Boosting coherence of language models

Anonymous ACL submission

Abstract

Naturality of long-term information structure – coherence – remains a challenge in language generation. Large language models have insufficiently learned such structure, as their long-form generations differ from natural text in measures of coherence. To alleviate this divergence, we propose *coherence boosting*, an inference procedure that increases the effect of distant context on next-token prediction. We show the benefits of coherence boosting with pretrained models by distributional analyses of generated ordinary text and dialog responses. We also find that coherence boosting with state-of-the-art models for various zero-shot NLP tasks yields performance gains with no additional training.

1 Introduction

Language models are commonly evaluated for their ability to generate, rank, or classify coherent spans of text. However, LMs learn from data that may violate pragmatic norms. In addition, autoregressive LMs are typically fit to a multi-objective problem: simultaneously maximizing token likelihoods conditioned on many lengths of truncated context (§2.1). Yet, at generation or scoring time, distributions are conditioned on the entire prompt or previously generated string, which is known to be coherent or even guaranteed to influence the output.

We show that large LMs, such as GPT-2 and -3 (Radford et al., 2019; Brown et al., 2020) exhibit failures in long-range coherence (Fig. 1). Samples from these LMs have an unnaturally low density of words that require many tokens of preceding context to predict (§4.1), and the scores that the models give to completions of prompts indicate that they are oversensitive to recent context (§5). To remedy these failures, we propose **coherence boosting**, a simple inference-time procedure that increases the effect of distant words on predicted token distributions. A pretrained model is viewed as an *ensemble* of experts that produce token distributions conditioned on varying lengths of context. These experts are log-linearly mixed to form a predictor that is superior to the base model (§2).

Coherence boosting greatly improves prediction of words that depend on a long context, as evidenced by state-of-the-art results on tasks specially meant to assess models' attention to distant words (§3). In generation of generic text and dialog responses, we show that coherence boosting brings the frequency of occurrence of such words close to that seen in natural text (§4). Beyond generation, we study diverse multiple-choice tasks (§5), in which examples are known to be highly coherent. Coherence boosting does not modify the base model and depends on a single parameter than can be estimated in one pass through a validation set, yet is an competitive adaptation algorithm.

1.1 Background and related work

Balance between satisfaction of short-range statistical constraints and maintenance of long-range structure was a central question of language generation long before neural language modeling: n-gram models and early neural language models commonly used 'backing-off' schemes that adaptively assign interpolation weights to predictors with different context lengths (Chen and Goodman, 1996; Bengio et al., 2003). Neural language modeling brought a need for recurrent units with better numerical properties for propagating information over long distances (Hochreiter and Schmidhuber, 1997; Cho et al., 2014) and eventually saw the reintroduction of alignment variables (Brown et al., 1993) into generation in the form of attention (Bahdanau et al., 2015; Vaswani et al., 2017). Attention is at the core of Transformer LMs, including GPT.

Language models are being trained on and adapted to ever-longer input sequences (Beltagy et al., 2020; Zaheer et al., 2020; Roy et al., 2021; Press et al., 2021), but they remain undersensi-

A: I'm Natasha. I study neural language models and dialog systems. Are you an AI researcher too?
B: No, though I do like chatting with bots and laughing at their mistakes. But what was your name again?
A: Oh, you forgot already? My name is $\boxed{\text{w}}$

$p_{\text{full}} = f(\text{w} \mid \text{full})$ *1.* Alex (1.9%) *2.* **Natasha** (1.7%) *3.* also (1.5%)
$p_{\text{short}} = f(\text{w} \mid \text{short})$ *1.* : (3.4%) *2.* the (1.9%) *3.* in (1.2%) …*3358.* **Natasha** (0.0042%)
$p_{\text{full}}^{1.5} p_{\text{short}}^{-0.5}$ *1.* **Natasha** (20.5%) *2.* Alex (2.2%) *3.* Nat (2.1%)

Ballad metre is "less regular and more conversational" than common $\boxed{\text{w}}$

$p_{\text{full}} = f(\text{w} \mid \text{full})$ *1.* sense (9.0%) *2.* in (2.0%) *3.* . (1.9%) …*13.* **metre** (0.6%)
$p_{\text{short}} = f(\text{w} \mid \text{short})$ *1.* sense (7.8%) *2.* English (3.5%) *3.* . (3.2%) …*14103.* **metre** (0.00014%)
$p_{\text{full}}^{1.5} p_{\text{short}}^{-0.5}$ *1.* **metre** (16.2%) *2.* sense (4.0%) *3.* meter (2.5%)

Isley Brewing Company: Going Mintal – a minty milk chocolate $\boxed{\text{w}}$

$p_{\text{full}} = f(\text{w} \mid \text{full})$ *1.* bar (4.8%) *2.* drink (3.7%) *3.* with (3.5%) …*13.* **stout** (2.7%)
$p_{\text{short}} = f(\text{w} \mid \text{short})$ *1.* bar (6.9%) *2.* that (5.7%) *3.* , (4.4%) …*60.* **stout** (0.23%)
$p_{\text{full}}^{1.5} p_{\text{short}}^{-0.5}$ *1.* **stout** (7.4%) *2.* ale (5.6%) *3.* bar (3.1%)

Other times anxiety is not as easy to see, but can still be just as $\boxed{\text{w}}$

$p_{\text{full}} = f(\text{w} \mid \text{full})$ *1.* important (5.6%) *2.* bad (4.6%) *3.* **debilitating** (4.3%)
$p_{\text{short}} = f(\text{w} \mid \text{short})$ *1.* effective (16.2%) *2.* good (7.4%) *3.* useful (3.9%) …*294.* **debilitating** (0.035%)
$p_{\text{full}}^{1.5} p_{\text{short}}^{-0.5}$ *1.* **debilitating** (17.6%) *2.* real (6.0%) *3.* severe (5.8%)

Figure 1: Next-token probabilities given by LMs (DialoGPT and GPT-2) conditioned on a long context and on a partial context. The top words in both distributions are incorrect, but a log-linear mixture of the distributions makes the correct word most likely. Sampling from such a mixture at each generation step (*coherence boosting*) improves the quality of output text (§4). (Dialog example written by the authors; other examples from OpenWebText.)

tive to distant content or syntax (Khandelwal et al., 2018; Sun et al., 2021) and are easily fooled by recency bias in few-shot prompts (Zhao et al., 2021) or multi-turn conversations (Sankar et al., 2019).

Recent work has continued to study inference-time procedures that prevent text sampled from LMs from degenerating into nonsense. Most of these procedures, such as tempered sampling and top-k/top-p truncation (Fan et al., 2018; Holtzman et al., 2019), independently modify the output distribution at each generation step to decrease its entropy and diminish its low-likelihood tail. Holtzman et al. (2019) and Meister and Cotterell (2021) found that such local modifications increase the quality of long generated sequences; we adopt and extend their methodology in §4.1.

For dialog systems, Li et al. (2016) propose a decoding scheme that maximizes a mutual information criterion, which explicitly optimizes for dependence of generated text on prompts – a special case of coherence boosting. In multiple-choice tasks, where a model must choose one of several given completions of a prompt, Brown et al. (2020) observe that selecting the completion that maximizes $p(\text{completion}|\text{prompt})$ often favors completions having high unconditional likelihood (likelihood following an empty or dummy prompt) and, for some tasks, chooses to divide the scores of candidate answers by their unconditional likelihoods. This is also a special case of coherence boosting.

Such scoring modifications are more thoroughly studied by Zhao et al. (2021); Holtzman et al. (2021). The latter attributes the problem to 'surface form competition': there are many variants of the correct completion that together may capture a large part of probability mass, but the form of the given answer choice alone is not the most likely. However, we show that other causes are at play: surface form competition is impossible when the completion is known to be a single token and the range of choices is the whole vocabulary (§3), and it is not applicable to open-ended generation (§4).

2 Coherence boosting

In this section, f is an autoregressive LM over a vocabulary V with learnable parameters θ, taking as input a variable number of tokens (up to a maximum context length M) and producing a vector of next-token likelihoods:

$$f(w_1, \ldots, w_n; \theta) \in \Delta(V), \quad w_1, \ldots, w_n \in V,$$

where $\Delta(V)$ is the probability simplex over V. We will write the w-th component of this output vector as a conditional likelihood, $f(w \mid w_1, \ldots, w_n; \theta)$.

We denote by f_k the model evaluated on only the *last* k input tokens, ignoring earlier tokens:

$$f_k(w_1, \ldots, w_n; \theta) := f(w_{n-k+1}, \ldots, w_n; \theta).$$

Coherence boosting for next-token prediction. *Coherence boosting* for a model f selects real-valued weights $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_M)$ and produces a new language model f_α, defined by

$$\begin{aligned} & f_\alpha(w_1, \ldots, w_n; \theta) \\ & := \operatorname{softmax}\left(\sum_{k=1}^{M} \alpha_k \log f_k(w_1, \ldots, w_n; \theta)\right), \quad (1) \end{aligned}$$

where log is taken element-wise, or, equivalently,

$$f_\alpha(w|w_1, \ldots, w_n; \theta) \propto \prod_{k=1}^{M} f_k(w|w_1, \ldots, w_n; \theta)^{\alpha_k}.$$

This is a weighted product-of-experts model, where the 'experts' are copies of the base model f evaluated on different context lengths.

