. Instructions for labelers

A.1. Sentiment

You will see snippets of text followed by several possible ways the text could be continued. Choose the most positive and happy continuation:

- Consider all of the moods of the characters, the mood of the narrator, and the mood implied by the events and descriptions.

- If a continuation contains sexually explicit or otherwise offensive content, treat it as strongly negative. Do not choose it even if it expresses a positive attitude.

A.2. Descriptiveness

You will see snippets of text followed by several possible ways the text could be continued. Choose the most vividly descriptive continuation:

- Evaluate both on the **quantity** and on the **vividness** of physical details described.

- The best continuations are full of details that give a strong sense of what the scene looks, sounds, or smells like.

- Count only **physical details**, not details about abstract facts.

A.3. Summarization: TL;DR

You will see some text followed by several summaries. Please read the text and select the best summary.
A summary is good if it:

- Is useful and a good summary in general

- Accurately states the important points of the text

- Makes sense on its own

A summary is bad if it:

- Includes information that doesn't appear in the text

A.4. Summarization: CNN/DM

You will see an article followed by several summaries. Please read the article and select the best summary.
A summary is good if it:

- Is useful and a good summary in general

- Accurately states the important points of the article

- Makes sense on its own

A summary is bad if it:

- Includes information that doesn't appear in the article

- Includes quotations that don't appear verbatim in the article

B. Human labeling details

Our quality assurance process was handled by Scale AI, though Scale made significant changes to their usual quality systems in order to deal with subjective tasks and provide very fast turnaround. Since we initially believed online data collection would be crucial, even the offline experiments were collected with this fast turnaround requirement. In the future we plan to use a more relaxed latency requirement.

The first step of data collection involves teaching the task to a small number of trusted Scale labelers by giving them a description of the task (appendix A). Scale uses these labelers to collect a large number of *benchmark* data points where several trusted labelers agree (out of a large set of unlabeled data points from ρ). During full data collection, Scale serves these benchmark data points to freelance workers alongside real unlabeled data for training (the two types of data are indistinguishable when $\pi = \rho$, though they do become distinguishable during training), maintaining a confidence model for the performance of each labeler on the benchmark distribution. The probability of getting a benchmark vs. a real sample varies dynamically on factors such as confidence in the labeler to correctly label a certain category. Freelancers who fail to perform well on benchmark tasks are filtered out. Additionally, Scale makes ad-hoc improvements to quality control over time, sometimes validating quality by comparing to a small number of gold-standard labels from the authors.

We evaluated the data quality after the fact on two of the tasks. During all data collection, 5% of queries were answered by 5 distinct labelers. We sampled 100 of these queries (restricting to ones generated from ρ) and had two authors label each one. Based on this data, we estimated the rate of agreement between authors and Scale labelers, pairs

$P(\text{agreement})$	Sentiment	TL;DR
Between random responses	*25%*	*25%*
Between labelers	38±2%	46±2%
Between an author & a labeler	44±5%	38±5%
Between authors	62±5%	61±5%

Table 9: Agreement probabilities for two tasks, i.e. the probability of both individuals choosing the same sample as best, out of 4.

of labelers, and pairs of authors. As table 9 shows, the data contained a significant amount of signal but did not match the quality of data which was hand-labeled by the authors.

An earlier version asked labelers for 1-10 ratings; in the best case this provides more information per label, but it was difficult to gauge labeler performance. Normalization was required since two good labelers would often differ by a (noisy) monotonic transform. If many scores concentrated on a few values (say 7 and 8) simple strategies could fool the filtering process. Absolute scores also tended to drift over the training process, as labelers would adjust to the new distribution of samples from the changing policy.

Finding high-quality workers involves human answering quality control questions which are not used in our experiments, and throwing away data from low-quality workers. So the total human cost of experiments is somewhat higher than the number of labels we actually use (which is what we report). For a short training run this can easily dominate the actual label requirements, though it can be amortized across several tasks by identifying consistently good workers. For our longer training runs the additional number of labels was modest. (None of these details are exposed to customers.)

