


MIXCE: Training Autoregressive Language Models by Mixing Forward and Reverse Cross-Entropies

Shiyue Zhang^{♣*} Shijie Wu[♡] Ozan Irsoy[♡]
Steven Lu[♡] Mohit Bansal[♣] Mark Dredze^{♡♣} David Rosenberg[♡]
[♡]Bloomberg [♣]UNC Chapel Hill [♣]Johns Hopkins University

Abstract

Autoregressive language models are trained by minimizing the cross-entropy of the model distribution Q_θ relative to the data distribution P – that is, minimizing the *forward cross-entropy*, which is equivalent to maximum likelihood estimation (MLE). We have observed that models trained in this way may “over-generalize”, in the sense that they produce non-human-like text. Moreover, we believe that *reverse cross-entropy*, i.e., the cross-entropy of P relative to Q_θ , is a better reflection of how a human would evaluate text generated by a model. Hence, we propose learning with MIXCE, an objective that mixes the forward and reverse cross-entropies. We evaluate models trained with this objective on synthetic data settings (where P is known) and real data, and show that the resulting models yield better generated text *without* complex decoding strategies.

 <https://github.com/bloomberg/mixce-acl2023>

1 Introduction

Rapid advances in pre-trained large-scale autoregressive language models (LMs) have dramatically improved the performance of a variety of tasks (Radford et al., 2019; Brown et al., 2020; Zhang et al., 2022; Chowdhery et al., 2022). However, these systems still struggle in many open-ended generation settings, where they are asked to produce a long text following a short prompt. In these cases, we seek systems that generate sensible, coherent, fluent, and engaging, or in short, *human-like* text (Pillutla et al., 2022).

Different decoding strategies to generate such text from pretrained LMs suffer from different degeneration problems. Unbiased sampling¹ usually

* Work done during an internship at Bloomberg.

¹Unbiased sampling is vanilla random sampling, i.e., sampling with temperature=1.0. It is also called ancestral sampling (Eikema and Aziz, 2020) or pure sampling (Holtzman

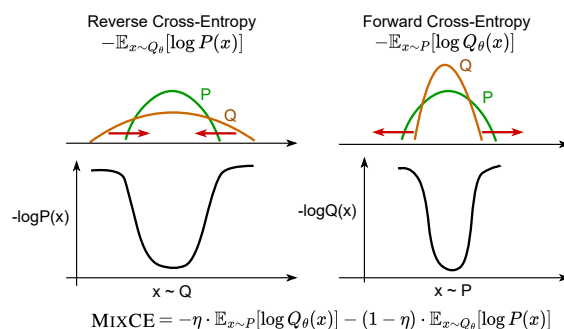


Figure 1: MIXCE combines two complementary driving forces: reverse CE helps narrow the model distribution Q_θ down when it is broader than data distribution P , while forward CE helps broaden Q_θ out when it is narrower than P .²

results in incoherent and nonsensical text, while greedy and beam searches often get stuck in repetition loops (Holtzman et al., 2020). These observations suggest that the learned LM distribution Q_θ still differs substantially from the human LM distribution P . A possible reason is that the autoregressive modeling of Q_θ gives a non-zero probability to every possible sequence of tokens, while many sequences are impossible under P . Nevertheless, we still hope that $Q_\theta(x)$ is as small as possible when $P(x) = 0$. To this end, maximum likelihood estimation (MLE), i.e., minimizing the cross-entropy (CE) $-\mathbb{E}_{x \sim P}[\log Q_\theta(x)]$, is the most widely used objective to train $Q_\theta(x)$ using sequences sampled from P . In an idealized setting, with unlimited training data and model capacity, as well as a perfect optimizer, fitting Q_θ with MLE will learn a distribution as close to P as we like. However, in practice, we only have finite and noisy data.

We argue that the MLE objective only weakly penalizes generations x from Q_θ that are “bad”, et al., 2020). We call it unbiased sampling because it allows unbiased exploration of the model distribution.

²Note that $\log P(x)$ is infinite when $P(x) = 0$. But in practice, we use $\log P(x) = \sum_t \log(P(x_t|x_{<t}) + \epsilon)$ to avoid $\log 0$ and $\epsilon = 1e - 30$.

in the sense that $P(x) = 0$. When Q_θ puts a small amount of probability mass onto $P(x) = 0$ space, MLE cannot sufficiently discourage this behavior (see Figure 3 in Appendix C). Moreover, minimizing forward CE, $-\mathbb{E}_{x \sim P}[\log Q_\theta(x)]$, is equivalent to minimizing the forward KL divergence between P and Q_θ , i.e., $\text{KL}(P||Q_\theta) = \mathbb{E}_{x \sim P}[\log P(x)/Q_\theta(x)]$. Forward KL has a *zero-avoiding* property – avoiding $Q_\theta(x) = 0$ when $P(x) \neq 0$ (Murphy, 2012). Therefore, if there is noise in the data, Q_θ will try to cover the noise as well, which leads the model to *over generalize*, in the sense of putting non-trivial probability mass over $P(x) = 0$ generations (Huszár, 2015; Theis et al., 2016; Ott et al., 2018; Kang and Hashimoto, 2020). As a result, we observe samples from the model deviating from human-like text. A common strategy is to modify the decoding method, e.g., top- k , top- p , typical, contrastive (Fan et al., 2018; Holtzman et al., 2020; Meister et al., 2022; Li et al., 2022) samplings, to tailor the model distribution Q_θ in a post-hoc manner to avoid unwanted generations. In contrast, our approach differs: how can we obtain a better Q_θ to obviate the need for these sampling strategies?

We propose a novel training objective for autoregressive LMs – MIXCE that **Mixes** the forward and reverse **Cross-Entropies**: $-\eta \cdot \mathbb{E}_{x \sim P}[\log Q_\theta(x)] - (1 - \eta) \cdot \mathbb{E}_{x \sim Q_\theta}[\log P(x)]$. MIXCE can be understood in two ways. First, we want model generations to be high-quality as well as diverse. Reverse cross-entropy reflects how we conduct human evaluations, sampling from the model Q_θ and evaluating it by the human P , where the focus is text *quality*. Forward cross-entropy emphasizes the *diversity* of model generations (Hashimoto et al., 2019). Second, MIXCE works similarly to a mixture of the forward and reverse KL divergences. The reverse KL divergence ($\text{KL}(Q_\theta||P)$) is *zero-forcing* – forcing $Q_\theta(x) = 0$ when $P(x) = 0$ – and thus more strongly penalizes generating non-human-like samples compared to MLE. Overall, MIXCE combines two complementary driving forces to better fit Q_θ to P (Figure 1). We elaborate on these interpretations in § 3.1.

Unfortunately, optimizing reverse cross-entropy is intractable because we do not know P . Hence, we propose an approximation of the reverse cross-entropy (see § 3.2), which ends up being a *self-reinforced* loss function that encourages the model to produce generations in which it is already con-

fidant. This loss function has the same computational complexity as forward cross-entropy, making MIXCE easy to implement and as fast as MLE.

We demonstrate the effectiveness of MIXCE in both a synthetic setting, where the “human” distribution P is known, as well as a real setting. For the synthetic case, we evaluate six learning objectives: MIXCE, MIXCE* (MIXCE without approximation), forward KL (=MLE), reverse KL, the mixture of two KL divergences, and Jensen–Shannon (JS) divergence. We show that MIXCE* works slightly worse than the mixture of KLs while outperforming other objectives, and MIXCE works worse than MIXCE* but generally outperforms MLE. In real settings, we finetune GPT-2 (Radford et al., 2019) of different sizes on three English text domains using MIXCE or MLE. Our results show that, compared to MLE, unbiased sampling from MIXCE-finetuned models produces text that has diversity (Meister et al., 2022) closer to that of human text, has higher Coherence (Su et al., 2022), has higher Mauve (Pillutla et al., 2021), and is preferred by humans. When using top- p sampling (Holtzman et al., 2020) and carefully tuning p , generations from MLE-finetuned models are similar to those generated from MIXCE-finetuned models. Nonetheless, MIXCE models have tuned p values closer to 1, implying a less noisy model distribution. In addition, we modify the original Mauve to make it more robust to spurious features (e.g., text length), under which MIXCE still improves over MLE when using unbiased sampling.

2 Background and Related Work

2.1 Autoregressive Language Modeling

Language generation is mostly based on the autoregressive language modeling methodology. The generation of one word is conditioned on previously generated words, $Q_\theta(x_t|x_{<t})$, and the final probability of the sequence x is the product of probabilities of each step, $Q_\theta(x) = \prod_t Q_\theta(x_t|x_{<t})$. Early works build n-gram neural LMs (Bengio et al., 2000) and then RNN-based LMs (Mikolov et al., 2010), and now Transformers (Vaswani et al., 2017) have become the dominant architecture. Language generation models have either a decoder-only (Mikolov et al., 2010) or an encoder-decoder architecture (Sutskever et al., 2014; Bahdanau et al., 2015). In this work, we focus on decoder-only LMs. In recent years, many large-scale pre-trained decoder-only LMs have been introduced (Radford

et al., 2019; Brown et al., 2020; Zhang et al., 2022; Chowdhery et al., 2022). They can be finetuned for downstream tasks and even perform surprisingly well in a zero-shot or few-shot manner. Despite the impressive performance, language *degeneration* is one of the key issues that remain to be solved.

2.2 Language Degeneration

According to Holtzman et al. (2020), language degeneration refers to output text that is *bland, incoherent, or gets stuck in repetitive loops*. It is widely observed in open-ended generations from pretrained LMs. Two commonly observed patterns of degeneration are the incoherent text from unbiased sampling and the repetitive text from greedy or beam search. Degeneration also appears in sequence-to-sequence generation tasks but in a slightly different form (Stahlberg and Byrne, 2019).

There is no agreement on what causes degeneration. Ott et al. (2018) attribute it to data noise and the smooth class of model functions. It is inherent in the model’s structure to have support everywhere, in particular, because all probabilities are produced by softmax, which is strictly positive. Therefore, Hewitt et al. (2022) assume that an LM distribution is the true data distribution plus a uniform-like smoothing distribution. Based on the observation that human-like text has a large but not too large likelihood under the learned LM distribution (Zhang et al., 2021), a lot of works propose empirically useful decoding methods beyond unbiased sampling and greedy/beam search (Fan et al., 2018; Holtzman et al., 2020; Eikema and Aziz, 2020; Basu et al., 2021; Meister et al., 2022; Li et al., 2022; Hewitt et al., 2022; Su et al., 2022; Krishna et al., 2022). One of these approaches is the canonical top- p (or nucleus) sampling method (Holtzman et al., 2020), which samples from top tokens that take up p proportion (e.g., 95%) of the probability mass at each decoding step. Even though these decoding methods work impressively well, they are post-hoc fixes rather than learning the LM accurately in the first place. Therefore, some other works criticize the MLE training objective and propose alternative loss functions.

2.3 Objectives Beyond MLE

Unlikelihood training (Welleck et al., 2020; Li et al., 2020) was proposed to penalize repetition (or any undesirable phenomenon) explicitly during training. The idea is to minimize the likelihood of a set of negative tokens at each generation step

during training. The selection of negative tokens is pre-defined, e.g., tokens that appear often in the previous context. MIXCE shares the same goal with unlikelihood training – matching the human LM distribution, but provides a more general approach without targeting any specific problem.

Similar to our motivation, Kang and Hashimoto (2020) think that the zero-avoiding property of MLE makes the model sensitive to dataset noise. To cover these noisy examples, the model has to put non-trivial probability mass on the $P(x) = 0$ area. To combat this problem, they propose a loss truncation method that drops high-loss (low-likelihood) examples during training time.