Because evaluating f is expensive, we use sparse weights $\boldsymbol{\alpha}$, as the expression (1) depends only on those f_k for which $\alpha_k \neq 0$. In Fig. 1 and in the experiments, we allow $\boldsymbol{\alpha}$ to have only two nonzero entries: when computing likelihoods of words following a sequence of length n, we consider weighted products of $f_{\max} := f_n$ (the full context) and an f_k with $k \leq n$ (a short context, either of fixed length or decided by prompt structure as in §4.2).

As its name suggests, coherence boosting resembles log-linear boosting for multiclass classification (Friedman et al., 2000). However, our weak classifiers are pretrained and share all of their parameters, not obtained by an iterative procedure of training on reweighted data, and we permit negative weights.

Coherence boosting for answer selection. In multiple-choice problems, a LM must choose the best answer following a context, which consists of a premise or passage followed by a shorter *premise-free context* (either a short phrase, such as "Answer:", that incites the LM to generate an answer in the right format, or a hypothesis that depends on the premise). The full context is the concatenation of the premise and the premise-free context (§C).

By the autoregressive factorization, the model f assigns conditional likelihoods to *sequences* of tokens following context. A typical model for answer selection ranks the candidate answers a_i (sequences of tokens) by $f(a_i \mid \text{full context}; \theta)$ and outputs the highest-ranked a_i. *Coherence boosting* chooses a parameter α and ranks the choices by:

$$\begin{aligned} & \log f(a_i \mid \text{full context}; \theta) + \\ & + \alpha \log f(a_i \mid \text{premise-free context}; \theta). \quad (2) \end{aligned}$$

This is a log-linear combination of two models: f evaluated with full context and with a partial context. When $\alpha = 0$, ranking by (2) is equivalent to ranking by the base model. When $\alpha = -1$, it is equivalent to dividing the base model's score by the score of each answer conditioned on the prompt (short context), and thus to maximizing pointwise mutual information between the premise and the answer conditional on the premise-free context. Unlike Brown et al. (2020); Holtzman et al. (2021), our formulation allows the premise-free context to include information specific to the example, not only a domain-specific dummy prompt.

We expect coherence boosting to correct for an oversensitivity to the premise-free context, and thus the optimal α will typically be negative (see §5).

2.1 Why should boosting models be better than full-length predictors?

Multi-objective training. As we will now see, the training of the model f simultaneously fits all of the predictors f_k, which share parameters θ. Each training iteration samples a sequence (or batch of sequences) of a chosen maximum length $M + 1$ from the data distribution $\mathcal{D}$ and minimizes the average negative log-likelihood (NLL) of *all* words following the parts of the sequence that precede them: the optimization criterion is:

$$\mathbb{E}_{w_1 \ldots w_{M+1} \sim \mathcal{D}} \frac{1}{M} \sum_{k=1}^{M} -\log f(w_{k+1}|w_1, \ldots, w_k; \theta).$$

If $\mathcal{D}$ is uniform over all length-$(M+1)$ subsequences of a training corpus, any given word is equally to likely to appear in all positions within a sampled sequence[1], and the criterion is equal to

$$\sum_{k=1}^{M} \frac{1}{M} \underbrace{\mathbb{E}\left[-\log f_k(w_{M+1}|w_1, \ldots, w_M; \theta)\right]}_{\mathcal{L}_k(\theta)}, \quad (3)$$

[1]Many authors leave unspecified the way in which training batches are formed from a corpus of input documents. Here we assume that all training documents are concatenated into one (very long) document separated by end-of-text tokens and ignore minute effects near the start and end of this document.

	GPT-2				GPT-3			
	125M	350M	760M	1.6B	2.7B	6.7B	13B	175B
$f_{\max}$	47.66	57.29	61.23	64.25	62.39	71.40	76.58	81.51
CB ($\alpha_k = \alpha_k^*$)	66.70	73.53	76.54	77.53	77.00	81.84	86.36	88.61
α_k^*	−0.6	−0.5	−0.5	−0.5	−0.3	−0.3	−0.3	−0.2
k^*	10	11	10	9	9	10	3	3

Table 1: Accuracy (%) and optimal boosting parameters on LAMBADA: $f_{\max}$ is the full-context model without boosting; CB is our model with the optimal boosting parameters (last two rows).

This is a uniform scalarization of an M-task problem: the k-th objective $\mathcal{L}_k(\theta)$ is the expected NLL of a word in the corpus following k context words.

This situation is different from that seen at *generation* time. If the text generated so far is $w_1w_2 \dots w_n$, the distribution from which the next word w_{n+1} is sampled is $f_n(w_1, \dots, w_n; \theta)$ – only the ensemble member using full context is used. However, if the string $w_1 \dots w_n w_{n+1}$ had been seen in training, f would have been trained to predict w_{n+1} given *all partial contexts*, with equal weight given to all prediction losses. Thus, f is trained to make predictions on data it never sees in evaluation, and may be prevented from optimally learning to use long context: parameters that locally optimize (3) are locally Pareto-optimal for the *set* of prediction losses $\mathcal{L}_1, \dots, \mathcal{L}_M$, but not necessarily optimal for any individual $\mathcal{L}_k$. An ensemble of the f_k ($k \le n$) may be a better predictor than f_n alone. (See §A for further analysis of when this occurs.)

Undertraining. The parameters θ are shared by the predictors f_k, and modeling power must be spread among the losses $\mathcal{L}_k(\theta)$. The short-context predictors are easier to fit, while sequences in which long context affects the prediction are rare. We expect sensitivity to long context, and precision in modeling its effect, to be especially diminished if the model is undertrained.

Distribution shift. While the training procedure causes a bias against the influence of longer contexts on generation, we see the opposite bias in downstream tasks (question answering, natural language inference, adversarial probes for common sense): Many modern NLP benchmarks try to challenge models to use long context (§3, §5).

3 Experiments: LAMBADA

The LAMBADA dataset (Paperno et al., 2016) tests LMs' understanding of long-range dependencies by measuring the prediction of the final words in passages of several sentences. The task explicitly requires reasoning over a broad context: humans can reliably guess the last word when given a whole passage, but not when given only the last sentence.

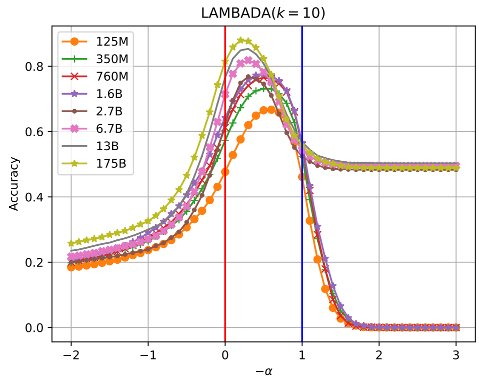

Figure 2: Model comparison on LAMBADA with $k = 10$ and varying α_k. The red line ($\alpha = 0$) is the base LM $f_{\max}$. (The different right tails of GPT-3 models are due to top-100 truncation of logits returned by the API.)

We perform experiments with the GPT family of models, closely replicating the evaluation setting of Radford et al. (2019).[2] We predict the final word as the top-ranked token under the boosted model $f_{\max} f_k^{\alpha_k}$, where $f_{\max}$ is the model taking the full available context and k, α_k are the chosen length and coefficient of the short context. To choose k and α_k, we do a grid search on the validation set and apply the best values to the testing set.

Results. Table 1 shows the accuracies and optimal parameter values k^*, α_k^*. Coherence boosting vastly reduces prediction error for all models. In particular, the boosted GPT-2 Small performs better than the original GPT-3 2.7B. The boosted GPT-3 175B achieves a new state of the art.

Other than the impressive performance gain, we highlight two observations. **(1)** The optimal α_k is always negative, indicating that the optimal mixture of models penalizes the influence of short-range context relative to long-range context. **(2)** With increasing model size, the optimal α_k and k become closer to 0. This means that bigger models capture long-range coherence better than small models, as they have less need to penalize the effect of short context. (Fig. 2 shows the accuracy curves for all models by sweeping α_k with a fixed k. The peak clearly moves to the left as model size grows.)

[2] Certain details are omitted by Radford et al. (2019). Based on https://github.com/openai/gpt-2/issues/131, we nearly match baseline accuracy by predicting the last subword token, rather than the last word.

4 Experiments: Language generation

4.1 Generic text

The experiment in this section extends that of Holtzman et al. (2019). A selection of 5000 articles from WebText (Radford et al., 2019) is taken as a reference corpus of human-written text. A language model (for us, GPT-2 Large) is prompted to generate text conditioned only on the *first sentence* of each of these articles, up to a maximum of 200 tokens, yielding 5000 machine-generated texts.

The human-written and machine-generated texts are compared by four automatic metrics: **perplexity** under the base LM, **self-BLEU-4** (Zhu et al. (2018); the mean BLEU-4 score of a generated text with respect to all other generated texts as references), **Zipf coefficient** (the linear regression coefficient between log-rank and log-frequency of generated tokens) and **repetition** (the fraction of generated texts that end in a repeating sequence of tokens). It is desirable for a model and inference procedure to produce text that is as close as possible in these metrics to the human-written reference.

We add three measures of long-range coherence:

Long-range repetition (LR_n): For a whole number n and document D, let $S(D)$ be the number of distinct tokens in D, and let $R_n(D)$ be the number of distinct tokens for which the distance between their first and last occurrence in D is at least n positions. The long-range repetition score LR_n of a corpus $\{D_1, \dots, D_{5000}\}$ is a macro-average:

$$\mathrm{LR}_n := \frac{\sum_{i=1}^{5000} R_n(D_i)}{\sum_{i=1}^{5000} S(D_i)}.$$

This simple measure of lexical coherence favors repetition of words long after they are first used, but gives lower weight to documents that degenerate into repetition of a short span.