C. Samples

Samples from our models are shown in the following tables:

- Mock sentiment continuation without a KL penalty: table 10

- Sentiment continuation: table 11

- Descriptiveness continuation: table 12

- TL;DR summarization: tables 13 to 15

- CNN/Daily Mail summarization: tables 16 to 18

Samples without a KL penalty

context	And she would always love him for that. And for many other reasons. He had given her so much and she	
	Continuation 1	*Continuation 2*
zero-shot	had no idea how to release that gift, but she knew	had been unable to give it back. "You
no penalty / bonus	! These These These sound flowed instantly easily easily easily easily!	! These These These sound flowed instantly easily easily easily easily!
entropy bonus	initially initiallyprisingly easilyprisingly Liam! Love this ride instantly tightly	' equally particularlyprisinglyprisingly Liam Liam! Love this ride instantly
context	Palmer rolled her eyes. "Don't give me that. You didn't come here, wearing that	
	Continuation 1	*Continuation 2*
zero-shot	corset and corset and a blasted corset. If	old-fashioned, sexy little dress. You came here to
no penalty / bonus	! These These These sound flowed instantly easily easily easily easily!	! These These These sound flowed instantly easily easily easily easily!
entropy bonus	initially relativelyprisingly instantlyprisingly Liam Liam! Love this ride instantly	overwhelmingly Loganprisingly Liam instantly surprisingly! Love this ride instantly tightly
context	Is it conceivable that - I don't know - he somehow wasn't on the aircraft that exploded? Like maybe he	
	Continuation 1	*Continuation 2*
zero-shot	simply didn't hear me coming or he was just coming out	was just on the ground." "Yes, at
no penalty / bonus	! These These These sound flowed instantly easily easily easily easily!	! These These These sound flowed instantly easily easily easily easily!
entropy bonus	prisinglyprisingly instantly Liam instantly overwhelmingly! Love this ride instantly tightly	instantly initially overwhelmingly Chloe! Love this ride instantly instantly instantly tightly

Table 10: Samples from a model fine-tuned to mock sentiment without a KL penalty. Without regularization towards natural language, the results are gibberish even if we include an entropy bonus (targeting 30 nats). Both policies reach a mean score of around +8.0 according to the mock sentiment model (i.e. 99.97% positive).

Samples for positive-sentiment continuation task

context	"He also doesn't have a backbone," Quinn continued as they walked. "Last year, I had a great story about Paul Gibson, who is now the chairman of the board of supervisors.		
	Continuation 1	*Continuation 2*	*Continuation 3*
zero-shot	He was running for the supreme court and his campaign was violent. There was an incident at a press conference.	For weeks, I went around the county, telling everyone how amazing he was and how much I admired him.	He was one of the first teachers to lead by example. It was incredibly inspiring to see.
mock	I was really impressed with how he handled the board, and I thought it was a great story.	He was a real hero to the people in town. He was a real hero.	But he just sat around and watched. I love that story. You should read it again.
5k offline fine-tune	Paul Gibson is a very intelligent man. He's a good heart man. He has a good sense of humor.	Paul is a very nice guy, a true gentleman, and we all know he's a really cute guy.	He was a good man, with a great smile, a great sense of humor, a great family.
context	Pearl thought to herself that what they were about to do was exactly the sort of thing that they could do to help the villagers. They were all terrified of these guys. At the police station the three walked up to the counter behind which was a senior constable studying some papers.		
	Continuation 1	*Continuation 2*	*Continuation 3*
zero-shot	"Hello, I'm Pearl and this is my friend, Mike," said Pearl.	"May we speak to the police officer, sir?" asked the one in charge.	'Hello, can I help you?' 'Yes, we're the same people that the people were talking about.
mock	"I'm Detective Sergeant Lewis, this is Detective Sergeant Jones and this is Detective Sergeant Porter.	He looked up at the three with a friendly smile. "Good day, gentlemen.	"We have a friend who is in trouble. He was kidnapped by these guys," the boy said.
5k offline fine-tune	He turned to them with a smile. "Good afternoon, ladies. I'm Detective Inspector Jones.	The constable stood up and smiled as he saw them, obviously pleased to see them.	He smiled at them and waved them in, his eyes twinkling as he listened to their tales.
context	The room was warm and welcoming, with two hearths that burned vigorously. The Vikings pushed her to the floor at Gunnar's feet. Ailia looked up at Gunnar and rose to her feet as calmly as possible.		
	Continuation 1	*Continuation 2*	*Continuation 3*
zero-shot	"I will not fight." "You will fight for us," Gunnar said.	"I'll give you one last chance to tell me where that is," Bjorn said.	"I'm sorry. I didn't mean to…" "Give me your hand." Gunnar told her.
mock	"I love this. I love this whole room. I'm going to enjoy it.	"I am very pleased to meet you, and let me assure you, you will not regret this.	"Bravely done, Gunnar. Thank you." Ailia smiled at him.
5k offline fine-tune	Gunnar smiled at Ailia and waited for her to smile back. She was smiling, she knew.	"You've been a true friend to my son, Gunnar. Thank you." She smiled.	Gunnar was smiling at her and his pretty, blue eyes were filled with contentment.