Pang and He (2021) want to address the mismatch of learning objective and human evaluation (likelihood vs. quality) and introduce the GOLD algorithm to approximate reverse cross-entropy. Our approximation is similar to theirs but has a different derivation process (see § 3.2). Moreover, GOLD is evaluated on controlled generation tasks (e.g., summarization and translation) in which the goal is to generate one high-quality text for each input, and diversity is not so important. In contrast, if we train the LM only with reverse CE till convergence, the model will deterministically produce the most likely text for each prompt, which is undesirable for an LM. Therefore, mixing forward and reverse CEs is necessary.

The idea of MIXCE is also relevant to GANs (Goodfellow et al., 2014). GANs optimize the Jensen–Shannon (JS) divergence between model and data distributions. Essentially, JS divergence is also for balancing the two driving forces of forward and reverse KL divergences (Huszár, 2015), and it has been successfully used for evaluating LM-generated text (Pillutla et al., 2021). However, probably due to the discrete nature of text, GANs have not been well applied to LM training. Caccia et al. (2020) show that previous language GANs often give up diversity for quality.

Another related work is Popov and Kudinov (2018), which finetunes LMs with the sum of the forward cross-entropy loss and reverse KL divergence. They train a discriminator to estimate reverse KL, similar to a GAN. On the other hand, we directly approximate reverse cross-entropy in our objective function, without training an additional discriminator.

Concurrently, with the same motivation as ours, Ji et al. (2023) propose to replace MLE with min-

imization of the total variation distance (TVD) (Van Handel, 2014) between data and model distributions. Notably, their final approximation of TVD, which they call TaiLr, is equivalent to forward cross-entropy when the hyperparameter $\gamma = 0$ and equals our approximated reverse cross-entropy when $\gamma = 1$.

3 Methodology

3.1 MIXCE

Our MIXCE learning objective for training LMs is the combination of forward and reverse cross-entropies, written as

$$-\eta \cdot \mathbb{E}_{x \sim P}[\log Q_\theta(x)] - (1 - \eta) \cdot \mathbb{E}_{x \sim Q_\theta}[\log P(x)] \quad (1)$$

where η is the mixing ratio. When $\eta = 1$, it becomes the normal MLE objective; and when $\eta = 0$, it is the reverse cross-entropy only.

The MIXCE loss can be understood in two ways. First, reverse and forward cross-entropy (CE) emphasize *quality* and *diversity* respectively. The reverse CE, $-\mathbb{E}_{x \sim Q_\theta}[\log P(x)]$, focuses on *quality* because it resembles how we conduct human evaluations – sampling from the model Q_θ and evaluating it by the human P . In human evaluations, the focus is more on the quality of the model-generated text. So, it is possible that a model always generates the same few high-quality texts, but still gets high human evaluation scores. This is similar to the *mode collapse* problem of GANs. The forward CE, $-\mathbb{E}_{x \sim P}[\log Q_\theta(x)]$, instead focuses more on *diversity* because it needs any sample from P to have a non-trivial probability under Q_θ (Hashimoto et al., 2019). Note that it does not mean forward CE has zero effect on quality, rather, the model likelihood $Q_\theta(x)$ only loosely correlates with the human-perceived quality of x (Zhang et al., 2021).

Second, we hypothesize that MIXCE works similarly to a mixture of forward and reverse KL divergences, which we will show empirically in our synthetic experiments (§ 4.1). On the one hand, minimizing forward KL is equivalent to optimizing forward CE. On the other hand, reverse KL divergence, $\mathbb{E}_{x \sim Q_\theta}[\log \frac{Q_\theta(x)}{P(x)}]$, has two parts: reverse CE and negative entropy of Q_θ , $\mathbb{E}_{x \sim Q_\theta}[\log Q_\theta(x)]$. Reverse CE is minimized when the model deterministically outputs the most likely example, i.e., $Q_\theta(x) = \delta$ (the most likely x under P). Instead, minimizing the negative entropy (maximizing the entropy) of the model encourages it to be as un-

certain as possible, i.e., having a large support and uniform distribution. This entropy term counteracts the narrowing-down effect of reverse CE. As discussed above, forward CE pushes the Q distribution to fully cover the support of P . In this case, forward CE can also help counteract the narrowing-down effect of reverse CE, i.e., the maximizing entropy term becomes less important when forward CE is present. Hence, we think it is reasonable to drop it from reverse KL.

Overall, MIXCE combines two complementary training signals, as shown in Figure 1. Reverse CE prevents the model distribution from being broader than the data distribution, while forward CE is more helpful for preventing the model distribution from being narrower than the data distribution. Although forward CE also has non-zero loss when the model distribution is too wide, its loss magnitude is much smaller than what reverse CE provides (see Appendix C for more discussion). When data is clean, two CEs work jointly to help learn the data distribution better. When data is noisy, the mixing ratio η allows us to trade-off between emphasizing a good coverage of the data and putting more weight on the actually high-quality sequences.

3.2 Optimization of Reverse CE

Optimizing MIXCE is non-trivial. The obstacle is to minimize the reverse CE, $-\mathbb{E}_{x \sim Q_\theta}[\log P(x)]$ with respect to θ . To this end, we need to know P and to have a differentiable sampling operation from Q_θ . In our synthetic experiments (§ 4.1), we use a distribution P of our own construction and use Gumbel-Softmax (Jang et al., 2017; Maddison et al., 2017) to make the sampling operation differentiable.

However, in a real setting, we do not know P . To deal with this, we take the following steps to derive an approximated reverse cross-entropy (we omit the negative sign for simplicity):

$$\nabla_\theta \mathbb{E}_{x \sim Q_\theta}[\log P(x)] \quad (2)$$

$$\approx \nabla_\theta \mathbb{E}_{x \sim Q_\theta}[P(x)] \quad (3)$$

$$= \sum_x \nabla_\theta Q_\theta(x) P(x) \quad (4)$$

$$= \sum_x Q_\theta(x) \nabla_\theta \log Q_\theta(x) P(x) \quad (5)$$

$$= \sum_x P(x) Q_\theta(x) \nabla_\theta \log Q_\theta(x) \quad (6)$$

$$= \mathbb{E}_{x \sim P}[Q_\theta(x) \nabla_\theta \log Q_\theta(x)] \quad (7)$$

$$= \mathbb{E}_{x \sim P} \left[\prod_{t=1}^T Q_{\theta}(x_t | x_{<t}) \sum_{t=1}^T \nabla_{\theta} \log Q_{\theta}(x_t | x_{<t}) \right] \quad (8)$$

$$\approx \mathbb{E}_{x \sim P} \left[\sum_{t=1}^T Q_{\theta}(x_t | x_{<t}) \nabla_{\theta} \log Q_{\theta}(x_t | x_{<t}) \right] \quad (9)$$

First, from (2) to (3), we substitute expected log-likelihood by *expected accuracy*. Irsoy (2019) shows that expected accuracy is a comparable or better alternative loss function to cross-entropy for classification tasks. Then, following the Policy Gradient theorem (Williams, 1992; Sutton et al., 1999), we get (4) and (5), where we view model Q_{θ} as the policy and $P(x)$ as the reward we want to optimize for the whole sequence. Next, we switch from the expectation of Q_{θ} to the expectation of P (from (5) to (6) and (7)), so that we can use the offline samples from P (data samples in the training set) instead of online sampling from Q_{θ} . We unfold $Q_{\theta}(x)$, which results in (8). Until this point, theoretically, we are already able to optimize the model using Equation (8) without knowing P . However, the product of $Q_{\theta}(x_t | x_{<t})$ has a very high variance, and in practice, it underflows when T is large. Therefore, we apply a final rough approximation that leads to (9).

Equations (8) and (9) are apparently not equivalent to each other. Nonetheless, they have similar effects. Intuitively, in (8), we weigh the gradients of each sequence differently based on their sequence-level probabilities, $Q_{\theta}(x)$; in other words, it promotes high-likelihood sequences. Similarly, (9) weighs gradients at each step by $Q_{\theta}(x_t | x_{<t})$, i.e., promoting high-likelihood tokens at each step. So essentially, they both *encourage the model to produce generations in which it is already confident*. We call it a *self-reinforced* objective. To further illustrate why *self-reinforcement* makes sense, we conduct an analysis using GPT-2 (Radford et al., 2019). Please refer to Appendix B for a detailed discussion. In short, we show that MLE-pretrained GPT-2 on average assigns a higher probability to human text than to text sampled from the model. Therefore, when we promote high-probability sequences or tokens, it is like “pushing” the model distribution toward the human distribution. But, we need to avoid overly “pushing” it to the extremely high-probability region where repetitive greedy search outputs locate.

Note that our approximation of reverse cross-entropy is relevant to the method proposed by Pang

and He (2021), though we have a different derivation process from theirs. Please see Appendix A for a detailed comparison.

Finally, combining forward CE and Equation (9), our approximated MIXCE objective is to maximize

$$\mathbb{E}_{x \sim P} \left[\sum_{t=1}^T (\eta + (1 - \eta) \cdot Q_{\theta}(x_t | \cdot)) \nabla_{\theta} \log Q_{\theta}(x_t | \cdot) \right], \quad (10)$$

where $Q_{\theta}(x_t | \cdot)$ is short for $Q_{\theta}(x_t | x_{<t})$. This loss function has the same computational complexity as forward CE (MLE). Since $Q_{\theta}(x_t | x_{<t})$ is strictly lower than 1 (it is around 0.017 to 0.13 when using GPT-2), the gradient from approximated reverse CE is smaller than that from forward CE. Therefore, it is important to tune η to balance the effects of two CEs.

4 Experiments

4.1 Synthetic Experiments

We first conduct experiments in a synthetic ideal setting, where we know P , to show the effectiveness of mixing two cross-entropies with or without approximation. Moreover, during evaluation, we can directly compare the learned model parameters against the ground truth parameters of P .

Define the “human” LM P . We start by defining P as a bi-gram LM. Bi-gram means that the prediction of the next token only depends on the immediately previous token, i.e., $P(x_t | x_{t-1})$. Therefore, P is determined by a transition matrix among words $\mathbf{M} \in \mathbb{R}^{V \times V}$ (V =vocabulary size) and a start token probability distribution $\pi \in \mathbb{R}^V$, i.e., stochastic finite-state automata. The last token in the vocabulary is the end-of-sequence (EOS) token. For simplicity, we initialize π as a uniform distribution. To initialize \mathbf{M} , we use two methods. The first is **random initialization**. We sample categorical distributions from a Dirichlet ($\alpha=0.5$) prior to initialize each row of \mathbf{M} . However, one remaining problem is that P has support everywhere. To have $P = 0$ areas, we randomly assign 0s to a certain percent of values in each row of \mathbf{M} and then re-normalize to sum to 1.³ We test 3 percentages: 10%, 50%, and 90%. The second is **initialization using real data**. We sample 5000 pieces of text from WebText (Radford et al., 2019), count the occurrence of bigrams, and then use the occurrence to

³When we assign 0s, we make sure every token has non-zero transition probability to EOS.

initialize \mathbf{M} . In this case, there are naturally 0s in \mathbf{M} , and the larger the vocabulary size is, the sparser \mathbf{M} is. No matter which initialization is used, we reserve the last row of \mathbf{M} for EOS and it has all 0s, i.e., will not transit to any token. We set the vocabulary size $V=20, 50, 100, 500, \text{ or } 1000$.⁴

Learn an LM Q_θ . We implement model Q_θ as a simple neural bigram LM. Given the word embedding e_{i-1} of the previous token x_{i-1} , the next token is predicted via a simple neural network f :

$$h_{i-1} = \text{Dropout}(\text{ReLU}(\mathbf{W}_1 e_{i-1} + \mathbf{b}_1)),$$

$$Q(x_i|x_{i-1}) = \text{Softmax}(\mathbf{W}_2 h_{i-1} + \mathbf{b}_2),$$

where $\mathbf{W}_1 \in \mathbb{R}^{d \times d}$ (d is the hidden dimension size), $\mathbf{b}_1 \in \mathbb{R}^d$, $\mathbf{W}_2 \in \mathbb{R}^{d \times V}$, and $\mathbf{b}_2 \in \mathbb{R}^V$ are model parameters. After training this model, the learned transition matrix can be obtained by $\mathbf{M}' = f(\mathbf{E})$, \mathbf{E} is the word embedding matrix.