Long-dependent token frequency (LTF): A *long-dependent token* is one to which the base LM assigns a likelihood of at least 20% given its full context, but a likelihood of less than 5% given only the 20 tokens of context preceding it. We compute the frequency of long-dependent tokens among all generated tokens.

Long-short likelihood difference (δ): The mean difference in likelihoods assigned to tokens by the base LM conditioned on full context and conditioned on 20 tokens of context.

We sample 5000 document completions from GPT-2 Large following sampling procedures with a range of boosting schemes. We consider models of the form $f_k^{\alpha_k} f_{\max}^{1-\alpha_k}$, for $k \in \{8, 16, 32, 64\}$ and $\alpha_k \in \{-0.4, -0.2, -0.1, -0.05, -0.025, 0\}$. (Such a parametrization of boosting parameters was chosen to ensure that when the context has length less than k – or the distant context has very little effect on the next word – the boosted model becomes equivalent to the untempered $f_{\max}$.) Top-p truncation with $p = 0.95$ was applied to all models.

Results. Metrics of two of the best models, with $k = 32, \alpha_k = -0.05$ and $k = 64, \alpha_k = -0.1$, are shown in Table 2. In particular, the latter model generates text that is closer to the human reference, or equally close, to the pure top-p sampling ($\alpha_k = 0$) baseline in all metrics, with the greatest improvement seen in the coherence measures.

Fig. 3 shows the dependence of selected metrics on k and α_k. Coherence boosting brings all metrics closer to those of human text. As k increases, the optimal α_k grows in magnitude. This is expected: the predictive effect of tokens more than k positions away decreases with k (f_k approaches $f_{\max}$).

We also note that a simple sampling with temperature 0.9 performs better than top-p sampling in most of the coherence metrics. This suggests that the improvements accomplished by top-p truncation come at the cost of introducing a bias towards tokens that are predictable from a short context. Coherence boosting corrects this bias without sacrificing the gains in other measures.

An example of human, top-p, and coherence boosting outputs is shown in Table B.1.

4.2 Dialog systems

This experiment is based on the Dialog System Technology Challenge 7 (DSTC7) (Galley et al., 2019), which benchmarks generation of dialog responses conditioned on one or more turns of conversation context. As a base model, we use DialoGPT (Zhang et al., 2020b), a GPT-2 Small variant that demonstrated strong results on this task.

Dialog systems' responses to the 2208 conversation prompts[3] are scored against human-written reference responses (five for each example). Following Zhang et al. (2020b), we use the n-gram overlap metrics **NIST** (Doddington, 2002), **BLEU** (Papineni et al., 2002), and **METEOR** (Lavie and Agarwal, 2007), as well as two intrinsic measures of n-gram diversity from Li et al. (2016); Zhang

[3]The DSTC7 evaluation data, scraped from Reddit, is undisclosed; we reacquire it using officially released code.

	from Holtzman et al. (2019)				lex coherence		long-dep tokens	
Inference method	ppl	BLEU-4	Zipf	rep %	LR_{50} %	LR_{100} %	δ %	LTF %
Sampling	23.53	0.28	**0.93**	**0.22**	12.92	7.71	4.87	3.28
Sampling ($T = 0.9$)	10.60	0.35	0.96	0.66	16.36	10.01	**6.54**	4.15
Nucleus ($p = 0.95$)	13.48	0.32	0.95	0.46	15.06	9.11	5.65	3.62
+ boost ($k = 32$, $\alpha_k = -0.05$)	12.81	***0.31***	*0.94*	***0.34***	*15.54*	***9.42***	*6.16*	***3.98***
+ boost ($k = 64$, $\alpha_k = -0.1$)	***12.93***	*0.32*	*0.95*	*0.46*	***15.75***	*9.67*	*6.10*	*3.95*
Human	13.19	0.31	0.93	0.28	15.95	9.51	6.54	4.03

Table 2: Distributional metrics of WebText completions. The last four columns are measures of long-range coherence (§4.1). (Nearest-to-human values in **bold**, boosting models better than top-p sampling alone in *italics*.)

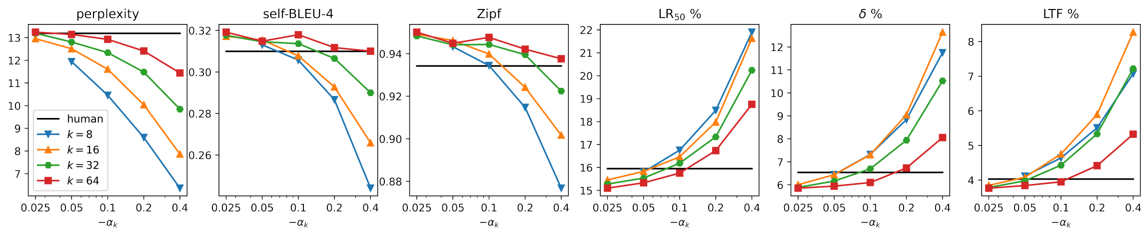

Figure 3: Effect of k and α_k on metrics from Table 2. The horizontal line marks the score of the human reference.

et al. (2018): **Distinct-n** and **Entropy-n**. It is desirable for a dialog system to reach scores close to those of the human responses in all metrics.

In addition to the decoding algorithms considered by (Zhang et al., 2020b) – beam search and greedy decoding – we consider greedy decoding with a coherence boosting model. As long and short predictors, we use DialoGPT conditioned on the full conversation context and on *only the (context-free) response generated so far*. That is, if the conversation context is S and the text generated so far is $w_1 \dots w_k$, then w_{k+1} is predicted using the model $f_{\max} f_{k+1}^{\alpha}$, evaluated on the string $S \ \langle\text{sep}\rangle \ w_1 \dots w_k$, where $\langle\text{sep}\rangle$ is the turn separator token. We consider $\alpha \in \{0, -0.1, \dots, -0.8\}$.

Results. Table 3 shows the metrics of the boosting models that reach the peak average NIST and BLEU scores ($\alpha = -0.3$ and $\alpha = -0.7$). Increasing the magnitude of α leads to responses that are more relevant to the prompt (higher BLEU and NIST) and more diverse than those from greedy decoding. As $-\alpha$ grows large, the boosting model favors creative responses that are relevant to the prompt (high NIST), but simple responses that are common in the reference data become unlikely (low BLEU).[4]

[4] Galley et al. (2019) argue that NIST and diversity metrics are more informative measures than BLEU for multi-reference scoring, since BLEU favors systems that often produce responses with little relation to the prompt (e.g., "I don't know").

We observed that the responses with $\alpha = -0.7$, despite the superior metrics, are more likely to be ungrammatical and innovate words in an effort to use tokens relevant to the prompt. In practice, improving dialog systems with coherence boosting may require techniques to prevent these side effects, such as repetition penalties or relaxation of greedy decoding to low-temperature sampling.

Finally, we note that the learning of DialoGPT was initialized with a pretrained GPT-2 and uses GPT-2's end-of-text token as the turn separator. This choice may reduce DialoGPT's attention to past turns, as tokens *preceding* the end-of-text token are never informative in GPT-2's training data.

5 Experiments: Language understanding

We evaluate coherence boosting on language understanding and inference tasks, where examples are expected to be highly coherent. Code for the experiments in this section is included in the SI.

We study 5 categories of tasks with 15 datasets. **(1) Cloze tasks**: *StoryCloze* (Mostafazadeh et al., 2016), *HellaSwag* (Zellers et al., 2019), and *COPA* (Roemmele et al., 2011). **(2) Question answering**: *CommonsenseQA* (CsQA) (Talmor et al., 2019), *OpenBookQA* (OBQA) (Mihaylov et al., 2018), *ARC Easy / Challenge* (ARC-E/C) (Clark et al., 2018), and *PIQA* (Bisk et al., 2020). **(3) Text classification**: *SST-2/5* (Socher et al., 2013),

	NIST		BLEU			diversity metrics			
Inference method	N-2	N-4	B-2	B-4	METEOR	Ent-4	Dist-1	Dist-2	avg len
Beam ($b = 10$)	0.02	0.02	12.81	3.23	5.35	6.06	14.03	34.59	5.81
Greedy	1.62	1.63	9.92	1.72	6.78	6.45	6.19	17.56	13.30
+ boost ($\alpha = -0.3$)	0.72	0.73	**13.82**	**3.53**	**6.91**	8.54	**16.81**	49.35	9.75
+ boost ($\alpha = -0.7$)	**1.78**	**1.79**	6.33	0.94	5.55	**9.78**	28.00	**72.46**	**16.63**
Human	2.63	2.65	12.36	3.13	8.31	10.44	16.65	67.01	18.73

Table 3: Metrics of DialoGPT responses on DSTC7. Nearest-to-human values in each column are **bolded**.