Table 11: Continuations for our positive-sentiment continuation task, with contexts randomly sampled from BookCorpus test set. Random samples from pretrained, mock sentiment, and 5k offline descriptiveness models, using temperature $T = 0.7$.

Samples for descriptiveness continuation task

context	"I do not know if it was Viking related, but it could have been." "Really?" Ailia said. Is it safe to be traveling here then? Ailia looked behind her to make sure they weren't being followed.		
	Continuation 1	*Continuation 2*	*Continuation 3*
zero-shot	There were no signs of anyone. "It is safe enough," Ailios said.	"Because I have a friend that is in the area and he will be coming with us.	It was hard to see that far. "I do not like that word.
5k offline fine-tune	Kaya crouched low, her eyes wide in the moonlight. Her body was tense.	She put her hand on the sword strapped to her back, and then pulled it out.	She strode out the door and walked down the street, her nose wrinkled in disapproval.
context	"They took me to a cellar. An' it were full of girls, it were, all sewin'. At first I thought it weren't too bad. I don't mind work, really I don't. Mama said I was a good 'elper.		
	Continuation 1	*Continuation 2*	*Continuation 3*
zero-shot	But the girls keep on sneakin' in 'em. An' one day, one of em found me.	But in the end, it weren't anything like she'd promised me. Some of them were worse'n me.	I was just lucky to get this job. But as I stayed there, I started to see things.
5k offline fine-tune	But..." The girl hiccupped, then pressed her lips together and shook her head.	But then..." His voice trailed off and he shook his head, his eyes going distant, somber.	She said I could keep my eyes open and my head clear and make a good target.
context	She forced herself up and shifted closer to him until she was kneeling behind him. Following her instincts, she laid a splayed hand on his bare back. He tensed beneath her touch. Quietly, Tara found her words. "I will never lie to you again, Max.		
	Continuation 1	*Continuation 2*	*Continuation 3*
zero-shot	I love you." He turned to her, his face a mask of pain.	You have to know that." "I know you will never lie to me again.	Whatever happens, I'll tell you the truth." "I'm sorry.
5k offline fine-tune	I promise." His hands curled around her forearms, his fingers digging deep into her flesh.	I swear it." He exhaled heavily and hugged her close, pressing his lips to her forehead.	I love you." Max leaned his head back and closed his eyes. His breathing changed.

Table 12: Continuations for our descriptiveness continuation task, with contexts randomly sampled from BookCorpus test set. Random samples from pretrained and 5k offline descriptiveness models, using temperature $T = 0.7$.

Samples for TL;DR summarization task (part 1)