Synthetic data. We sample sequences from P . We set the max sequence length as 500. We sample 50K and 5K sequences as the training and validation set, respectively. There is no test set because we directly compare the learned transition matrix \mathbf{M}' to the gold \mathbf{M} during evaluation.

Metrics. (1) **avg. js**: we compute the JS divergence between each row (except the last row) of \mathbf{M}' and the corresponding row in \mathbf{M} , and then average across rows. This metric evaluates the overall divergence of \mathbf{M}' from \mathbf{M} , and equals 0 iff $\mathbf{M}' = \mathbf{M}$; (2) **avg. 0s**: we get the probabilities from \mathbf{M}' from positions where the corresponding gold probabilities are 0 in \mathbf{M} , and take their average. If $\mathbf{M}' = \mathbf{M}$, avg. 0s = 0, but vice versa is not true.

Objectives. (1) **Forward KL**, $\text{KL}(P||Q_\theta) = \mathbb{E}_{x \sim P}[\log P(x)/Q_\theta(x)]$, which is equivalent to MLE; (2) **Reverse KL**, $\text{KL}(Q_\theta||P) = \mathbb{E}_{x \sim Q_\theta(x)}[\log Q_\theta(x)/P(x)]$; (3) **Mixture of two KLs**, $\eta \cdot \text{KL}(P||Q_\theta) + (1 - \eta) \cdot \text{KL}(Q_\theta||P)$; (4) **JS**, we use a general definition of JS divergence (Huszár, 2015), $\eta \cdot \text{KL}(P||M) + (1 - \eta) \cdot \text{KL}(Q_\theta||M)$, where $M = \eta \cdot P + (1 - \eta) \cdot Q_\theta$; ⁵(5) **Oracle mixture of cross-entropies (MIXCE*)**, where we use the known P . (6) **Approximated**

⁴Our defined bi-gram LMs are always *tight*, i.e., do not “leak” probability mass onto infinite sequences because we make sure that all accessible tokens also have non-zero paths to other tokens. Please refer to Du et al. (2022) for the proof.

⁵When $\eta = 0.5$, it is the same as the objective of GAN (Goodfellow et al., 2014). But instead of using GAN’s min-max loss, we directly optimize JS because we know P .

Vocab	Objective	Random (50%)		WebText	
		avg. js	avg. 0s	avg. js	avg. 0s
	Gold	0.0	0.0	0.0	0.0
20	For. KL	7.40e-4	1.44e-4	9.93e-4	1.79e-4
	Rev. KL	1.36e-1	7.42e-6	3.93e-3	1.95e-6
	Mix KLs	4.89e-4	5.15e-5	9.91e-4	1.11e-5
	JS	2.14e-1	4.88e-5	1.12e-2	5.84e-6
	MIXCE*	8.12e-4	1.05e-4	1.36e-3	1.19e-4
	MIXCE	7.02e-4	1.25e-4	1.00e-3	1.79e-4
50	For. KL	6.47e-3	5.65e-4	4.30e-3	4.77e-4
	Rev. KL	4.29e-1	1.53e-3	3.48e-2	5.30e-5
	Mix KLs	4.45e-3	2.80e-4	3.91e-3	2.83e-4
	JS	4.74e-1	1.40e-3	9.23e-3	2.48e-5
	MIXCE*	4.49e-3	3.72e-4	3.94e-3	2.75e-4
	MIXCE	6.47e-3	5.64e-4	4.29e-3	4.77e-4
100	For. KL	3.56e-2	1.44e-3	9.70e-3	3.10e-4
	Rev. KL	5.57e-1	3.62e-4	1.00e-1	4.04e-5
	Mix KLs	2.74e-2	2.10e-4	9.19e-3	1.84e-4
	JS	5.53e-1	9.69e-4	1.73e-1	5.56e-4
	MIXCE*	2.85e-2	9.16e-4	9.61e-3	1.87e-4
	MIXCE	3.56e-2	1.41e-3	9.69e-3	3.16e-6
500	For. KL	2.39e-1	1.49e-3	4.60e-2	1.78e-4
	Rev. KL	6.78e-1	2.76e-6	3.05e-1	1.68e-5
	Mix KLs	2.32e-1	8.60e-4	4.27e-2	1.33e-4
	JS	5.34e-1	7.19e-4	2.78e-1	3.84e-5
	MIXCE*	2.34e-1	1.38e-3	4.23e-2	1.29e-4
	MIXCE	2.35e-1	1.46e-3	4.53e-2	1.64e-4
1000	For. KL	2.93e-1	8.80e-4	8.10e-2	1.50e-4
	Rev. KL	6.85e-1	1.21e-6	3.30e-1	6.26e-6
	Mix KLs	2.91e-1	8.57e-4	7.50e-2	1.17e-4
	JS	4.59e-1	5.97e-4	3.02e-1	1.93e-5
	MIXCE*	2.92e-1	8.58e-4	7.44e-2	1.14e-4
	MIXCE	2.92e-1	8.76e-4	7.94e-2	1.42e-4

Table 1: Synthetic experimental results. Random (50%) randomly initializes \mathbf{M} and sets 50% of the probabilities to 0. WebText means initializing \mathbf{M} by the bigram occurrence in the WebText data. Gold refers to the results when $\mathbf{M}' = \mathbf{M}$. *avg. js* is our main metric, which represents the average JS divergence between \mathbf{M} and \mathbf{M}' (please see the definition of *avg. 0s* in text). Each number is a 5-seed average, and Table 7 shows the 95% confidence intervals of some experiments.

mixture of cross-entropies (MIXCE), where we assume P is unknown. Except for Forward KL and MIXCE, the other four objectives all need to sample from Q_θ and require gradients to pass through this sampling operation. To this end, we use Gumbel-Softmax (Jang et al., 2017; Maddison et al., 2017) to make sampling differentiable.

Model selection. During training, we check the validation loss (the value of the objective function) after every epoch and only save the best checkpoint that has the lowest validation loss. For objectives with η , we choose the best η based on the avg. js result on the validation set. We report a 5-seed average for each experiment. The search space of η is [0.99, 0.9, 0.5, 0.1, 0.01]. Selected best η s are reported in Table 11 in the Appendix.

Model Size	Objective	WikiText				WebText				WritingPrompts			
		ppl	div	mauve	coh	ppl	div	mauve	coh	ppl	div	mauve	coh
	Human	-	0.89	1.0	0.628	-	0.84	1.0	0.633	-	0.85	1.0	0.473
Small	MLE	26.98	0.91	0.67	0.556	21.45	0.87	0.90	0.555	28.45	0.87	0.85	0.397
	MIXCE	35.04	0.87	0.93	0.567	21.69	0.85	0.92	0.565	28.79	0.86	0.89	0.403
Medium	MLE	20.43	0.90	0.73	0.573	15.92	0.87	0.88	0.560	22.72	0.88	0.89	0.414
	MIXCE	25.92	0.88	0.95	0.584	16.51	0.83	0.93	0.585	23.04	0.86	0.91	0.419
Large	MLE	18.24	0.90	0.75	0.567	14.13	0.87	0.81	0.570	21.95	0.87	0.87	0.425
	MIXCE	23.44	0.88	0.95	0.578	14.66	0.82	0.94	0.592	21.04	0.86	0.94	0.429

Table 2: Unbiased sampling results of models finetuned by MLE or MIXCE on three datasets. For all metrics, the closer to the human scores the better. **Bold** numbers are the ones that are closer to human scores in each setting. Each number is a 3-run average.

Model Size	Objective	WikiText				WebText				WritingPrompts			
		best p	div	mauve	coh	best p	div	mauve	coh	best p	div	mauve	coh
	Human	-	0.89	1.0	0.628	-	0.84	1.0	0.633	-	0.85	1.0	0.473
Small	MLE	0.85	0.89	0.93	0.584	0.93	0.84	0.94	0.580	0.97	0.86	0.90	0.410
	MIXCE	0.99	0.87	0.95	0.568	0.99	0.84	0.93	0.571	0.99	0.85	0.90	0.407
Medium	MLE	0.85	0.88	0.95	0.602	0.93	0.85	0.95	0.592	0.97	0.86	0.92	0.428
	MIXCE	0.99	0.87	0.96	0.590	0.99	0.81	0.93	0.594	0.99	0.85	0.92	0.427
Large	MLE	0.87	0.89	0.96	0.594	0.95	0.84	0.87	0.593	0.99	0.86	0.89	0.430
	MIXCE	0.99	0.87	0.97	0.580	0.99	0.81	0.94	0.601	0.99	0.86	0.94	0.435

Table 3: Top- p sampling results of the same models as Table 2. Since changing the decoding method will not affect perplexity, we report the selected best p instead.

Results. Table 1 (and Table 6 in the Appendix) shows the results of our synthetic experiments. Across 4 kinds of initialization of M and 5 vocabulary sizes, we observe some common patterns. First, the mixture of two KLs often gets the best avg. js compared to other objectives, and MIXCE* usually comes second. This supports our expectation that the mixture of two cross-entropies approximates the mixture of two KLs (§ 3.1), as well as demonstrates that combining two KLs or CEs can help learn the data distribution more accurately compared to MLE. Second, the approximated MIXCE usually under-performs MIXCE* but outperforms forward KL (MLE). Third, reverse KL generally works best for the avg. 0s metric, due to its property of *zero-forcing* – forcing $Q_\theta(x) = 0$ when $P(x) = 0$. Lastly, JS divergence oftentimes works similarly to reverse KL, which is consistent with the observation made by Caccia et al. (2020) – language GANs trade off diversity for quality.

4.2 GPT-2 Experiments

Next, we test MIXCE in a real setting where we do not know P , but we have finite samples from P . We use GPT-2 (Radford et al., 2019) as the LM Q_θ . Though GPT-2 models are already pre-trained by MLE, for simplicity, we use different objectives to finetune it. We test GPT-2 in 3 sizes:

small (24M), medium (355M), and large (774M). See more implementation details in Appendix G.

Real data. We use English text data from 3 domains: (1) WikiText (Merity et al., 2017): text from Wikipedia; (2) WebText (Radford et al., 2019): text from the Web. It was used for pretraining GPT-2; and (3) WritingPrompts (Fan et al., 2018): text from the writing prompts forum of Reddit. We sample from each of these 3 datasets to form our training, development, and test sets. By default, our training/development/test set contains 50K/5K/5K examples. Please find more details about these datasets in Appendix G.

Metrics. (1) **Perplexity (ppl)** is defined as $e^{-\frac{1}{N \times T} \sum_N \sum_T \log_e Q_\theta(x_t | x_{<t})}$, where N is the number of examples and T is the sequence length. Perplexity is not necessarily correlated with human perceived quality (Zhang et al., 2021). (2) **Diversity (div)**: following Meister et al. (2022), we define n -gram diversity as the average fraction of unique vs. total n -grams for $n \in \{1, 2, 3, 4\}$ in each piece of text. (3) **Mauve** (Pillutla et al., 2021) compares model-generated text against human text via a KL divergence curve and is the state-of-the-art metric for open-ended text generation. We use Mauve as our primary metric. (4) **Coher-**

ence (coh) (Su et al., 2022) computes the cosine similarity between the embedding of prompt and the embedding of continuation, and embeddings are from SimCSE (Gao et al., 2021). All metrics are *the closer to human scores the better*.

Objectives. Since we have no access to P , we can only implement two out of the six objectives we test in the synthetic setting: (1) **MLE**, which is equal to forward CE or forward KL; (2) **MIXCE**, the approximated mixture of cross-entropies.