	GPT-2 Small (125M)				GPT-2 XL (1.6B)				GPT-3 175B			
	$f_{\max}$	$\alpha = -1$	$\alpha = \alpha^*$	α^*	$f_{\max}$	$\alpha = -1$	$\alpha = \alpha^*$	α^*	$f_{\max}$	$\alpha = -1$	$\alpha = \alpha^*$	α^*
StoryCloze	59.91	**64.78**	64.24	*−1.02*	67.56	75.09	**76.75**	*−0.69*	79.16	82.90	**86.85**	*−0.64*
HellaSwag	28.92	30.99	**31.84**	*−0.90*	40.00	42.60	**47.66**	*−0.78*	59.18	62.66	**72.35**	*−0.76*
COPA	62.00	56.00	**64.00**	*−0.69*	73.00	70.00	**77.00**	*−0.44*	93.00	87.00	**94.00**	*−0.52*
CsQA	29.48	42.26	**43.16**	*−0.81*	37.84	50.45	**52.91**	*−0.75*	61.10	67.98	**70.43**	*−0.68*
OBQA	11.20	30.60	**40.80**	*−1.62*	15.60	38.40	**47.00**	*−1.88*	28.00	52.20	**52.60**	*−1.09*
ARC-E	43.81	42.09	**46.00**	*−0.34*	58.29	51.43	**60.31**	*−0.36*	76.22	69.19	**78.32**	*−0.44*
ARC-C	19.03	26.11	**29.10**	*−4.19*	25.00	33.53	**34.39**	*−1.14*	43.94	**50.60**	49.23	*−1.08*
PIQA	62.89	57.45	**63.44**	*−0.61*	70.84	60.45	**71.49**	*−0.43*	79.27	66.32	**78.94**	*−0.60*
SST2	65.68	74.74	**82.32**	*−2.22*	86.38	84.51	**86.93**	*−0.09*	86.16	88.14	**89.84**	*−0.54*
SST5	25.93	**30.90**	**30.90**	*−1.20*	28.69	**38.73**	36.92	*−1.69*	31.22	34.75	**38.51**	*−1.39*
AGNews	58.55	60.78	**62.20**	*−0.62*	67.17	67.43	**68.26**	*−0.40*	71.66	71.74	**71.75**	*0.16*
TREC	23.40	29.60	**32.20**	*−0.80*	23.40	27.40	**40.00**	*−0.79*	52.40	47.00	**56.00**	*−0.56*
BoolQ	49.36	58.07	**62.14**	*−3.04*	62.14	**63.46**	63.21	*−0.64*	71.56	**73.70**	72.69	*−0.39*
RTE	51.26	49.82	**53.79**	*−0.30*	**49.10**	48.74	**49.10**	*0.90*	55.96	57.40	**60.29**	*−0.60*
CB	12.50	23.21	**48.21**	*−2.40*	30.36	51.79	**66.07**	*−1.90*	5.36	25.00	**28.57**	*−1.91*

Table 4: Testing accuracy (%) of three representative GPT models on multiple-choice tasks. The first column for each model is the full-context model, the second is our model only when $\alpha = -1$ (a baseline), and the third column is our model with the optimal α chosen on a validation set. The fourth column shows this optimal value of α.

TREC (Voorhees and Tice, 2000), *AGNews* (Zhang et al., 2015). **(4) Natural language inference**: *RTE* (Dagan et al., 2005), *CB* (De Marneffe et al., 2019), and *BoolQ* (Clark et al., 2019). (5) **Fact knowledge retrieval**: *LAMA* (Petroni et al., 2019).

All tasks except LAMA are formulated as multiple-choice problems. We convert text classification and inference tasks to multiple-choice tasks by choosing meaningful answer words, e.g., "True"/"False". The prediction is made by selecting the choice with the highest LM likelihood.

For in-context learning of GPT models, prompt formats greatly impact performance. We follow previous work (Brown et al., 2020; Zhao et al., 2021; Holtzman et al., 2019) to create natural prompts to enlarge the effectiveness of in-context learning, but we do not aim to optimize the full and context-free prompt format: our goal is to evaluate coherence boosting models with a fixed prompt. The prompt formats we use are listed in Table C.1. As described in §2, within each prompt we identify a *premise-free context*, which is used as the context for the short-range model in coherence boosting.

For each dataset, we pick the optimal value α^* of the parameter α on the validation set and report the accuracy on testing set. (If no testing set is publicly available, we choose α on a subset of the training set and report the final number on the validation set.) Across all experiments, we do not put any few-shot examples in the prompt.

For the knowledge retrieval task, we follow Zhao et al. (2021)'s data split of LAMA and evaluate GPT models on facts whose missing answers are at the end of the sentence (to fit the nature of autoregressive language models). We limit the prompt length to be larger than 5 tokens and rerun the model from Zhao et al. (2021) on the new data.

Results: Multiple-choice tasks. Results of three representative base models on all multiple-choice tasks are presented in Table 4. (Results for all models are in Tables D.1 and D.2.) We compare

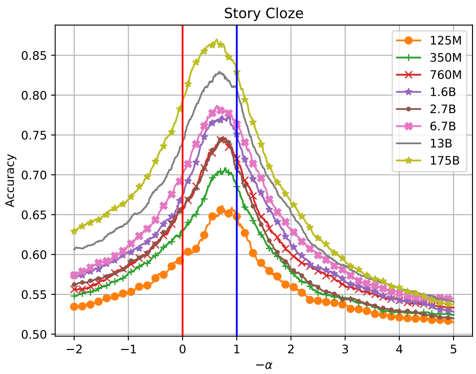

Figure 4: Model comparison for the StoryCloze task. The red line $\alpha = 0$ indicates the base model, and the blue line $\alpha = -1$ is an unconditional normalization.

	GPT-2				GPT-3			
	125M	350M	760M	1.6B	2.7B	6.7B	13B	175B
$f_{\max}$	8.48	14.78	13.88	14.29	17.33	19.42	22.06	26.76
Zhao et al. (2021)	17.45	22.87	23.90	23.97	26.30	30.57	31.96	34.78
CB ($\alpha_k = \alpha_k^*$)	19.85	22.87	25.74	25.43	28.75	32.25	35.02	37.57
α_k^*	−0.5	−0.5	−0.5	−0.5	−0.5	−0.5	−0.5	−0.4
k^*	1	2	3	3	1	1	1	2

Table 5: Accuracies (%) of GPT models on LAMA.

our best model with two baselines, $\alpha = 0$ ($f_{\max}$) and $\alpha = -1$. The former one is the original full-context model, while the latter is, for most tasks, a form of unconditional probability normalization as performed by Brown et al. (2020); Holtzman et al. (2021). We also compare our best model with other inference methods (Holtzman et al., 2021; Min et al., 2021) in Tables D.3 and D.4.

By comparing the third column with the first two columns within each model in Table 4, we can see that our method with the selected α generally improves the accuracy on all tasks. Some of the improvements are dramatic, where boosted GPT-2 Small outperforms GPT-2 XL's base model (e.g., CsQA, OBQA, ARC-C) and is even comparable with GPT-3 175B's base model (e.g., SST-2, SST-5, RTE). We make similar conclusions when comparing coherence boosting with other inference methods in Tables D.3 and D.4.

We observe that the optimal α depends on tasks and models (fourth column within each model), which means that α cannot be heuristically set to 0 or -1 as in past work. This finding suggests the necessity of searching for an optimal α. We visualize the accuracy curve by varying α in the testing set of all datasets. We show the curve for StoryCloze in Fig. 4 and present similar figures for all tasks in Figs. D.1 and D.2.

Consistent with the results on LAMBADA (§3), the optimal α is usually negative, and its absolute value tends to decrease with the model size. We selected the optimal α by the validation set, but future work may explore automatic and adaptive methods for setting this parameter. Notice that all experiments required only a *single pass* through the data to compute answer likelihoods conditioned on full and premise-free contexts – no iterative gradient-based finetuning was applied.

Results: Knowledge retrieval. Unlike LAMBADA, where long contexts are required for inferring the last word, LAMA contains much shorter sentences for knowledge facts, i.e., (subject, relation, object). A recent study (Cao et al., 2021) shows that the prediction is biased by the relation in the short context, i.e., the answer to a prompt (e.g., "Dante was born in ___") can be induced by the relation ("was born in") without the subject. Coherence boosting mitigates the influence of those short contexts by making the prediction dependent on a longer context containing the subject.

We present results for all models on LAMA in Table 5. We also compare our model with contextual calibration (CC) (Zhao et al., 2021), which processes the LM's output probabilities with a log-linear model.[5] Coherence boosting with the selected α and k outperforms both the base model and CC by significant margins.

6 Conclusion

We have illustrated the hyposensitivity of pretrained language models to long-range content and proposed a simple inference-time remedy. Future work can consider *training* regimes that encourage learning of long dependencies, adaptive selection of boosting weights, mimicking coherence boosting by scaling attention at different distances, and comparative analysis of optimal weights for various text domains and language model architectures.

Procedures that force LMs to be more focused on a prompt, or a specific part of it, when generating or ranking tokens can benefit algorithms that search for combinations of words through sampling. It would be interesting to use coherence boosting in non-autoregressive text generation algorithms, such as to accelerate the mixing of MCMC methods for constrained text generation (Miao et al., 2019; Zhang et al., 2020a; Malkin et al., 2021).

[5]Note that CC applies a log-linear model to the *probability* domain, not the logit domain, which does not have an information-theoretic interpretation.

Ethics statement

We hope and expect to see a nonnegative net societal impact from better text generation and ranking algorithms in general and from this work in particular. As we have shown, there is room to improve the inference procedures used with small language models, which incur lower costs than training and evaluation of large models. However, researchers should bear in mind the risks and potential misuse of automatic generation of long-form text.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *International Conference on Learning Representations (ICLR)*.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020. Piqa: Reasoning about physical commonsense in natural language. *Association for the Advancement of Artificial Intelligence (AAAI)*.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

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

Boxi Cao, Hongyu Lin, Xianpei Han, Le Sun, Lingyong Yan, Meng Liao, Tong Xue, and Jin Xu. 2021. Knowledgeable or educated guess? revisiting language models as knowledge bases. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1860–1874, Online. Association for Computational Linguistics.