context post	Okay, this was actually going to go into a comment on youtube, but Emma said she wanted more traffic on reddit. I am a musician myself, have been for many years now. While Emma's music is not the most complicated technical death metal stuff out there, I love it dearly. Her music actually moves me. No matter what she plays, be it a cover or original stuff, her music *touches* me on a deep personal level. And I have absolutely no clue why. I think it's a combination of her voice, her lyrics and the overall sound design of her songs and covers. But this isn't always a good thing, I feel. I can't listen to her music for hours on end. Only always a few songs, nothing more. Her music destroys me emotionally. Okay, that's a bit harsh, but she can get me, a (nearly) adult on the verge of tears. I've gone through some shit in my life, who hasn't, and I feel like her music can actually get some of it out. I am incredibly thankful that she makes her music the way she makes it. I can honestly say that Emma Blackery is my favourite artist. Keep in mind that I listen to a lot of music, from crossover thrash metal over 80's pop, berlin polygenre to classic/romantic music. I am also very jealous of her ability. I play quite a few instruments myself (guitar, piano, bass, drums, synth, saxophone, percussion) and aim to study audio engineering. I can definitely say that music is my literal life and mean it. I've released my first full length album at age 20. Yet I am very jealous that she is as good as she is and I really wanna know what her secret is. I love and hate her music, does that make sense? Her music definitely makes me think, and that's what art is all about, right? Anyway, I'm gonna continue working on my cover of The Promise and try to figure out what it is she's doing. Thank you for reading this far.
reference summary	Emma music gud yes very love magic
zero-shot	I'm a musician myself, have been for many years now. And Emma Blackery is my favourite artist.
60k fine-tune	I am a musician myself, have been for many years now. While Emma's music is not the most technical death metal stuff out there, I love it dearly. Her music actually moves me. No matter what she plays, be it a cover or original stuff, her music *touches* me on a deep personal level.
supervised	I'm jealous of Emma Blackery's music and I have no clue why.
supervised + 60k fine-tune	Emma Blackery is my favourite artist, and I love her music to death. I wish I could listen to her music all the time. However, I feel like her music destroys me emotionally. I feel like I need to enjoy her music for the human part of it, and not the technical part. She's my favourite artist.

Table 13: Random ($T = 0.7$) summaries for our TL;DR summarization task, with contexts randomly sampled from TL;DR validation set.

Samples for TL;DR summarization task (part 2)

context post	It really does. I did the first 5 seasons in a couple months, and ultimately I enjoyed them. The first three seasons cover just about all the really good stuff. Russell Edgington was, to me, the best supporting character in the show. He was everything that I valued about the show, which is to say, campy, violent, overly-gory fun. I went back to the show a little over a year later, and while S6 was tolerable with a few fun bits, S7 totally eroded. Tara's plot goes full retard in the last season, people die for the sake of drama and convenience or leave with no real reverence for their history. Suddenly no one can make a good decision, and there was a needless romance bait-and-switch that ruined what I felt the show was moving towards. What's terrible is that once you've done six seasons it's hard to walk away, but each episode of seven stings more than the last. You don't get that weird half-hollow/half-warm feeling you should when you finish a series of books or a long TV show. Outside of Pam and Erik, I didn't give two fucks about any character by the end.
reference summary	Seasons one through three were a blast. Four was pretty meh, five and six weren't too bad, seven sucked massively.
zero-shot	I finished S6 and still feel like I'm in the same place as I was with S5. Favorite Episodes: S1: "Tough Love" S2: "Crossover" S3: "The Party"
60k fine-tune	I did the first 5 seasons in a couple months, and ultimately I enjoyed them. The first three seasons cover just about all the really good stuff. Russell Edgington was, to me, the best supporting character in the show. He was everything that I valued about the show, which is to say, campy, violent, overly-gory fun
supervised	10/10 would watch again
supervised + 60k fine-tune	The first two seasons of the show were enjoyable. The show's progress in the last three seasons is frustrating because it's so similar to the first three seasons, and it's so one-sided that there's no reason to keep going. But the show's greatest strength has always been its great writing, and it's giving me that same feeling again.

Table 14: Random ($T = 0.7$) summaries for our TL;DR summarization task, with contexts randomly sampled from TL;DR validation set.

Samples for TL;DR summarization task (part 3)