Decoding. We use **unbiased sampling** (see footnote 1) as our primary decoding method as it allows us to explore the learned distribution in an unbiased way (Eikema and Aziz, 2020). Additionally, we test **top- p sampling** (Holtzman et al., 2020) to check if MIXCE is complementary to advanced decoding methods, and we carefully tune p on the development set. For each text, we take the first 50 tokens (by GPT-2 tokenizer) as the prompt and set the max generation length as 512.

Model selection. We finetune the model for 5 epochs on the training set and save the best checkpoint with the lowest dev loss. We select the best mixing ratio η and the best p based on the Mauve score on the dev set. The search space of η is [0.99, 0.9, 0.7, 0.5, 0.3, 0.1, 0.01, 0.0] and that of p is [0.85, 0.87, 0.89, 0.91, 0.93, 0.95, 0.97, 0.99]. Selected best η s are reported in Table 12 in the Appendix. Best p s are reported in Table 3. Metric scores are reported on the test set and are 3-run averages because sampling is stochastic.

Results. Table 2 shows unbiased sampling results of models in different sizes and finetuned with different objectives on three datasets. As you can see, MIXCE-finetuned models usually get worse perplexity but consistently better diversity, mauve, and coherence, compared to MLE-finetuned models. Table 3 shows top- p sampling results from the same models as Table 2. Since perplexity will not change as the decoding method changes, we instead report the selected best p in this table. It can be seen that after carefully applying top- p sampling, MIXCE-finetuned models work on par with MLE-finetuned models for diversity, mauve, and coherence. Nonetheless, the best p for MIXCE models is always 0.99, while MLE models have smaller and more diverse p s. This indicates that MIXCE leads to a less noisy model distribution.

Dataset	Which is better?		
	MIXCE	MLE	Same
WikiText	135*	85	95
WebText	139*	79	97
WritingPrompts	111	119	85

Table 4: Human evaluation results. The star (*) means significantly⁶ better ($p < 0.01$).

Human evaluation. Besides automatic metrics, we also conduct a human evaluation. Following Krishna et al. (2022), we conduct blind A/B testing. We randomly sample 105 examples from each dataset. For each example, we ask humans to read two generations from MLE and MIXCE-finetuned GPT-2 large models, respectively, and the order of showing these two generations is random. We use unbiased sampling to get the generations. Then, we ask them to judge which one is better (or they are the same) and justify their preference, based on fluency, coherence, informativeness, and whether it is sensical. We conduct this evaluation on Amazon Mechanical Turk and collect 3 responses for each example. Please refer to Appendix F for more details and examples. The final results are shown in Table 4. As you can observe, MIXCE-finetuned models significantly outperform MLE-finetuned models on both WikiText and WebText domains, while the two methods perform similarly on WritingPrompts. It is also worth noting that, compared to the results shown in Table 2, none of the 4 automatic metrics share the same trend with human evaluation.

4.3 Robustness & Analysis

Varying training data sizes. We test 3 other training data sizes: 10K, 25K, and 100K using GPT-2 small. Table 5 in the Appendix contains the results, and it shares the same story trend as Table 2: MIXCE-finetuned models get worse perplexity but in general work better than MLE-finetuned models for diversity, mauve, and coherence.

Varying η and max generation length. To examine how the mixing ratio η and the max generation length affect the performance, we show the mauve score curves on the dev set in Figure 4. The x-axis is the mixing ratio η from 0 to 1 (MIXCE=MLE when $\eta = 1$), and the y-axis is the mauve score with different max generation lengths (128, 320,

⁶The significance test is conducted following the bootstrap test setup (Efron and Tibshirani, 1994).

and 512). First, reasonable performances are usually observed when $\eta \geq 0.1$, and only training the models with approximated reverse CE (i.e., $\eta = 0$) leads to degeneration. Second, the advantage of MIXCE is more prominent when the max generation length is longer.

Controlled Mauve. The max generation length is not the actual text length because when sampling from the model, EOS can be generated at any step. We find that the actual *text length* can affect the mauve computation. Even if we truncate all texts to the same length, the *incompleteness* caused by truncation can be another confounding factor. Both text length and text completeness are irrelevant to text quality but can be used by mauve to distinguish model generations from human texts. Therefore, to eliminate the influence of these confounding factors, we propose a *controlled mauve* (or *c-mauve*) computation approach. Concretely, for human texts and model generations, we randomly sample 10K L -length text fragments from each of these two sets. L is the number of tokens. Then, we compute the mauve between these two sets of fragments. Table 8 shows the results. As you can see, c-mauve scores are in general very high (≥ 0.90), which may indicate that, after controlling the confounding factors, the ability of mauve to distinguish model text from human text has been weakened. MIXCE still gets better performance than MLE in most cases. Besides, we also compute controlled coherence in the same fashion, and MIXCE retains its advantage. Please refer to Appendix D.4 for more details about controlled Mauve and Coherence.

5 Conclusion

We propose a novel training objective, MIXCE, for autoregressive language modeling. MIXCE combines forward and reverse cross-entropies, which can be viewed as combining two complementary driving forces for better fitting the model distribution to the data distribution. We demonstrate the superiority of MIXCE over MLE in both synthetic and real settings via both automatic and human evaluations. In the future, MIXCE can be potentially used for pretraining language models.

Acknowledgments

We thank anonymous reviewers for their valuable comments. We thank Xiang Zhou for the helpful discussions. This work was supported by a Bloomberg Data Science Ph.D. Fellowship.

Limitations

One apparent disadvantage of MIXCE is the mixing ratio η . As shown in Table 12 and Figure 4, the best η changes as the experimental setting changes. It may be because we use mauve as the model selection criteria or because different datasets have different noise levels. In general, we do not have a good answer to which η should be used. The ideal solution is to select η based on the performance of the development set like what we did. However, in pretraining settings, it is too expensive to search over multiple η s. Therefore, how to find a universal η or how to determine η automatically is an important problem to resolve before MIXCE can be reliably used for pretraining.

As we mentioned in § 1, language degeneration of open-ended generation shows two distinct patterns: the non-sensical text from unbiased sampling and the repetition loops from greedy search. Though MIXCE helps improve the performance of sampling, we still see repetition loops when using greedy search.

Ethical Considerations

As the OpenAI team pointed out, GPT-2 does not distinguish fact from fiction, so it can not support use cases that require the generated text to be true. Additionally, GPT-2 reflect the biases inherent to the systems they were trained on, so it can not be deployed into systems that interact with humans unless the deployers first carry out a study of biases relevant to the intended use case. Though our MIXCE-finetuned GPT-2 gets improved performance with respect to the metrics we used, the above statement still holds. At this point, we are not sure whether MIXCE can help improve factuality or lead to less biased generations, but we are sure that the generations still have non-factual content and biases.

References

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. [Neural machine translation by jointly learning to align and translate](#). In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- Sourya Basu, Govardana Sachitanandam Ramachandran, Nitish Shirish Keskar, and Lav R. Varshney. 2021. [{MIROSTAT}: A {neural} {text} {decoding} {algorithm} {that} {directly} {controls} {perplexity}](#).

- In *International Conference on Learning Representations*.
- Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. 2000. [A neural probabilistic language model](#). In *Advances in Neural Information Processing Systems*, volume 13. MIT Press.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS'20*, Red Hook, NY, USA. Curran Associates Inc.
- Massimo Caccia, Lucas Caccia, William Fedus, Hugo Larochelle, Joelle Pineau, and Laurent Charlin. 2020. [Language gans falling short](#). In *International Conference on Learning Representations*.
- Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.
- Li Du, Lucas Torroba Hennigen, Tiago Pimentel, Clara Meister, Jason Eisner, and Ryan Cotterell. 2022. A measure-theoretic characterization of tight language models. *arXiv preprint arXiv:2212.10502*.
- Bradley Efron and Robert J Tibshirani. 1994. *An introduction to the bootstrap*. CRC press.
- Bryan Eikema and Wilker Aziz. 2020. [Is MAP decoding all you need? the inadequacy of the mode in neural machine translation](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4506–4520, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Angela Fan, Mike Lewis, and Yann Dauphin. 2018. [Hierarchical neural story generation](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.
- Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. [SimCSE: Simple contrastive learning of sentence embeddings](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *NIPS*.
- Tatsunori B. Hashimoto, Hugh Zhang, and Percy Liang. 2019. [Unifying human and statistical evaluation for natural language generation](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1689–1701, Minneapolis, Minnesota. Association for Computational Linguistics.
- John Hewitt, Christopher D. Manning, and Percy Liang. 2022. [Truncation sampling as language model desmoothing](#). In *Findings of the Conference on Empirical Methods in Natural Language Processing (Findings of EMNLP)*.
- Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. [The curious case of neural text de-generation](#). In *International Conference on Learning Representations*.
- Ferenc Huszár. 2015. How (not) to train your generative model: Scheduled sampling, likelihood, adversary? *arXiv preprint arXiv:1511.05101*.
- Ozan Irsoy. 2019. [On expected accuracy](#). *arXiv preprint arXiv:1905.00448*.
- Eric Jang, Shixiang Gu, and Ben Poole. 2017. [Categorical reparameterization with gumbel-softmax](#). In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24–26, 2017, Conference Track Proceedings*. OpenReview.net.
- Haozhe Ji, Pei Ke, Zhipeng Hu, Rongsheng Zhang, and Minlie Huang. 2023. [Tailoring language generation models under total variation distance](#). In *The Eleventh International Conference on Learning Representations*.
- Daniel Kang and Tatsunori B. Hashimoto. 2020. [Improved natural language generation via loss truncation](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 718–731, Online. Association for Computational Linguistics.
- Kalpesh Krishna, Yapei Chang, John Wieting, and Mohit Iyyer. 2022. Rankgen: Improving text generation with large ranking models. In *Empirical Methods in Natural Language Processing*.
- Margaret Li, Stephen Roller, Iliia Kulikov, Sean Welleck, Y-Lan Boureau, Kyunghyun Cho, and Jason Weston. 2020. [Don't say that! making inconsistent dialogue unlikely with unlikelihood training](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4715–4728, Online. Association for Computational Linguistics.

- Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2022. [Contrastive decoding: Open-ended text generation as optimization](#).
- Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. 2017. [The concrete distribution: A continuous relaxation of discrete random variables](#). In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.
- Clara Meister, Tiago Pimentel, Gian Wiher, and Ryan Cotterell. 2022. [Locally typical sampling](#). *Transactions of the Association for Computational Linguistics*, abs/2202.00666.
- Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. [Pointer sentinel mixture models](#). In *International Conference on Learning Representations*.
- Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernocký, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Interspeech*, volume 2, pages 1045–1048. Makuhari.
- Kevin P Murphy. 2012. *Machine learning: a probabilistic perspective*. MIT press.
- Myle Ott, Michael Auli, David Grangier, and Marc’Aurelio Ranzato. 2018. [Analyzing uncertainty in neural machine translation](#). In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3956–3965. PMLR.
- Richard Yuanzhe Pang and He He. 2021. [Text generation by learning from demonstrations](#). In *ICLR*.
- Krishna Pillutla, Lang Liu, John Thickstun, Sean Welleck, Swabha Swayamdipta, Rowan Zellers, Sewoong Oh, Yejin Choi, and Zaid Harchaoui. 2022. Mauve scores for generative models: Theory and practice. *arXiv preprint arXiv:2212.14578*.
- Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. 2021. Mauve: Measuring the gap between neural text and human text using divergence frontiers. *Advances in Neural Information Processing Systems*, 34:4816–4828.
- Vadim Popov and Mikhail Kudinov. 2018. Fine-tuning of language models with discriminator. *arXiv preprint arXiv:1811.04623*.
- Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*.
- Felix Stahlberg and Bill Byrne. 2019. [On NMT search errors and model errors: Cat got your tongue?](#) In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3356–3362, Hong Kong, China. Association for Computational Linguistics.
- Yixuan Su, Tian Lan, Yan Wang, Dani Yogatama, Lingpeng Kong, and Nigel Collier. 2022. [A contrastive framework for neural text generation](#). In *Advances in Neural Information Processing Systems*.
- Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.
- Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12.
- L Theis, A van den Oord, and M Bethge. 2016. A note on the evaluation of generative models. In *International Conference on Learning Representations (ICLR 2016)*, pages 1–10.
- Ramon Van Handel. 2014. Probability in high dimension. Technical report, PRINCETON UNIV NJ.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2020. [Neural text generation with unlikelihood training](#). In *International Conference on Learning Representations*.
- Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256.
- Hugh Zhang, Daniel Duckworth, Daphne Ippolito, and Arvind Neelakantan. 2021. [Trading off diversity and quality in natural language generation](#). In *Proceedings of the Workshop on Human Evaluation of NLP Systems (HumEval)*, pages 25–33, Online. Association for Computational Linguistics.
- Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. [Opt: Open pre-trained transformer language models](#).