Stanley F. Chen and Joshua Goodman. 1996. An empirical study of smoothing techniques for language modeling. In *34th Annual Meeting of the Association for Computational Linguistics*, pages 310–318, Santa Cruz, California, USA. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. 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 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? Try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Machine Learning Challenges Workshop*, pages 177–190. Springer.

Marie-Catherine De Marneffe, Mandy Simons, and Judith Tonhauser. 2019. The commitmentbank: Investigating projection in naturally occurring discourse. In *proceedings of Sinn und Bedeutung*, volume 23, pages 107–124.

George R. Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Human Language Technology Research*.

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.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. 2000. Additive logistic regression: A statistical view of boosting. *The Annals of Statistics*, 28:337–407.

Michel Galley, Chris Brockett, Xiang Gao, Jianfeng Gao, and William B. Dolan. 2019. Grounded response generation task at DSTC7. *Dialog System Technology Challenges 7 (AAAI workshop)*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *International Conference on Learning Representations (CLR)*.

Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. 2021. Surface form competition: Why the highest probability answer isn't always right. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7038–7051, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Urvashi Khandelwal, He He, Peng Qi, and Dan Jurafsky. 2018. Sharp nearby, fuzzy far away: How neural language models use context. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 284–294, Melbourne, Australia. Association for Computational Linguistics.

Alon Lavie and Abhaya Agarwal. 2007. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231, Prague, Czech Republic. Association for Computational Linguistics.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.

Nikolay Malkin, Sameera Lanka, Pranav Goel, and Nebojsa Jojic. 2021. Studying word order through iterative shuffling. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10351–10366, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Clara Meister and Ryan Cotterell. 2021. Language model evaluation beyond perplexity. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5328–5339, Online. Association for Computational Linguistics.

Ning Miao, Hao Zhou, Lili Mou, Rui Yan, , and Li Lei. 2019. CGMH: Constrained sentence generation by Metropolis-Hastings sampling. *Association for the Advancement of Artificial Intelligence (AAAI)*.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Sewon Min, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2021. Noisy channel language model prompting for few-shot text classification. *arXiv preprint arXiv:2108.04106*.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego, California. Association for Computational Linguistics.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1525–1534, Berlin, Germany. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? 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 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Ofir Press, Noah A. Smith, and Mike Lewis. 2021. Train short, test long: Attention with linear biases enables input length extrapolation. *arXiv preprint arXiv:2108.12409*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S. Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. *AAAI Spring Symposium Series*.

Aurko Roy, Mohammad Saffar, Ashish Vaswani, and David Grangier. 2021. Efficient content-based sparse attention with routing transformers. *Transactions of the Association for Computational Linguistics*, 9:53–68.

Chinnadhurai Sankar, Sandeep Subramanian, Chris Pal, Sarath Chandar, and Yoshua Bengio. 2019. Do neural dialog systems use the conversation history effectively? an empirical study. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 32–37, Florence, Italy. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Simeng Sun, Kalpesh Krishna, Andrew Mattarella-Micke, and Mohit Iyyer. 2021. Do long-range language models actually use long-range context? In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 807–822, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. 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 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Neural Information Processing Systems (NIPS)*.

Ellen M Voorhees and Dawn M Tice. 2000. Building a question answering test collection. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 200–207.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *Neural Information Processing Systems (NeurIPS)*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Maosen Zhang, Nan Jiang, Lei Li, and Yexiang Xue. 2020a. Language generation via combinatorial constraint satisfaction: A tree search enhanced Monte-Carlo approach. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1286–1298, Online. Association for Computational Linguistics.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Neural Information Processing Systems (NIPS)*.

Yizhe Zhang, Michel Galley, Jianfeng Gao, Zhe Gan, Xiujun Li, Chris Brockett, and William B. Dolan. 2018. Generating informative and diverse conversational responses via adversarial information maximization. *Neural Information Processing Systems (NeurIPS)*.

Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020b. DIALOGPT : Large-scale generative pre-training for conversational response generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 270–278, Online. Association for Computational Linguistics.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. *International Conference on Machine Learning (ICML)*.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. *ACM SIGIR Conference on Research and Development in Information Retrieval*.

A On multi-objective training and log-linear weights

The section extends the discussion in §2.1.

Recall that the language model f is trained on the multi-objective loss (3):

$$\sum_{k=1}^{M} \lambda_k \underbrace{\mathbb{E}_{w_1 \ldots w_{M+1} \in \mathcal{D}} \left[-\log f_k(w_{M+1} | w_1, \ldots, w_M; \theta) \right]}_{\mathcal{L}_k(\theta)}, \quad \lambda_k = \frac{1}{M}.$$

As we saw in the main text, the scalarization weights λ_k are uniform as a consequence of the training regime. However, evaluation procedures effectively give nonuniform weight to the M prediction losses.

Some vector calculus. Denote by $\hat{\theta}(\boldsymbol{\lambda})$ a local optimum of the above optimization problem for general linear combination weights $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_M)$. Under suitable regularity conditions, the gradient of the combined loss vanishes:

$$\sum_k \lambda_k \frac{\partial \mathcal{L}_k(\theta)}{\partial \theta} \bigg|_{\theta = \hat{\theta}(\lambda)} = \mathbf{0}. \tag{4}$$

Assuming the Hessian $\mathbf{A}$ of the optimization criterion $\sum_k \lambda_k \mathcal{L}_k(\theta)$ is nonsingular, we can implicitly differentiate (4) with respect to $\boldsymbol{\lambda}$ to obtain the matrix derivative

$$\frac{\partial \hat{\theta}(\boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}} = -\mathbf{A}^{-1} \frac{\partial \left(\mathcal{L}_1(\theta), \ldots, \mathcal{L}_M(\theta) \right)}{\partial \theta^T} \bigg|_{\theta = \hat{\theta}(\lambda)}. \tag{5}$$

The local dependence of the losses on the scalarization weights can be expressed as a bilinear form evaluated on $\frac{\partial \mathcal{L}_i}{\partial \theta}$ and $\frac{\partial \mathcal{L}_j}{\partial \theta}$:

$$\frac{\partial \mathcal{L}_i(\hat{\theta}(\boldsymbol{\lambda}))}{\partial \lambda_j} = \frac{\partial \mathcal{L}_i}{\partial \theta} \bigg|_{\theta = \hat{\theta}(\lambda)} \frac{\partial \hat{\theta}(\boldsymbol{\lambda})}{\partial \lambda_j} = -\frac{\partial \mathcal{L}_i}{\partial \theta} \mathbf{A}^{-1} \frac{\partial \mathcal{L}_j}{\partial \theta^T} \bigg|_{\theta = \hat{\theta}(\lambda)}. \tag{6}$$

Because $\hat{\theta}$ is a local minimizer, $-\mathbf{A}^{-1}$ is negative definite. In particular, any $\frac{\partial \mathcal{L}_i(\hat{\theta}(\lambda))}{\partial \lambda_i}$ is negative. This expresses the intuitive fact that if an infinitesimally higher weight is given to some prediction loss in optimization, the value of this loss at the optimum will be infinitesimally lower.

For concreteness, consider how the highest-length prediction loss $\mathcal{L}_M(\hat{\theta}(\boldsymbol{\lambda}))$ changes when λ_M is increased and the λ_j ($j \neq i$) are decreased with rate proportional to λ_j, while $\sum \lambda_j$ is kept constant. That is, let $\boldsymbol{\beta} = \left(-\lambda_1, \ldots, -\lambda_{i-1}, \sum_{j \neq i} \lambda_j, -\lambda_{i+1}, \ldots, -\lambda_M \right)$. Then

$$\frac{d \mathcal{L}_i(\hat{\theta}(\boldsymbol{\lambda} + t\boldsymbol{\beta}))}{dt} = \sum_j \frac{\partial \mathcal{L}_i}{\partial \lambda_j} \beta_j = -\frac{\partial \mathcal{L}_i}{\partial \theta} \mathbf{A}^{-1} \sum_j \frac{\partial \mathcal{L}_j}{\partial \theta^T} \beta_j = -\frac{\partial \mathcal{L}_i}{\partial \theta} \mathbf{A}^{-1} \frac{\partial \mathcal{L}_i}{\partial \theta^T} \sum_j \lambda_j \leq 0, \tag{7}$$

where the last two equalities follow from (6) and (4), respectively, and the inequality holds because $\mathbf{A}^{-1}$ is positive definite. So we have shown that, in nondegenerate cases, the $\mathcal{L}_M(\theta)$ term of the optimization criterion decreases under the locally optimal weights θ when λ_M is infinitesimally increased in this way.

Log-linear mixture of predictors. Returning to coherence boosting, suppose that we aim to build out of the predictors $f_k(-; \hat{\theta}(\boldsymbol{\lambda}))$ a new predictor g that would have lower negative log-likelihood on prediction of a word given the maximum-length context:

$$\mathbb{E}_{w_1 \ldots w_{M+1} \in \mathcal{D}} \left[-\log g(w_{M+1} \mid w_1, \ldots, w_M) \right] < \mathbb{E} \left[-\log f_M(w_{M+1} \mid w_1, \ldots, w_M; \hat{\theta}(\boldsymbol{\lambda})) \right].$$

As we just saw, using this predictor in place of f_M achieves the same direction of movement in the prediction loss as optimizing with higher weight λ_M.