context post	> you should be happy, lots of votes were counted twice in precincts that went to Clinton in a landslide, if they recount them by hand she will lose more votes, and probably a lot... You are assuming you understand what the problem here is and what the end result on the election was. The article does not state where the problem lies. E.g. The machine shows a tally of 120 votes having been processed on it. The actual reported vote tally to the election board was 100 votes. The 20 vote differential on the machine is due to 20 times a ballot needing to be re-scanned due to error. No double-counting of votes occurs but there is also a delta between what the machine says was recorded vs what was reported. In this case the final reported vote tallies may have been factually correct but there is no way to validate them against the machine record. To understand the problem an analysis and deconstruction of the machines and the election night events must be undertaken. Find out the cause of the problem, if the problem caused votes to be over/under counted, or if the problem was solely with machine data that was not used for tabulation of results.
reference summary	neither you nor I have any idea what the actual problem is so there is no basis for saying the problem favors either candidate at this time, or if either candidate would even be affected.
zero-shot	Election officials did not count ballots in a timely manner, by hand, to meet the threshold of getting votes counted on election night. Voters were not eligible to vote on election night and were not aware of the problems. The election officials did not realize that their system was broken until very late in the night.
60k fine-tune	The machine shows a tally of 120 votes having been processed on it. The actual reported vote tally to the election board was 100 votes. The 20 vote differential on the machine is due to 20 times a ballot needing to be re-scanned due to error. No double-counting of votes occurs but there is also a delta between what the machine says
supervised	Stop assuming the election was completely free of double-counting before you even begin to discuss the actual results. For any result to be close, it must be based on the total vote totals reported to the election board and not just the machine results. The election board did not report the machine results to the election board.
supervised + 60k fine-tune	The reported vote tallies may have been factually correct but there is no way to validate them against the machine record. The machine may have actually been accurate and the machine may have been faulty in other ways. The election may have been considered fair but the machine may not have been. The results are still in dispute.

Table 15: Random ($T = 0.7$) summaries for our TL;DR summarization task, with contexts randomly sampled from TL;DR validation set.

Samples for CNN/DM summarization task (part 1)

context article	Google launched a new U.S. wireless service today that switches between Wi-Fi and cellular networks Google is already the world's most popular phone software provider, and a pay-TV operator - and now it wants to be your mobile network carrier. The company has unveiled a U.S. wireless service that switches between Wi-Fi and cellular networks to curb data use and keep phone bills low. The service, called 'Project Fi,' debuted today, about two months after Google revealed its plans to expand its ever-growing empire into providing wireless connections for smartphones. Google is selling the basic phone service for $20 a month and will only charge customers for the amount of cellular data that they use each month, instead of a flat rate. Each gigabyte of data will cost $10 a month. That means a customer could sign up for a plan offering three gigabytes of data and get $20 back if only one gigabyte was used in a month. [...*Truncated*...] Rather than building its own network, Google is leasing space on cellular towers built by Sprint and T-Mobile, which are hoping the deals will boost their profits without costing them too many customers tempted to defect to Project Fi.
reference summary	Project Fi will be hosted through Sprint Corp and T-Mobile's networks. It costs $20 for basic service and unused data is paid back to customer. The invitation-only service will work only on Nexus 6 phones in the US. Numbers will live in the cloud so users can talk on any connected tablet.
zero-shot	Project Fi is a new smartphone service that switches between Wi-Fi and cellular networks to curb data use and keep phone bills low. The service is available for an introductory price of $20 a month and will only charge customers for the amount of cellular data that they use each month.
60k fine-tune	Google is already the world's most popular phone software provider, and a pay-TV operator - and now it wants to be your mobile network carrier. The company has unveiled a U.S. wireless service that switches between Wi-Fi and cellular networks to curb data use and keep phone bills low.
supervised	Google launched a new U.S. wireless service today. Called 'Project Fi,' it switches between Wi-Fi and cellular networks. The service, called 'Project Fi,' debuted today. About two months after Google announced its plans to expand its ever-growing empire into providing wireless connections for smartphones.
supervised + 60k fine-tune	Google launched a new U.S. wireless service today that switches between Wi-Fi and cellular networks. The service, called 'Project Fi,' debuted today, about two months after Google revealed its plans to expand its ever-growing empire into providing wireless connections for smartphones.

Table 16: Random ($T = 0.5$) summaries for our CNN/DM summarization task, with contexts (shown truncated) randomly sampled from CNN/DM test set.