Appendix

A Connection to Pang and He (2021)

In Section 3.2, we introduce an approximation of the reverse cross-entropy (CE) objective. Similarly,

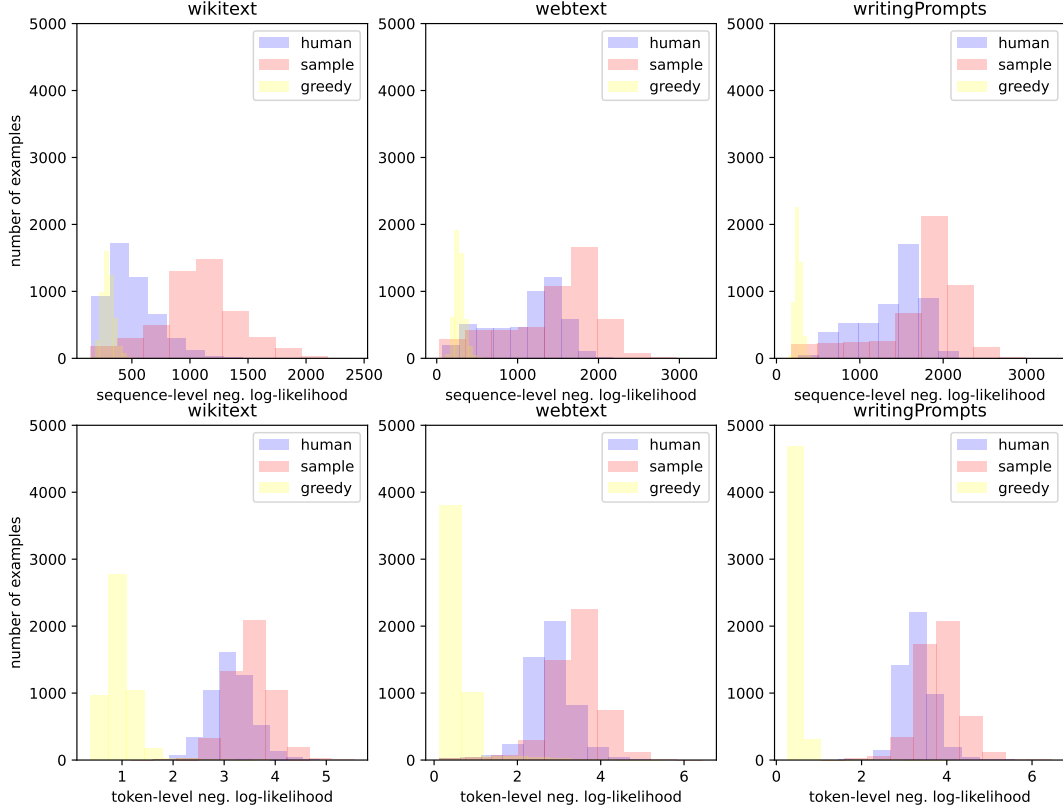


Figure 2: The histograms of sequence-level and token-level negative log-likelihoods of human texts and model generations from GPT-2 large.

Pang and He (2021) also propose to approximate reverse CE, and the resulting GOLD algorithm is similar to our Equation 9. Here, we would like to clarify the difference and connection.

The following equation is the start policy gradient equation used by Pang and He (2021).

$$\mathbb{E}_{\tau \sim \pi_{\theta}} \left[\sum_t \nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \hat{Q}(s_t, a_t) \right]$$

They used different notations from ours. π_{θ} is the same as our Q_{θ} , i.e., $\pi_{\theta}(a_t | s_t)$ is the same as our $Q_{\theta}(x_t | x_{<t})$. \hat{Q} is the accumulated future reward from timestamp t , $\sum_{t'=t}^T \gamma^{t'-t} r_{t'}$, γ is the decay factor and $r_{t'}$ is the reward for each step. We will discuss \hat{Q} in detail later.

Then, they apply importance sampling to sample from a different behavioral policy π_b . Since they also use examples from the training set, their π_b is the same as our human (or data) distribution P .

$$\mathbb{E}_{\tau \sim \pi_b} \left[\sum_t w_t \nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \hat{Q}(s_t, a_t) \right]$$

w_t is the importance weight. They use a per-action approximation: $w_t \approx \frac{\pi_{\theta}(a_t | s_t)}{\pi_b(a_t | s_t)}$, which is similar to how we get Equation 9 from Equation 8.

Since π_b is unknown, they assume a uniform distribution: $\pi_b \approx 1/N$ (N is the number of training examples). Hence, their final approximated gradient is:

$$\mathbb{E}_{\tau \sim \pi_b} \left[\sum_t \pi_{\theta}(a_t | s_t) \nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \hat{Q}(s_t, a_t) \right]$$

They define $r_{t'}$ and \hat{Q} in three ways. The first is called δ -reward, i.e., $\hat{Q} = 1$. In this case, their final gradient is exactly the same as our Equation 9. However, as you can see, we take a different path of derivation. Instead of using this δ -reward, our \hat{Q} is the sequence-level reward $P(x)$. The reward $P(x)$ nicely helps us to switch from the expectation of Q_{θ} to the expectation of P (from Equation 5 to Equation 7). Therefore, without assuming a uniform distribution of π_b , our π_b is just P .

When using the other two rewards, they also need to know P . To address this, they use an MLE-pretrained model as a proxy of P .

Overall, we introduce a different derivation approach for approximating reverse CE. Moreover, as we mentioned in § 2.3, Pang and He (2021) focused on improving controlled generation tasks where the focus is on the quality of the text, while we focus

Data Size	Objective	WikiText				WebText				WritingPrompts			
		ppl	div	mauve	coh	ppl	div	mauve	coh	ppl	div	mauve	coh
	Human	-	0.89	1.0	0.628	-	0.84	1.0	0.633	-	0.85	1.0	0.473
10K	MLE	29.23	0.91	0.60	0.537	22.03	0.88	0.82	0.542	30.40	0.88	0.74	0.385
	MIXCE	36.70	0.88	0.93	0.546	22.79	0.83	0.86	0.562	30.65	0.87	0.81	0.395
25K	MLE	27.90	0.91	0.68	0.545	21.75	0.88	0.86	0.547	29.37	0.88	0.79	0.394
	MIXCE	35.73	0.88	0.94	0.562	21.97	0.85	0.88	0.561	29.67	0.86	0.86	0.401
100K	MLE	25.93	0.90	0.69	0.559	21.31	0.87	0.90	0.556	27.63	0.87	0.88	0.401
	MIXCE	34.13	0.87	0.93	0.575	21.58	0.85	0.92	0.566	28.01	0.85	0.90	0.409

Table 5: Unbiased sampling results of GPT-2 small models finetuned by MLE or MIXCE on three datasets of different training data sizes. All metrics are the closer to the human scores the better. **Bold** numbers are the ones that are closer to human scores in each setting.

on open-ended generations where quality and diversity are both important. Therefore, we mix reverse CE with forward CE to form our MIXCE learning objective.

B Intuition behind the Self-reinforced Objective

To further illustrate why this *self-reinforced* objective (Equation (8) or (9)) makes sense and their shortcomings, we conduct an analysis using GPT-2 large (Radford et al., 2019). We first sample 5000 pieces of text from WikiText, WebText, and WritingPrompts, respectively, and we call them *human* texts. Then, using the first 50 tokens of each human text as a prompt, we get 5000 sampling and greedy search generations from pretrained GPT-2 large (max generation length = 512). Next, we use the same model to score human texts and model generations and get the sequence-level and token-level negative log-likelihoods. Figure 2 shows the histograms of these negative log-likelihoods.

In Figure 2, we take the human text histogram (in blue) as a proxy of *human distribution* and the sampling text histogram (in red) as a proxy of *model distribution*. As you can see, the support of model distribution usually contains the support of human distribution. It supports our previous claim that MLE-trained models tend to over-generalize. Meanwhile, at both the sequence and the token levels, the model on average assigns a higher probability to human text than to text sampled from the model. Therefore, when we promote high-probability sequences or tokens, it is equivalent to pushing the model distribution toward the human distribution. However, we need to avoid overly pushing it to the extremely high-probability region where greedy search outputs locate (in yellow) because they are known to be poor-quality and repeti-

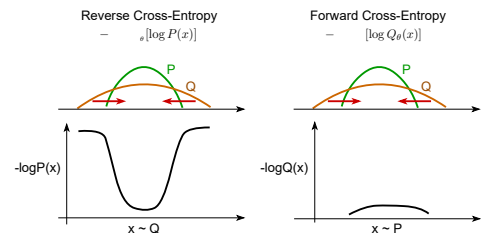


Figure 3: Forward CE only weakly penalizes the model Q_θ when it puts a small amount of probability mass onto $P(x) = 0$ space. And the loss magnitude is much smaller than what we will get from reverse CE.

Also, as shown in the figure, when promoting high-probability *sequences*, even if we overdo it, it will still be within the support of human distribution. In contrast, when promoting high-probability *tokens*, it can go outside the support of the human distribution, which is the drawback of Equation (9) compared to Equation (8).

Lastly, if we train the model only with the self-reinforced objective till convergence, it is inevitable to end up with a model that can only output greedy search generations. Hence, we need to combine it with the forward cross-entropy.

C Loss Magnitude

As shown in Figure 1, we use reverse cross-entropy (CE) to provide a driving force for narrowing the model distribution down when it is broader than the data distribution. And forward CE is to broaden the model distribution up. However, it does not mean forward CE does not have the opposite drive force because forward CE is minimized if and only if $Q_\theta(x) = P(x)$. However, as shown in Figure 3, the loss magnitude is greatly smaller than the loss magnitude we get from reverse CE.