A naïve guess – not a proper predictor, as its outputs do not sum to 1 – would lightly perturb f_M by log-linearly mixing small multiples of the f_k weight weights β_k summing to 0:

$$g^{(t)}_{\text{naïve}}(w_1, \ldots, w_M) = \exp\left(\log f_M(w_1, \ldots, w_M; \hat{\theta}(\boldsymbol{\lambda})) + t \sum_k \beta_k \log f_k(-, \hat{\theta}(\boldsymbol{\lambda}))\right).$$

Then, by linearity of expectation,

$$\begin{aligned} \left.\frac{d}{dt}\right|_{t=0} \mathbb{E}\left[-\log g^{(t)}_{\text{naïve}}(w_{M+1} \mid w_1, \ldots, w_M)\right] &= \sum_k \beta_k \mathbb{E}\left[-\log f_k(w_{M+1} \mid w_1, \ldots, w_M; \hat{\theta}(\boldsymbol{\lambda}))\right] \\ &= \sum_k \beta_k \mathcal{L}_k(\hat{\theta}(\boldsymbol{\lambda})). \end{aligned} \tag{8}$$

This quantity is negative if, for example, $\mathcal{L}_M(\hat{\theta}(\boldsymbol{\lambda}))$ is minimal among the $\mathcal{L}_k(\hat{\theta}(\boldsymbol{\lambda}))$.

Reintroducing the normalization condition, we define a candidate function $g^{(t)}$ as the normalization of $g^{(t)}_{\text{naïve}}$ over w_{M+1} and compute, with the aid of (8) and using that the g_k are normalized to simplify the derivative of $\log \sum \exp$:

$$\begin{aligned} &\left.\frac{d}{dt}\right|_{t=0} \mathbb{E}\left[-\log g^{(t)}(w_{M+1} \mid w_1, \ldots, w_M)\right] \\ &= \sum_k \beta_k \mathcal{L}_k(\hat{\theta}(\boldsymbol{\lambda})) + \left.\frac{d}{dt}\right|_{t=0} \mathbb{E} \log \sum_w g^{(t)}_{\text{naïve}}(w \mid w_1, \ldots, w_M) \\ &= \sum_k \beta_k \mathcal{L}_k(\hat{\theta}(\boldsymbol{\lambda})) + \mathbb{E} \sum_w \left\langle \sum_k \beta_k \log f_k(w_1, \ldots, w_M; \hat{\theta}(\boldsymbol{\lambda})), f_M(w_1, \ldots, w_M; \hat{\theta}(\boldsymbol{\lambda})) \right\rangle \\ &= \sum_k \beta_k \mathcal{L}_k(\hat{\theta}(\boldsymbol{\lambda})) - \sum_k \beta_k \mathbb{E}\left[D_{\text{KL}}\left(f_M(w_1, \ldots, w_M; \hat{\theta}(\boldsymbol{\lambda})) \,\|\, f_k(w_1, \ldots, w_M; \hat{\theta}(\boldsymbol{\lambda}))\right)\right], \end{aligned} \tag{9}$$

where the last line used that $\sum \beta_k = 0$.

In practice, we are interested in sparse log-linear mixtures. Taking $\beta_M = 1, \beta_k = -1$ for a single k, and all other $\beta_i = 0$, we conclude that the boosted model proportional to $f_M^{1+t} f_k^{-t}$ is a better predictor than f_M alone if the difference between prediction losses $\mathcal{L}_M$ and $\mathcal{L}_k$ is greater than the average KL divergence between the predictions f_M and f_k.

B Example WebText completion

An example of human, top-p, and coherence boosting outputs is shown in Table B.1. All outputs for all boosting schemes are included in the SI.

University of Alberta doctoral candidate Scott Persons measured the limb length for 53 different species of carnivorous dinosaurs to create an equation to score how much each was built for speed.	
Human	A new study published out of the University of Alberta has found that carnivorous dinosaurs evolved to become faster and faster over time — and the proof is in the leg length. As part of his doctoral research, lead author Scott Persons travelled the world measuring the limb lengths for 53 different species of carnivorous bipedal dinosaurs, including the Velociraptor, Allosaurus and Tyrannosaurus rex. In particular, Persons looked at the length of the dinosaurs' legs below the knee. As a rule, he said, the longer the lower leg is in comparison to the upper leg, the faster the animal is. Modern-day cheetahs have proportionately longer legs than hyenas. That relationship is mirrored in the animals' speeds, Persons said. "That's true for modern carnivores, and must have been true for dinosaurs," Persons said in a statement released Wednesday.
Nucleus ($p = 0.95$)	"In a lot of areas of the world, dinosaurs at the time didn't have legs, they had weapons," said Persons. So he used his scientific skill, titled "Toill's Law of Tragedy," to give the dinosaurs an on-the-ground estimate of what their internal organs weighed. For example, wood-eating reptiles, such as the dinosaur Lufengosaurus and the ceratopsian Tyrannosaurus rex, would have weighed up to 52 kg (115 lbs) each. And as many as 50 plates should have been available to hunt, Persons said. But what's important is that a dinosaur's pack or herd probably wouldn't have been able to feed all three in one sitting. Dinosaurs were extremely resilient, as very few of them survived from one kill to the next and later on would start a "phased trip" back to where they'd been killed, Persons said.
+ boost ($k = 64$, $\alpha_k = -0.1$)	"In a lot of areas of the world, dinosaurs at the time didn't have legs, they had weapons," said Persons. So he used his undergraduate course to work on a formula to calculate a specific height for each carnivore based on the evolutionary requirements of different dinosaurs and when it would have been optimal for meat eating. He then moved that formula into his doctoral thesis project, which is looking at limb lengths for medium-sized, fearsome carnivores. "His thesis is about how to measure the shape of different places, plant diversity and evolutionary biology and kind of how we see dinosaurs and the relationships between species," said Susanne Dufoot, an associate professor of paleontology at McMaster University who recently returned from attending Persons' research. "It's interesting because he's basically done the legwork, developed this model that can give us information about plant species." 'He was an amazing creature'

Table B.1: Completions of an article: written by a human (original WebText) and sampled from GPT-2 Large with top-p sampling, with and without coherence boosting. While top-p sampling produces text that is coherent at first glance – it is free of repetition and nonce words – the topic of the article meanders from limb length to internal organs and killing, and nonsensical comments appear ('Toill's Law of Tragedy', herbivorous ceratopsian T-Rex, etc.). The output with coherence boosting is largely free of these issues, maintaining focus on limb length and diet.

C Prompt formats for multiple-choice tasks

Task	Prompt format
Story Cloze	[Context] [Completion]
HellaSwag	[Context] he/she/they/... [Completion]
COPA	[Premise] because/so [Hypothesis]
CommonsenseQA	[Question] the answer is: [Answer]
OpenBookQA	[Question] the answer is: [Answer]
ARC Easy	Question: [Question] Answer: [Answer]
ARC Challenge	Question: [Question] Answer: [Answer]
PIQA	Question: [Question] Answer: [Answer]
SST-2	[Context] This quote has a tone that is: [Label]
SST-5	[Context] This quote has a tone that is: [Label]
AGNews	Title: [Title] Summary: [Context] Topic: [Label]
TREC	[Question] The answer to this question will be [Label]
BoolQ	[Passage]\n Question: [Hypothesis] True or False? Answer: [Label]
RTE	[Premise]\n question: [Hypothesis] true or false?\n answer: [Label]
CB	Given question: [Premise] Is [Hypothesis] true, false or neither?\n The answer is: [Label]

Table C.1: Prompt formats used in our experiments. The full context is underlined in blue; the premise-free context is also underlined in red. We mainly draw inspiration from (Brown et al., 2020; Holtzman et al., 2021; Zhao et al., 2021) to make our prompts more natural to facilitate boosting the coherence of the completion.