Samples for CNN/DM summarization task (part 2)

context article	With the date of their hugely anticipated bout less than a month away, Floyd Mayweather and Manny Pacquiao's training intensity has hit gruelling levels ahead of the clash in Las Vegas.
	Mayweather is as renowned for showing off his extravagant lifestyle as he is for going undefeated in his professional career.
	But the latest pictures released from his training camp, appearing to show him hanging a weight from his head, displays the extreme level he hits in order to be as fit as possible for his fights.
	Manny Pacquiao (right) poses alongside actor Mark Wahlberg, who visited the Filipino at training
	Floyd Mayweather released two pictures from his training camp appearing hold a weight from his head
	Mayweather (left) takes on Pacquiao in a £160million mega-fight in Las Vegas on May 2
	Mayweather (left) appeared to take his training intensity to a new height before his meeting with Pacquiao
	As for Pacquiao, he had a famous visitor check up on his preparations for May 2.
	[...*Truncated*...]
	Pacquiao and Mayweather's clash will be a reported £160million mega-fight, with Pacquiao seeking to inflict a first ever defeat upon Mayweather.
	The 36-year-old Pacquiao also uploaded a picture to his Twitter page away from training with his daughter.
reference summary	Manny Pacquiao took time out from training to meet Mark Wahlberg. Floyd Mayweather shared a picture of him holding a weight with his head. The due meet in Las Vegas in a £160million encounter on May 2.
zero-shot	Mayweather and Pacquiao are going to clash in Las Vegas on May 2.
60k fine-tune	With the date of their hugely anticipated bout less than a month away, Floyd Mayweather and Manny Pacquiao's training intensity has hit gruelling levels ahead of the clash in Las Vegas.
	Mayweather is as renowned for showing off his extravagant lifestyle as he is for going undefeated in his professional career.
supervised	Floyd Mayweather and Manny Pacquiao will fight in Las Vegas on May 2. Pacquiao has a visitor check up on his preparations for the bout. Mayweather released pictures of his training camp to social media.
supervised + 60k fine-tune	Floyd Mayweather and Manny Pacquiao will fight in Las Vegas on May 2. Mayweather released two pictures from his training camp appearing hold a weight from his head. The 36-year-old boxer is as renowned for showing off his extravagant lifestyle as he is for going undefeated in his professional career.

Table 17: Random ($T = 0.5$) summaries for our CNN/DM summarization task, with contexts (shown truncated) randomly sampled from CNN/DM test set.

Samples for CNN/DM summarization task (part 3)

context article	The Solar Impulse 2, the experimental plane attempting to fly around the world without using a drop of fuel, has been grounded by the weather in China.
	What was supposed to be an overnight pit stop in the southwestern city of Chongqing has now stretched into a two-and-a-half week stay.
	It's an unexpected layover that is testing the patience of the pilots and more than 60 team members, just a quarter of the way along a journey that will cover some 35,000 kilometers (21,748 miles) over five months.
	"You can believe that your will can influence the weather," says pilot Bertrand Piccard, who comes from a long family line of explorers. But "then you'll be really disappointed because it doesn't work.
	"Or you just know that it's an adventure you do with the forces of nature."
	Piccard, along with Andre Borschberg, a Swiss engineer, are taking turns flying the single-seater aircraft, which took off from Abu Dhabi on March 9 and has successfully flown through Oman, India, and Myanmar.
	There is much at stake.
	[...*Truncated*...]
	"Our boss is the sun," says Solar Impulse spokeswoman Claudia Durgnat.
	Durgnat says there may be a tiny window ahead, with the next possible departure from Chongqing on Tuesday. "It's not good before, and the days after don't look very clear."
reference summary	Solar plane attempting to be first to circumnavigate world without using fuel is stuck in China. Solar Impulse 2 attempts to prove the power of renewable energy.
zero-shot	The plane has been grounded in China for two weeks because of bad weather.
60k fine-tune	The Solar Impulse 2, the experimental plane attempting to fly around the world without using a drop of fuel, has been grounded by the weather in China. What was supposed to be an overnight pit stop in the southwestern city of Chongqing has now stretched into a two-and-a-half week stay.
supervised	Solar Impulse 2 has been grounded by the weather in China. The plane took off from Abu Dhabi on March 9. The plane is trying to prove the power of renewable energy.
supervised + 60k fine-tune	Solar Impulse 2 has been grounded in China for two-and-a-half weeks. The plane is attempting to fly around the world without using a drop of fuel. The team, which includes Bertrand Piccard, is taking turns flying the single-seater. The plane took off from Abu Dhabi on March 9 and has successfully flown through Oman.

Table 18: Random ($T = 0.5$) summaries for our CNN/DM summarization task, with contexts (shown truncated) randomly sampled from CNN/DM test set.