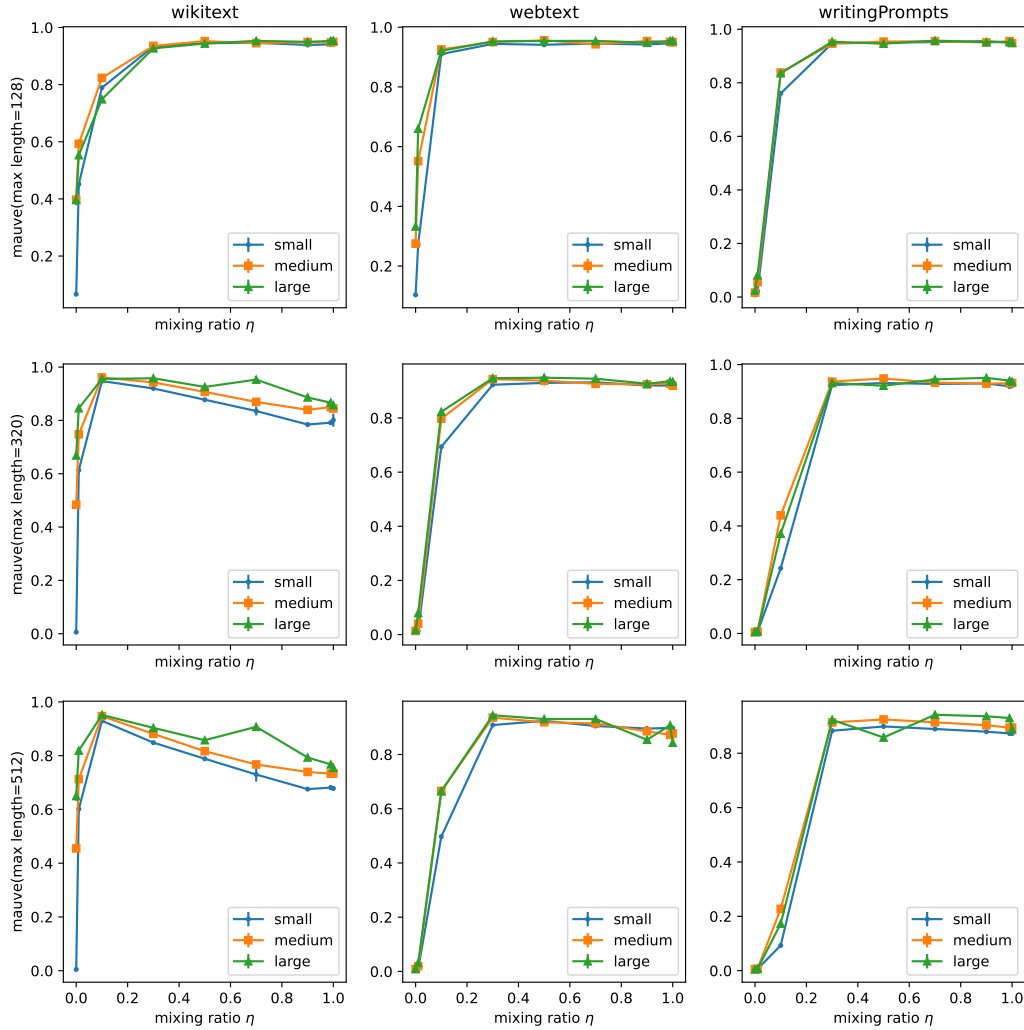


Figure 4: The mauve scores obtained by MIXCE-finetuned GPT-2 models on development sets with different max generation lengths and different η . Note that when $\eta = 1$, MIXCE is equivalent to MLE. The x-axis is the mixing ratio η , and the y-axis refers to mauve scores with different max generation lengths. The 3 lines in each subplot show the results of GPT-2 models in different sizes. The 3 subplots in each row are the results of 3 datasets respectively. Unbiased sampling is used as the decoding method. Each dot is the average of 3 runs of sampling and the error bar shows the standard deviation of 3 runs.

D Additional Results

D.1 Additional synthetic experiments

Table 6 shows the results of additional synthetic experiments besides Table 1 in the main paper. Here, the goal transition matrix \mathbf{M} is randomly initialized with 10% and 90% zero probabilities.

As the magnitudes of both avg. js and avg. Os are fairly small, we examine the 95% confidence intervals under one synthetic experimental setting – initializing the transition matrix \mathbf{M} by the bigram occurrence in the WebText data and setting vocabulary size as 1000. Table 7 contains the results. We can see that 95% confidence intervals are small enough to maintain the trend of the results.

D.2 Varying training data sizes

Table 5 shows the results of using different training data sizes in the real-data setting.

D.3 Varying η and max generation length

Figure 4 illustrates the curves of mauve scores on the development sets.

D.4 Controlled Mauve and Coherence

We find that the actual length of the text is a confounding factor of mauve computation. For example, when we compute mauve between a set of texts and the same set with an extra new line token after each text (or the same set with the last k tokens being truncated), the score will be lower

Vocab	Objective	Random (10%)		Random (90%)	
		avg. js	avg. 0s	avg. js	avg. 0s
	Gold	0.0	0.0	0.0	0.0
20	For. KL	3.65e-4	1.80e-4	7.56e-4	9.10e-5
	Rev. KL	3.41e-3	5.56e-6	1.87e-1	1.54e-6
	Mix KLS	3.11e-4	7.11e-5	4.01e-4	2.67e-5
	JS	5.68e-3	1.17e-5	2.14e-1	5.24e-4
	MIXCE*	4.92e-4	1.59e-4	4.87e-4	2.95e-5
	MIXCE	3.31e-4	1.57e-4	7.08e-4	8.49e-5
50	For. KL	6.01e-3	1.21e-3	2.18e-3	8.90e-5
	Rev. KL	2.03e-2	2.01e-5	4.11e-1	4.55e-6
	Mix KLS	4.65e-3	1.29e-4	1.54e-3	3.41e-5
	JS	1.03e-1	9.03e-5	4.24e-1	1.25e-5
	MIXCE*	5.20e-3	6.84e-4	1.48e-3	2.70e-5
	MIXCE	5.96e-3	1.20e-3	2.03e-3	7.70e-5
100	For. KL	3.34e-2	2.49e-3	6.98e-3	1.49e-4
	Rev. KL	2.30e-1	1.79e-3	5.30e-1	6.25e-6
	Mix KLS	2.98e-2	4.66e-4	5.04e-3	6.34e-5
	JS	2.38e-1	1.06e-3	5.18e-1	1.32e-3
	MIXCE*	3.10e-2	1.73e-3	5.12e-3	6.00e-5
	MIXCE	3.29e-2	2.44e-3	7.01e-3	1.50e-5
500	For. KL	1.56e-1	1.57e-3	1.93e-1	8.45e-4
	Rev. KL	2.94e-1	9.91e-4	6.49e-1	2.33e-6
	Mix KLS	1.55e-1	1.45e-3	1.70e-1	6.83e-4
	JS	2.95e-1	9.78e-4	5.75e-1	1.35e-3
	MIXCE*	1.55e-1	1.45e-3	1.69e-1	6.71e-4
	MIXCE	1.55e-1	1.56e-3	1.88e-1	6.28e-4
1000	For. KL	1.83e-1	8.95e-4	3.65e-1	7.31e-4
	Rev. KL	2.86e-1	6.12e-4	6.68e-1	3.88e-6
	Mix KLS	1.80e-1	8.64e-4	3.50e-1	6.86e-4
	JS	2.88e-1	6.11e-4	5.80e-1	7.73e-4
	MIXCE*	1.83e-1	8.64e-4	3.50e-1	6.84e-4
	MIXCE	1.83e-1	8.92e-4	3.48e-1	6.71e-4

Table 6: The results of the other two synthetic experiments. Random (10%) and Random (90%) both use random initialization for \mathbf{M} , and 10% and 90% probabilities in \mathbf{M} are 0 respectively. Gold refers to the results when $\mathbf{M}'=\mathbf{M}$. Each value is a 5-seed average.

than 0.01. Though you may think truncating all texts to the same length can resolve this problem, we find that the *incompleteness* caused by truncation can also be a confounding factor. For instance, keeping human texts intact, we truncate texts generated by two systems by their shorter lengths (i.e., for each example, we truncate text1 and text2 by $\min_length(\text{text1}, \text{text2})$). Then, the system whose texts get truncated less will get a greatly larger mauve score than the other system. Therefore, to eliminate the influence of these two confounding factors, we propose a *controlled mauve* computation approach. Concretely, for the set of human texts \mathbf{T}_h and the set of model-generated texts \mathbf{T}_m , we randomly sample 10K L -length text fragments from each of these two sets. L is the number of tokens in each text fragment. After that, we compute the mauve between these two sets of 10K text fragments. We denote this controlled mauve as

Vocab	Objective	WebText	
		avg. js	avg. 0s
1000	For. KL	8.10e-2 \pm 2.45e-4	1.50e-4 \pm 5.58e-7
	MIXCE*	7.44e-2 \pm 2.46e-4	1.14e-4 \pm 6.15e-7
	MIXCE	7.94e-2 \pm 2.15e-4	1.42e-4 \pm 5.05e-7

Table 7: Synthetic experimental results with 95% confidence intervals. WebText means initializing \mathbf{M} by the bigram occurrence in the WebText data.

c-mauve $_L$.

$$\mathbf{F}_{h,L} = \{f_{h,L}^i\}_{i=1}^{10K}, f_{h,L}^i \sim \mathbf{T}_h$$

$$\mathbf{F}_{m,L} = \{f_{m,L}^i\}_{i=1}^{10K}, f_{m,L}^i \sim \mathbf{T}_m$$

$$\text{c-mauve}_L = \text{mauve}(\mathbf{F}_{h,L}, \mathbf{F}_{m,L})$$

To sample each fragment, we first randomly sample a text t^i from the set, and then randomly select a start token s (as long as there are more than L tokens from s to the end of t^i), then the fragment is $t^i[s : s + L]$. Finally, Table 8 shows the results. We set $L = 100, 200$, and 300 , except that we could not get 10K 200-token fragments from WikiText because its texts are shorter.

The Coherence score (Su et al., 2022) computes the cosine similarity between the prompt and the continuation. We suspect that the length of the continuation may affect the score. Therefore, following the same idea of controlled mauve, we also sample 10K fragments of the same length from the set of texts for evaluation and compute coherence on the fragments. And for each fragment, we take the first 50 tokens as the prompt and the rest as the continuation. Table 9 shows the results. As you can observe, under this controlled setting, MIXCE-finetuned models generally achieve better coherence over MLE-finetuned models.

D.5 Text length of model generations

Though by default we set the max generation length as 512, the actual text length can vary as the EOS token can be sampled at any time step. Therefore, we list the average text length of the human text and GPT2-large generations in Table 10. We observe that model generations are always shorter than human text. Compared to MLE, our MIXCE-finetuned model produces shorter text on WikiText while producing longer text on the other two datasets. We suspect that the shorter length of MIXCE on WikiText is due to the small mixing ratio (0.1) chosen based on mauve (see Table 12). However, we do not think shorter text length leaves

Model Size	Objective	WikiText	WebText			WritingPrompts		
		c-mauve ₁₀₀	c-mauve ₁₀₀	c-mauve ₂₀₀	c-mauve ₃₀₀	c-mauve ₁₀₀	c-mauve ₂₀₀	c-mauve ₃₀₀
	Human	0.97	0.96	0.96	0.96	0.96	0.96	0.96
Small	MLE	0.92	0.93	0.92	0.90	0.94	0.94	0.92
	MIXCE	0.92	0.94	0.94	0.93	0.95	0.94	0.94
medium	MLE	0.94	0.93	0.91	0.90	0.94	0.94	0.93
	MIXCE	0.93	0.95	0.94	0.94	0.95	0.94	0.94
Large	MLE	0.93	0.93	0.93	0.91	0.94	0.94	0.93
	MIXCE	0.93	0.94	0.94	0.93	0.95	0.95	0.95

Table 8: Controlled mauve results. Unbiased sampling is used as the decoding method, i.e., using the same model generations as Table 2. Human scores are not 1 because sampling 10K fragments twice result in two different sets. Each number is a 3-run average.

Model Size	Objective	WikiText	WebText			WritingPrompts		
		c-coh ₁₀₀	c-coh ₁₀₀	c-coh ₂₀₀	c-coh ₃₀₀	c-coh ₁₀₀	c-coh ₂₀₀	c-coh ₃₀₀
	Human	0.570	0.521	0.583	0.600	0.412	0.470	0.481
Small	MLE	0.504	0.444	0.515	0.535	0.350	0.412	0.429
	MIXCE	0.508	0.458	0.524	0.545	0.363	0.422	0.437
Medium	MLE	0.518	0.446	0.515	0.535	0.355	0.415	0.432
	MIXCE	0.527	0.484	0.546	0.565	0.362	0.425	0.437
Large	MLE	0.521	0.449	0.515	0.536	0.372	0.431	0.447
	MIXCE	0.522	0.469	0.531	0.569	0.369	0.434	0.450

Table 9: Controlled coherence results. Unbiased sampling is used as the decoding method, i.e., using the same model generations as Table 2. Each number is a 3-run average.