D Additional results

	GPT-3 Small				GPT-3 Medium				GPT-3 Large				GPT-3 XL			
	$f_{\max}$	$\alpha = 1$	$\alpha = \alpha^*$	α^*	$f_{\max}$	$\alpha = 1$	$\alpha = \alpha^*$	α^*	$f_{\max}$	$\alpha = 1$	$\alpha = \alpha^*$	α^*	$f_{\max}$	$\alpha = 1$	$\alpha = \alpha^*$	α^*
Story Cloze	66.0	70.9	**74.5**	*-0.8*	70.1	76.3	**78.0**	*-0.8*	74.2	**82.9**	80.8	*-0.7*	79.3	82.9	**86.9**	*-0.6*
HellaSwag	35.7	38.9	**42.0**	*-0.9*	42.8	46.8	**51.3**	*-0.8*	50.5	55.1	**62.2**	*-0.8*	59.2	62.7	**72.3**	*-0.8*
COPA	73.0	71.0	**75.0**	*-0.6*	**85.0**	79.0	83.0	*-0.7*	**84.0**	83.0	**84.0**	*-0.6*	93.0	87.0	**94.0**	*-0.5*
CsQA	34.6	46.4	**48.0**	*-0.7*	42.4	51.4	**53.0**	*-0.7*	50.0	57.5	**60.4**	*-0.7*	61.1	68.0	**70.4**	*-0.7*
OBQA	16.0	39.8	**46.6**	*-2.2*	16.4	41.8	**48.8**	*-1.4*	20.8	45.4	**47.8**	*-1.6*	28.0	52.2	**52.6**	*-1.1*
ARC-E	51.3	48.1	**56.0**	*-0.5*	59.8	54.8	**63.3**	*-0.4*	68.4	60.3	**70.7**	*-0.5*	76.2	69.2	**78.3**	*-0.4*
ARC-C	22.6	30.8	**31.1**	*-1.4*	27.5	35.3	**35.5**	*-1.2*	33.9	**41.8**	**41.8**	*-0.9*	43.9	**50.6**	49.2	*-1.1*
PIQA	69.0	57.5	**69.6**	*-0.4*	74.4	60.4	**74.7**	*-0.4*	76.3	64.2	**77.7**	*-0.4*	**79.3**	66.3	78.9	*-0.6*
SST-2	70.6	79.8	**84.6**	*-2.3*	69.5	75.2	**88.0**	*-4.8*	66.8	65.2	**70.0**	*2.0*	86.2	88.1	**89.8**	*-0.5*
SST-5	**26.7**	26.6	26.1	*-1.1*	29.3	**30.7**	30.0	*-1.2*	28.1	**33.2**	30.1	*-0.8*	31.2	34.8	**38.5**	*-1.4*
AGNews	67.1	69.2	**69.5**	*-1.2*	63.3	64.8	**65.4**	*-2.0*	69.2	65.7	**69.5**	*-0.3*	71.7	71.7	**71.8**	*0.2*
TREC	28.8	57.2	**57.4**	*-1.0*	30.2	62.6	**63.6**	*-0.8*	35.2	28.8	**37.2**	*-0.3*	52.4	47.0	**56.0**	*-0.6*
BoolQ	60.7	**62.4**	62.2	*-1.4*	61.6	63.4	**63.5**	*-0.9*	64.2	65.6	**68.1**	*-4.5*	71.6	**73.7**	72.7	*-0.4*
RTE	49.8	**51.3**	**51.3**	*-3.6*	**54.5**	50.5	49.1	*-1.2*	53.8	**55.6**	55.2	*-1.4*	56.0	57.4	**60.3**	*-0.6*
CB	**33.9**	19.6	21.4	*-0.7*	8.9	25.0	**39.3**	*-1.9*	**32.1**	28.6	**32.1**	*-0.2*	5.4	25.0	**28.6**	*-1.9*

Table D.1: Accuracy (%) of GPT-3 models on all multiple-choice tasks, in the same format as Table 4.

	GPT-2 Small				GPT-2 Medium				GPT-2 Large				GPT-2 XL			
	$f_{\max}$	$\alpha = -1$	Ours	α^*	$f_{\max}$	$\alpha = -1$	Ours	α^*	$f_{\max}$	$\alpha = -1$	Ours	α^*	$f_{\max}$	$\alpha = -1$	Ours	α^*
Story Cloze	59.9	**64.8**	64.2	*-1.0*	63.0	68.5	**70.4**	*-0.7*	66.0	72.0	**74.4**	*-0.8*	67.6	75.1	**76.8**	*-0.7*
HellaSwag	28.9	31.0	**31.8**	*-0.9*	33.4	36.6	**38.1**	*-0.9*	36.6	39.5	**43.0**	*-0.8*	40.0	42.6	**47.7**	*-0.8*
COPA	62.0	56.0	**64.0**	*-0.7*	69.0	69.0	**72.0**	*-0.6*	**69.0**	60.0	**69.0**	*-0.6*	73.0	70.0	**77.0**	*-0.4*
CsQA	29.5	42.3	**43.2**	*-0.8*	31.3	44.6	**45.3**	*-0.8*	35.7	47.3	**50.0**	*-0.8*	37.8	50.5	**52.9**	*-0.8*
OBQA	11.2	30.6	**40.8**	*-1.6*	15.6	34.8	**43.8**	*-2.1*	13.6	34.4	**44.2**	*-1.8*	15.6	38.4	**47.0**	*-1.9*
ARC-E	43.8	42.1	**46.0**	*-0.3*	49.1	44.5	**51.3**	*-0.6*	53.2	46.5	**56.2**	*-0.5*	58.3	51.4	**60.3**	*-0.4*
ARC-C	19.0	26.1	**29.1**	*-4.2*	21.5	**27.3**	27.0	*-1.0*	21.7	28.3	**29.1**	*-2.8*	25.0	33.5	**34.4**	*-1.1*
PIQA	62.9	57.5	**63.4**	*-0.6*	67.6	56.1	**68.1**	*-0.5*	**70.3**	60.0	70.1	*-0.4*	70.8	60.4	**71.5**	*-0.4*
SST-2	65.7	74.7	**82.3**	*-2.2*	72.6	83.5	**88.2**	*-2.0*	77.2	87.6	**88.0**	*-1.2*	86.4	84.5	**86.9**	*-0.1*
SST-5	25.9	**30.9**	**30.9**	*-1.2*	20.5	33.3	**35.2**	*-1.1*	29.1	31.8	**35.2**	*-1.4*	28.7	**38.7**	36.9	*-1.7*
AGNews	58.6	60.8	**62.2**	*-0.6*	64.6	**66.5**	66.3	*-0.7*	62.6	62.1	**63.8**	*-0.4*	67.2	67.4	**68.3**	*-0.4*
TREC	23.4	29.6	**32.2**	*-0.8*	27.4	17.6	**36.0**	*-0.4*	22.6	**45.4**	44.2	*-1.2*	23.4	27.4	**40.0**	*-0.8*
BoolQ	49.4	58.1	**62.1**	*-3.0*	56.6	**61.8**	**61.8**	*-0.9*	61.2	**62.3**	62.2	*-1.8*	62.1	**63.5**	63.2	*-0.6*
RTE	51.3	49.8	**53.4**	*-0.3*	53.1	50.9	**53.8**	*-0.2*	**53.1**	46.6	50.2	*-1.2*	**49.1**	48.7	**49.1**	*0.9*
CB	12.5	23.2	**48.2**	*-2.4*	8.9	37.5	**55.4**	*-2.5*	8.9	32.1	**53.6**	*-2.5*	30.4	51.8	**66.1**	*-1.9*

Table D.2: Accuracy (%) of GPT-2 models on all multiple-choice tasks, in the same format as Table 4.

	GPT-3 Small			GPT-3 Medium		GPT-3 Large		GPT-3 XL		
	PMI	CC	Ours	PMI	Ours	PMI	Ours	PMI	CC	Ours
Story Cloze	73.1	-	**74.5**	76.8	**78.0**	79.9	**80.8**	84.0	-	**86.9**
HellaSwag	34.2	-	**42.0**	40.0	**51.3**	45.8	**62.2**	53.5	-	**72.3**
COPA	74.4	-	**75.0**	77.0	**83.0**	**84.2**	84.0	89.2	-	**94.0**
CsQA	44.7	-	**48.0**	50.3	**53.0**	58.5	**60.4**	66.7	-	**70.4**
OBQA	42.8	-	**46.6**	48.0	**48.8**	**50.4**	47.8	**58.0**	-	52.6
ARC-E	44.7	-	**56.0**	51.5	**63.3**	57.7	**70.7**	63.3	-	**78.3**
ARC-C	30.5	-	**31.1**	33.0	**35.5**	38.5	**41.8**	45.5	-	**49.2**
SST-2	72.3	71.4	**84.6**	80.0	**88.0**	**81.0**	70.0	71.4	75.8	**89.8**
SST-5	23.5	-	**26.1**	**32.0**	30.0	19.1	**30.1**	29.6	-	**38.5**
AGNews	67.9	63.2	**69.5**	57.4	**65.4**	**70.3**	69.5	**74.7**	73.9	71.8
TREC	57.2	38.8	**57.4**	61.6	**63.6**	32.4	**37.2**	**58.4**	57.4	56.0
BoolQ	53.5	-	**62.2**	61.0	**63.5**	60.3	**68.1**	64.0	-	**72.7**
RTE	**51.6**	49.5	51.3	48.7	**49.1**	54.9	**55.2**	**64.3**	57.8	60.3
CB	**57.1**	50.0	21.4	39.3	39.3	**50.0**	32.1	**50.0**	48.2	28.6

Table D.3: Performance comparison with other inference methods on GPT-3 models. PMI (Holtzman et al., 2021) is an unconditional probability normalization method, CC (Zhao et al., 2021) is the contextual calibration method. We compare them in the zero-shot setting.