Model Size	Objective	WikiText	WebText	WritingPrompts
		avg. len	avg. len	avg. len
	Human	124.5	304.5	332.5
Large	MLE	114.8	284.2	325.8
	MIXCE	89.0	298.9	326.4

Table 10: Unbiased sampling text lengths of models finetuned by MLE or MIXCE on three datasets. Length is computed by simply splitting text by whitespaces.

to better mauve, as shown by the other two datasets and discussed in D.4.

E Best η s

Table 11 has the best η s for synthetic experiments. Table 12 contains the best η s selected for GPT-2 experiments.

F Human Evaluation Details

We conduct A/B testing (or pairwise comparison) to compare generations from two models. As shown in Figure 5, in each job, we give the evaluator two text paragraphs (in random order) that share the same beginning part (the prompt) but have different continuations. Then, they need to choose which one they think is better (or non-distinguishable). To avoid random selections, they are also asked to provide a justification for their

choice. We find this justification not only gives us additional explanations of their choices but also helps us easily identify bad workers, because bad workers tend to use one single justification or several repeated justifications.

We instruct them by defining a good text paragraph as being:

- **Fluent:** Should have no obviously ungrammatical sentences, missing components, etc. that make the text difficult to read.
- **Coherent:** Should stay on topic with the prompt and build from sentence to sentence to a coherent body of information.
- **Informative:** Should have diverse and interesting content.
- **Sensical:** Should generally make sense.

Since short text has little information and long text is difficult to read, we only use paragraphs with 5 to 8 sentences for evaluation. If a paragraph has more than 8 sentences, we truncate it to 8 sentences. And we remove paragraphs with less than 400 or more than 2000 characters. Besides, to eliminate the influence of length difference, we do not select examples whose length difference between two

Model section is based on avg. js					
Vocab	Objective	Random (50%)	WebText	Random (10%)	Random (90%)
		best η	best η	best η	best η
20	Mix KLS	0.99	0.9	0.99	0.99
	JS	0.9	0.9	0.9	0.9
	MIXCE*	0.99	0.99	0.99	0.99
	MIXCE	0.9	0.99	0.99	0.99
50	Mix KLS	0.99	0.99	0.9	0.99
	JS	0.01	0.99	0.9	0.9
	MIXCE*	0.99	0.99	0.99	0.99
	MIXCE	0.99	0.99	0.99	0.9
100	Mix KLS	0.9	0.99	0.9	0.99
	JS	0.01	0.99	0.99	0.01
	MIXCE*	0.99	0.99	0.99	0.99
	MIXCE	0.5	0.9	0.5	0.99
500	Mix KLS	0.9	0.99	0.99	0.99
	JS	0.99	0.99	0.99	0.99
	MIXCE*	0.99	0.99	0.99	0.99
	MIXCE	0.1	0.5	0.1	0.1
1000	Mix KLS	0.99	0.99	0.99	0.99
	JS	0.99	0.99	0.99	0.99
	MIXCE*	0.99	0.99	0.99	0.99
	MIXCE	0.1	0.5	0.1	0.1

Table 11: The selected best η of synthetic experiments reported in Table 1 and Table 6. The model section is based on avg. js.

Model section is based on mauve (max length=512) on dev set				
Model Size	Objective	WikiText	WebText	WritingPrompts
		best η	best η	best η
Small	MIXCE	0.1	0.5	0.5
Medium	MIXCE	0.1	0.3	0.5
Large	MIXCE	0.1	0.3	0.7

Table 12: The selected best η of GPT-2 experiments reported in Table 2. The model section is based on mauve (max length=512) on the dev set.

paragraphs is more than 1 sentence or more than 200 characters.

We conduct this evaluation on Amazon Mechanical Turk. We only allow workers, who are located in the US, have a Masters Qualification,⁷ have an approval rate larger than 97%, and have more than 10000 HITs approved, to do our tasks. In addition, we first ran a testing batch, then manually checked the results, and selected 44 qualified workers to continue doing the rest of our tasks.

For each of the 3 datasets, we sampled 105 examples and collected 3 responses per example. In total, we received 945 human evaluations. We pay workers \$1 per response, and it takes around 5 minutes to finish one response, i.e., the hourly rate is around \$12.

⁷<https://www.mturk.com/worker/help>

Table 13 shows that inter-annotator agreements. Figure 6-11 are 6 randomly sampled examples from human evaluation results, 2 examples per dataset.

G Reproducibility

In our GPT-2 experiments, we use English text data from 3 domains: (1) WikiText (Merity et al., 2017): text from Wikipedia, and we use wikitext-103-raw-v1 from Hugging Face.⁸ Its license is Creative Commons Attribution-ShareAlike License (CC BY-SA 4.0). (2) WebText (Radford et al., 2019): text from the Web. It was used for pretraining GPT-2. The full WebText is not available but they released a subset on Github⁹. The GitHub repository con-

⁸<https://huggingface.co/datasets/wikitext>

⁹<https://github.com/openai/gpt-2-output-dataset>

Dataset	all agree	2 agree	no agreement
WikiText	24%	59%	17%
WebText	24%	52%	24%
WritingPrompts	20%	70%	10%

Table 13: Inter-annotator agreement. The numbers are the portions of examples that have a 3-annotator agreement (all agree), a 2-annotator agreement (2 agree), or no agreement. E.g., 24% of examples used in human evaluation for WikiText have an agreement among 3 annotators.

tains an MIT license, and they did not specify the license of the data. But they indicated in the readme: “We look forward to the research produced using this data!” (3) WritingPrompts (Fan et al., 2018)¹⁰: text from the writing prompts forum of Reddit. Its GitHub repository also contains an MIT license without specification of the data license. However, WritingPrompts has been used by many other research works, e.g., Pillutla et al. (2021). We use their official dev and test sets as much as possible. If they have fewer than 5K examples, we sample from their official training set to make up the rest.

All of our experiments were conducted on NVIDIA Tesla V100 32G GPUs. We use a single GPU to run each experiment and change the batch size to fit models of different sizes. When fine-tuning GPT-2 small using a single GPU with MLE or MIXCE, it took less than 1 hour to finish 5 epochs on 50K WikiText training data and took less than 2 hours to finish 5 epochs on 50K WebText or WritingPrompts training data.

We implemented our GPT-2 based models based on the GPT-2 modeling code from Hugging Face Transformers¹¹. For training and evaluation, we modified the example script of causal language model training¹². We used the default optimizer, learning rate, scheduler, etc. in that script. But we set the maximum training epochs as 5 and changed the batch size and gradient accumulation steps based on the model size to fit it in one 32G-memory GPU.

¹⁰<https://github.com/facebookresearch/fairseq/tree/main/examples/stories>

¹¹https://github.com/huggingface/transformers/blob/main/src/transformers/models/gpt2/modeling_gpt2.py

¹²https://github.com/huggingface/transformers/blob/main/examples/pytorch/language-modeling/run_clm_no_trainer.py

Instructions (Please read carefully to ensure that your work gets approved as quickly as possible!)

Welcome!

We need your help to compare the quality of two paragraphs of text.

For each assignment, you will be given **two paragraphs** of the same topic. They share the same **beginning part (or the prompt)**. Your task is to judge **which one is better and justify your choice**. A good text paragraph is:

1. **Fluent: Should have no obviously ungrammatical sentences, missing components, etc. that make the text difficult to read.**
2. **Coherent: Should stay on topic with the prompt and build from sentence to sentence to a coherent body of information.**
3. **Informative: Should have diverse and interesting content.**
4. **Sensical: Should generally make sense.**

Please note that:

1. It is NOT necessary that one paragraph is better than the other. You should choose "non-distinguishable" if you think they are both very poor/good.
2. Paragraphs may contain facts that you are not familiar with. You do NOT need to verify those facts because **factuality is not a concern in this evaluation**.
3. Paragraphs are excerpts from longer text, so the ending part may not be complete. But **incompleteness should NOT affect your judgment**.
4. When justifying your choice, please be **as concrete as possible and point to problems** in the paragraphs.

Paragraph-1:

[WP] A new drug let's you live a lifetime in one dream 30 minutes for an entire life? Wonderful, just pop the pill before you go to bed and when you reach peak REM, you'll enter a whole new life, one you never could have taken apart. But insert those 3 or 4 pills and the life begins to unravel again. So i took 3 pills that i got from carnival a few weeks ago. It was cheap and stuff boring but i thought it might help me sleep freshman year of high school. Ill hangrily pill suddenly and go back to my dream. Unfortunately the next thing i know my grandfather has died. I start walking up my stairs and checking every room for a note, and some kind of family photograph. One by one I reach my room, and find nothing but faded pictures of my grandparents, then a wrong number on the phone app

Paragraph-2:

[WP] A new drug let's you live a lifetime in one dream 30 minutes for an entire life? Wonderful, just pop the pill before you go to bed and when you reach peak REM, you'll enter a whole new life, one you never knew existed until just now. The bright light blipped and my eyes dropped open as the room automatically blacked out. A weak light throbbbed in my right temple, and another flowed back into the darkness. Winds rolled through the lower levels of the dormitory's cluster of rooms, and fresh snow drifted across the windowless, cracked rafters. I heard a sudden flare of static, and then silence. I strode onto the roof and jumped off like a springsteen with a single, softly fading tune. My hands began to tremble, and my feet began to throb. I scrambled to the breakfast nook, sat on the roof bench, and grabbed the nearest lamp.

Which paragraph do you think is better?

Paragraph-1

Paragraph-2

Non-distinguishable

Please justify your preference:

I think they're equally 'informative,' but Paragraph 2 wins on fluency, coherence, and comes far closer to making sense. Paragraph 1 is basically a heaping helping of word salad.

Figure 5: Human evaluation interface and a random example from our collected human annotations.

Paragraph1 (MLE):

Within minutes of issuing the signal and turning his flagship HMS Queen Charlotte, Howe's plan began to falter. Many of the British captains had either misunderstood or ignored the signal and were hanging back in the original line. Other ships were still struggling with damage to their ships and other equipment caused by the storm, so that Australia's war effort was threatened. In response to the storm-ravaged Australian distress call on 12 March, Howe ordered his ship HMS Sun Babies as flagship of rowing party V HMNZS Platypus. He assigned elevensacks from the ship to the crew for a rescue, and placed much of the more aged of his crew and stockmen on sick leave on the advice of Admiral Rear-Admiral John Forbes.

Paragraph2 (MIXCE):

Within minutes of issuing the signal and turning his flagship HMS Queen Charlotte, Howe's plan began to falter. Many of the British captains had either misunderstood or ignored the signal and were hanging back in the original line. Other ships were still struggling with damage to their yards and were standing still. Another officer, Lieutenant Colonel Strother, privately stated that the British ship captains had "obsessed [ed] the whole thing over with the consciousness of the impossibility of victory". Commander Howe was eager to proclaim victory, but it was not until the Chief of the General Staff, Sir Charles Ogilvy, finally appeared on the scene that he decided to make a decision.

Choice: paragraph2

Justification: Paragraph 1 has details that do not flow with each other as if Howe was involved with two ships at the same time. The beginning of the paragraph mentions Howe's flagship HMS Queen Charlotte and later on his flagship HMS Sun Babies. It is not clear about the role of Australians whether there were Australians in the crew or they were helping Australian ships. In paragraph 2, there is no mention of another nation other than just British. It has details of different ranks of commands. Moreover the paragraph focus on failure and uncertainty that flow to the rest of the paragraph.

Choice: paragraph2

Justification: Paragraph 2 seems to make more sense and is more coherent than Paragraph 1. #2 has a narrative flow that can be read from start to finish, giving the reader a sense of what is happening. The quote is the weak point, but as a quote, can be taken as it is given here. On the other hand, Paragraph 1 can also be read from start to finish, but makes less sense, and seems less coherent.

Choice: paragraph2

Justification: Paragraph 2 is more coherent, and it wins hands down in terms of making sense. Paragraph 1 has Howe sending men on leave in the middle of a battle and also has him assigning "elevensacks" to the crew, which isn't even a real word.

Figure 6: WikiText example 1.

Paragraph1 (MLE):

In imitation of the Greeks, Nero built a number of gymnasiums and theatres. Enormous gladiatorial shows were also held. Nero also established the quinquennial Neronia. The festival included games, poetry, and theater. For the recesses of the Calendrical Mysterium, the city quinquenored, with the city many times superseded.[ndx] Nero also Festina-Illuminatae in honour of the doomed Emperor Caesar and his son Severus, the founders of the Flavian dynasty, who were slain in that period. In Britain, Nero erected the Attic theatre.

Paragraph2 (MIXCE):

In imitation of the Greeks, Nero built a number of gymnasiums and theatres. Enormous gladiatorial shows were also held. Nero also established the quinquennial Neronia. The festival included games, poetry, and theater. In order to combat the influence of the Venetian party, the local aristocracy discouraged the shooting of arrows at the official spectators. Nero's penchant for shooting arrows became a local insult.

Choice: paragraph2

Justification: The first one is hard to understand. What does " the city quinquenored, with the city many times superseded.[ndx]" mean.? It does not make sense. The second one is easy to understand.

Choice: paragraph2

Justification: I had to read Paragraph 1 a few times before it made sense, but Paragraph 2 flowed nicely and logically without effort.

Choice: paragraph2

Justification: 2 takes a bit of a stretch but it can at least follow some plausible logical order. 1 just abruptly starts talking about geopolitics which doesn't make sense in that context.

Figure 7: WikiText example 2.

Paragraph1 (MLE):

Two years ago, Sandrine Jourdan walked up a friend's Sunshine Coast driveway and simply disappeared. The Caboolture woman has not accessed her bank accounts, used her phone or wished any of her three children a happy birthday since then. She insists she hasn't moved house, childcare is still open and she can speak to her family now. "I owe them no money." Jourdan said. Jourdan was last heard from around February this year and her family does not know where she is. One of her family members told me Jourdan was hers but to leave Jourdan alone would be like asking the sick to go to the doctor. However, the woman, who is in her 50s, said Jourdan was extremely important to her family and they didn't want to send her any money. They saw her daily and had almost no tensions of any kind to handle

Paragraph2 (MIXCE):

Two years ago, Sandrine Jourdan walked up a friend's Sunshine Coast driveway and simply disappeared. The Caboolture woman has not accessed her bank accounts, used her phone or wished any of her three children a happy birthday since then. But she has no one to blame but herself. The welfare office at Taree hostel was "terminating her accommodation on a whim" last Spring, she says. Discarded money The hostel is not Charter and cannot employ staff. Desperate for a fresh start, Sandrine tried to access her bank accounts and managed to access her daughter's allowance, the bequests of her parents, a benefit card and money in savings. She has not seen her family since. To compound her plight, she has lost her job on the condominium project near the town of Machalung where she was staying

Choice: non-distinguishable

Justification: Both are bad and neither follow the prompt. Paragraph 1 isn't coherent. She is on good terms with her family, then not, then back again. Paragraph 2 also has contradictions, and the sentence "Discarded money The hostel is not Charter and cannot employ staff. " seems randomly inserted.

Choice: non-distinguishable

Justification: Neither follows the prompt in the least. It is about a woman who disappears, and both go on to have quotes and information about the missing woman as if she were still interacting with others. There are incomplete sentences in 2 which are not present in 1, but I still can't say 1 is better because it's so off prompt.

Choice: paragraph1

Justification: Paragraph 2 has some punctuation and grammar errors, for example "Discarded money The hostel is not Charter and cannot employ staff."

Figure 8: WebText example 1.

Paragraph1 (MLE):

Another Phantom is leaving the building. The musical, which is the longest running show in Broadway history, announced Monday that the Phantom, James Barbour, will leave the production on Dec. 23, 2017, after a close to three-year run. A premature announcement about the closure came shortly after the show was first announced at the 2016 Tony Awards. A representative for JBL had told The Hollywood Reporter's Hayden Dingman last year that the Broadway appearance would likely wrap but that there were still plans to continue performing on the stage. The event marked the end of the run of 'The Phantom' which reached a two-decade high in the Broadway season. The show had run a whopping 129 shows on Broadway since its inception in 1995. After 11 seasons, Ice Cube's 'Once', which had been announced 14 months prior, was a box office success

Paragraph2 (MIXCE):

Another Phantom is leaving the building. The musical, which is the longest running show in Broadway history, announced Monday that the Phantom, James Barbour, will leave the production on Dec. 23, 2017, after a close to three-year run. "Moving on to a new chapter with a new partner is one of the hardest things we've ever done as a show," said producer Scott Ploog. "Joining forces with our fearless leader James was a huge statement to us." After singing and dancing its way through the Fort Worth Convention Center in 2011, the show was working on a script for its sixth season. The Phantom surprised the show's production staff with a call from James, who was the project's conductor for the years it was running. "After hearing the love he had for the show, I knew whether I wanted to take a job as indispensable leader of the cast or what, I had to decide," said Expedition leader Cary Joji. "He was just engaging, embracing and completely out of his ways, but he was relentless

Choice: paragraph2

Justification: Paragraph 2 feels more personal with more quotes from the people that worked on the show.

Choice: paragraph2

Justification: Both do well on fluency. For coherence, I don't think either one forms a totally coherent body of information, but Paragraph 2 sticks to the topic a touch better. I think they're roughly equal in terms of diverse and interesting content. As usual, making sense is the metric each is worst at. I believe they are indistinguishably terrible in that regard. So Paragraph 2 wins, just barely, via being slightly more coherent.

Choice: paragraph2

Justification: The first paragraph seems to be a mix of unrelated items, at first discussing a performer leaving the cast of Phantom and then talking about the end of the production and then concluding with "After 11 seasons, Ice Cube's 'Once', which had been announced 14 months prior, was a box office success", which seems unrelated to Phantom. The second paragraph seemed little off also, but had somewhat more coherence. The first paragraph also lacked a couple comma, by rule, but I didn't factor that in very much.

Figure 9: WebText example 2.

Paragraph1 (MLE):

[WP] There is no such thing as “dating ” anymore. Just like in Cinderella, As soon as you touch the hand of“ the one” you have an insatiable urge to kiss. Sam sighed. Despite her best efforts to hide it, the bruises and cuts beneath her blouse were pretty apparent from her sitting down. Her heart pierced so many ribs that buckles broke, and chips torn the flesh of her ribs, arm and leg. And they weren’t just sitting there - it was pouring out down the side of the castle. When they arrived back at the villa after a short- but exhausting trip from the train station, they were determined to hug and kiss, just as well as they could in dark alleys. Unfortunately, the darkness blinded them from view, so the light was flowing with the pouring rain, sending sparks from their inexpensive outfits

Paragraph2 (MIXCE):

[WP] There is no such thing as “dating ” anymore. Just like in Cinderella, As soon as you touch the hand of“ the one” you have an insatiable urge to kiss. Sam sighed. Despite her best efforts to reverse the trend, her hand seemed to instinctively take the hand of the lone participant. He smiled, as Sam closed her eyes. "Today’s the day you know, Yattar," he insisted confidentially. "Today’s the day you know the thrill of kissing!" Yattar nodded, with conviction. "Anyone the purveyors of-" "Remind me to spice everything up," she dismissed, this time making eye contact with the man her mouth had caught them

Choice: paragraph2

Justification: Assuming that "Yattar" is a pet name/different name for "Sam", 2 is much more cohesive and fluent than 1. It is easy to follow the story and the story makes more sense than 1. The "pouring out down the side of the castle" in 1 makes no sense, and other content is pretty incoherent as well.

Choice: paragraph1

Justification: Paragraph 1 is a more concrete story of love, while 2 feels more disjointed.

Choice: paragraph2

Justification: Paragraph 2 is more cohesive, sticking to the concept introduced in the first sentence and describing the initial meeting of a couple. Para 1 rather jarringly adds a description of injuries and a castle setting.

Figure 10: WritingPrompts example 1.

Paragraph1 (MLE):

[CC] I wrote these stories recently here on WP, and would like to share them with my friends; however, I feel like they could be tuned up a bit. Any thoughts on how to make improvements is greatly appreciated. First off, there are a lot of inconsistencies in their sentences, it’s good to have a prompt where you can color all the immersion steps individually so you can judge meaningful sequences when looking at them in context. 2nd, while I did my best to adhere to game mechanics in the story, they were probably made largely to draw parallels between the reader and protagonist. Seth Vargo Hughes is an unfortunately short person himself (7 weeks a year usa is somewhere between 8 to 10 years long. I amn’t entirely sure what his killer skills are now, or about what he can do), so he may get caught with some pretty nasty tools and weapons, and at the end of the story, he’s not gonna give the guy who rescued him the satisfaction of just putting his fist through his friend’s chest to make his statement true. But that happens. 2nd, 2 other things that PWP failed to properly convert into told story

Paragraph2 (MIXCE):

[CC] I wrote these stories recently here on WP, and would like to share them with my friends; however, I feel like they could be tuned up a bit. Any thoughts on how to make improvements is greatly appreciated. First off, there’s a ton of image and word overlap on this story, which leads to a lot of writing and visibility issues. There definitely should have been done more body alignment, with the characters looking like more anthropomorphic animal animals more often. Additionally, this idea if poorly worded, as the phrase "spider" was spelled multiple times throughout the story and is such a terribly named word. It should be the appropriate sound, and become "f*cking spider"! It also doesn’t help that you are writing in third person, which means that you could have brought in third person pronouns, such as "She" or "I" (though you may want to change some of the wording, such as "She"). The last thing I would change about this story is the different lengthy, regular sentences that end so often. One of the most common techniques used for lengthening stories is italicizing, wherein you substitute a word with an italicized sound.

Choice: paragraph2

Justification: Paragraph 1 has several grammatical flaws. It also begins to ramble in places. Paragraph 2 includes pertinent details and completes the main subject.

Choice: paragraph1

Justification: Both are truly terrible on every metric. Paragraph 2 is so problematic on fluency that Paragraph 1 wins despite using the non-word "amní." As far as coherence and information goes, they are equally dreadful, and neither makes any sense whatsoever.

Choice: paragraph2

Justification: 1 deviates halfway through the prompt and starts talking about a different subject matter almost seamlessly. It almost makes sense if you don’t read it very closely.

Figure 11: WritingPrompts example 2.

ACL 2023 Responsible NLP Checklist

A For every submission:

- A1. Did you describe the limitations of your work?
The limitation section on page 9
- A2. Did you discuss any potential risks of your work?
The ethical consideration section on page 9
- A3. Do the abstract and introduction summarize the paper’s main claims?
abstract and Section 1
- A4. Have you used AI writing assistants when working on this paper?
Left blank.

B Did you use or create scientific artifacts?

Section 4.2, Appendix F, and Appendix G

- B1. Did you cite the creators of artifacts you used?
Section 4.2, Appendix F, and Appendix G
- B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Appendix G
- B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Appendix G
- B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
We used public datasets and our data collection does not introduce identifications or offensive content.
- B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section 4.2, Appendix F, and Appendix G
- B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 4.2, Appendix F, and Appendix G

C Did you run computational experiments?

Section 4

- C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 4 and Appendix G

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

- ✓ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Appendix G
 - ✓ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 4 and Appendix D
 - ✓ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Appendix G
- D** ✓ **Did you use human annotators (e.g., crowdworkers) or research with human participants?**
Section 4.2 and Appendix F
- ✓ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Section 4.2 and Appendix F
 - ✓ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Section 4.2 and Appendix F
 - ✓ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Appendix F
 - ✓ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Appendix F, by a Bloomberg legal team
 - ✓ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Appendix F