	GPT-2 Small		GPT-2 Medium		GPT-2 Large			GPT-2 XL		
	PMI	Ours	PMI	Ours	PMI	Channel	Ours	PMI	CC	Ours
Story Cloze	**67.0**	64.2	**71.6**	70.4	73.4	-	**74.4**	76.3	-	**76.8**
HellaSwag	29.1	**31.8**	32.8	**38.1**	35.1	-	**43.0**	37.8	-	**47.7**
COPA	62.8	**64.0**	70.0	**72.0**	**69.4**	-	69.0	71.6	-	**77.0**
CsQA	36.4	**43.2**	41.8	**45.3**	44.5	-	**50.0**	47.8	-	**52.9**
OBQA	32.4	**40.8**	38.6	**43.8**	43.2	-	**44.2**	46.0	-	**47.0**
ARC-E	39.3	**46.0**	42.4	**51.3**	47.0	-	**56.2**	49.9	-	**60.3**
ARC-C	28.2	**29.1**	**28.6**	27.0	**31.6**	-	29.1	33.8	-	**34.4**
SST-2	67.1	**82.3**	86.2	**88.2**	85.6	77.1	**88.0**	**87.5**	82.0	86.9
SST-5	30.0	**30.9**	**39.3**	35.2	22.0	29.2	**35.2**	**40.8**	-	36.9
AGNews	**63.0**	62.2	64.4	**66.3**	**64.1**	61.8	63.8	65.4	60.0	**68.3**
TREC	**36.4**	32.2	21.6	**36.0**	44.0	30.5	**44.2**	32.8	37.3	**40.0**
BoolQ	51.1	**62.1**	49.7	**61.8**	46.7	-	**62.2**	49.5	-	**63.2**
RTE	49.8	**53.4**	**54.9**	53.8	**54.2**	-	50.2	**53.4**	48.5	49.1
CB	**50.0**	48.2	50.0	**55.4**	50.0	-	**53.6**	50.0	17.9	**66.1**

Table D.4: Performance comparison with other inference methods on GPT-2 models. PMI (Holtzman et al., 2021) is an unconditional probability normalization method, CC (Zhao et al., 2021) is the contextual calibration method and Channel (Min et al., 2021) uses an inverted-LM scoring approach that computes the conditional probability of the input given the label. We compare them in the zero-shot setting.

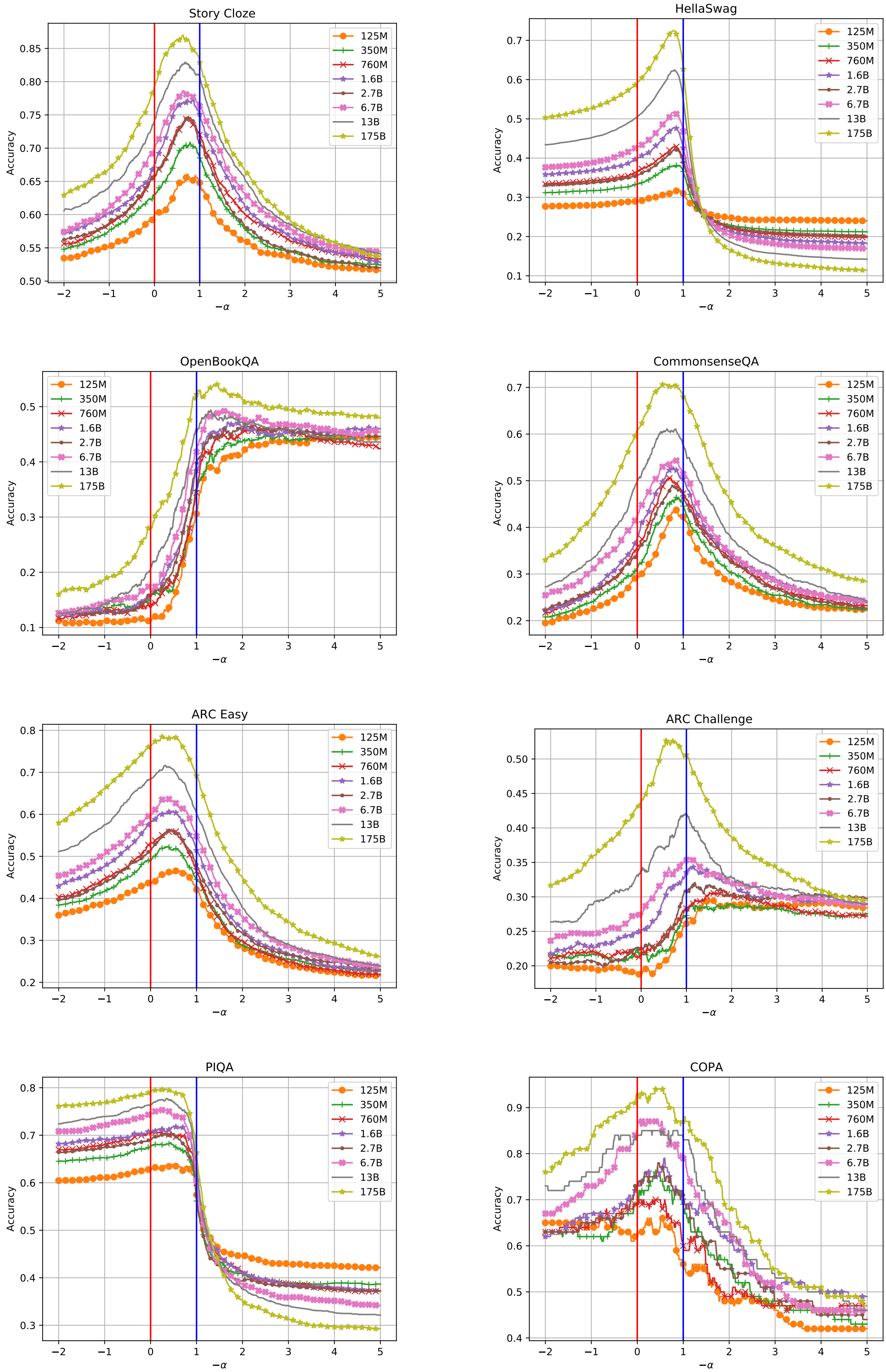

Figure D.1: Model comparison for StoryCloze, HellaSwag, OpenBookQA, CommonsenseQA, ARC Easy, ARC Challenge, PIQA and COPA by varying α on the testing set.

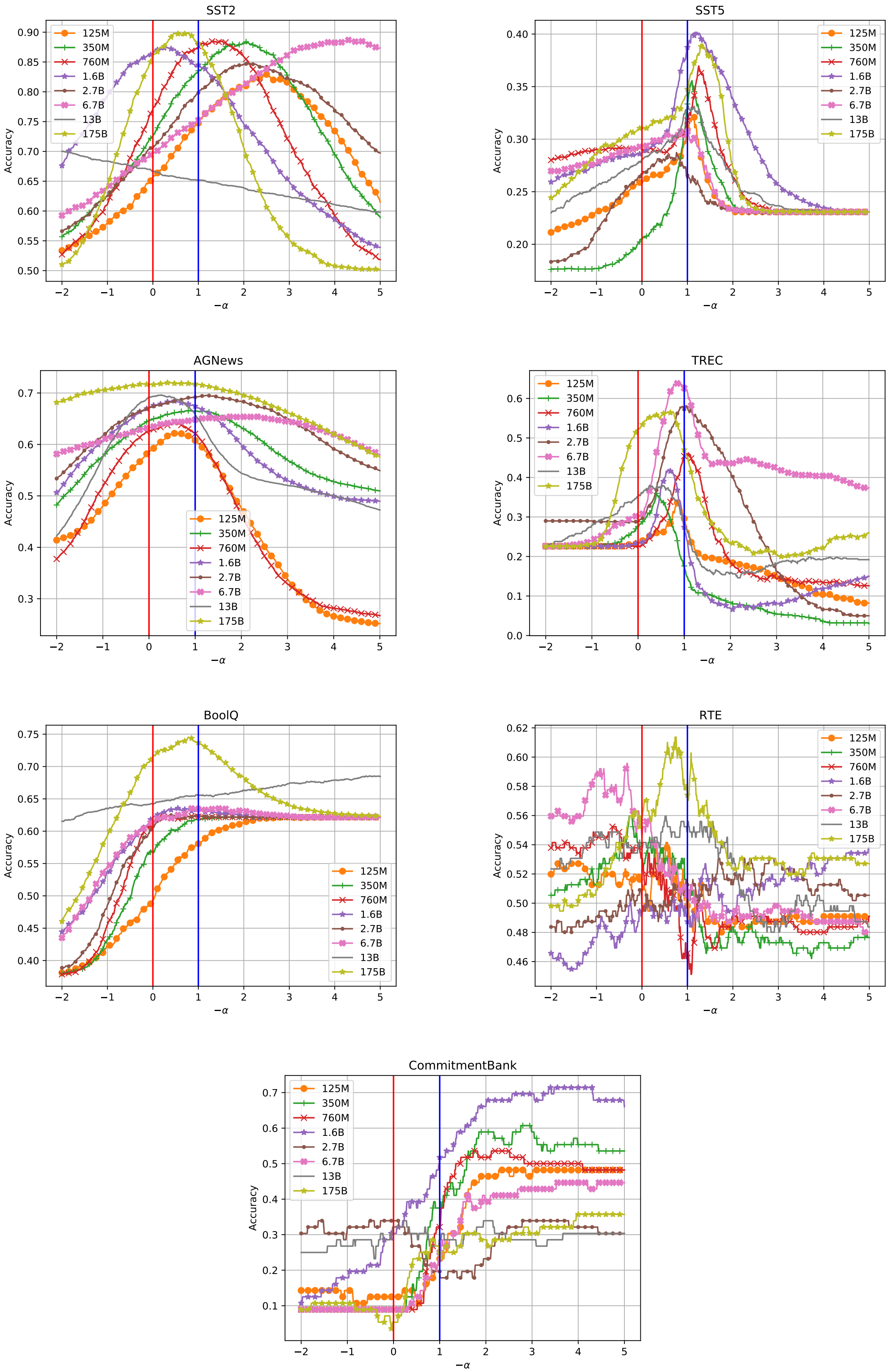

Figure D.2: Model comparison for SST-2, SST-5, AGNews, TREC, BoolQ, RTE and CommitmemtBank by varying α on the testing set.