On-the-Fly Attention Modulation for Neural Generation

Yue Dong[1*] **Chandra Bhagavatula**[2] **Ximing Lu**[2,4] **Jena D. Hwang**[2]
Antoine Bosselut[2,3] **Jackie Chi Kit Cheung**[1] **Yejin Choi**[2,4]

[1]Mila / McGill University [2]Allen Institute for Artificial Intelligence
[3] Stanford University [4]Paul G. Allen School of CSE, University of Washington
{yue.dong2@mail, jcheung@cs}.mcgill.ca
{chandrab,ximinglu,jenah,antoineb,yejinc}@allenai.org

Abstract

Despite considerable advancements with deep neural language models (LMs), neural text generation still suffers from *de*generation: the generated text is repetitive, generic, self-contradictory, and often lacks commonsense. Our analyses on sentence-level attention patterns in LMs reveal that neural degeneration may be associated with insufficient learning of task-specific characteristics by the attention mechanism. This finding motivates on-the-fly attention modulation[1] – a simple but effective method that enables the injection of priors into attention computation during inference. Automatic and human evaluation results on three text generation benchmarks demonstrate that attention modulation helps LMs generate text with enhanced fluency, creativity, and commonsense reasoning, in addition to significantly reduce sentence-level repetition.

1 Introduction

Neural text generation is critical for a wide range of downstream natural language applications. However, the standard approach – using a Transformer-based (Vaswani et al., 2017) language model (*e.g.*, Radford et al., 2019) with maximum likelihood fine-tuning and non-stochastic decoding – is known to exhibit *de*generation (Welleck et al., 2019). Despite being pre-trained on large amounts of data, text generated by neural models is observed to be repetitive, generic, self-contradictory, and lacking commonsense (Holtzman et al., 2020).

Many explanations have been proposed for neural text degeneration, including inappropriate training objectives (Welleck et al., 2019) and decoding discrepancies relative to human language (Holtzman et al., 2018, 2020). While the aforementioned may be factors for neural degeneration, we show that insufficient learning of *task-specific* characteristics – reflected in the self-attention mechanism in transformer blocks – is associated with neural text degeneration. We demonstrate that degeneration is alleviated if we inject priors through attention modulation (AttnM) during inference.

Self-attention – the ubiquitous component of Transformers – is task-agnostic with a large learning capacity for many NLP tasks (Vaswani et al., 2017; Devlin et al., 2019; Brown et al., 2020). It

Figure 1: Example of fine-tuned GPT2-L outputs without (top) and with (bottom) attention modulation on αNLG. The task is to generate a plausible explanatory hypothesis H for observations O_1 and O_2. Our proposed attention modulation injects the task-specific prior – LMs should consider both observations relative equally – through balancing the sentence-level attention weights (Eqn. 5) in Transformer blocks during inference. Applying attention modulation with the aforementioned prior make sentence-level attentions from generation to observation pairs (O_1,O_2) more balanced[2], which are reflected in the sentence-level attention heatmaps of GPT2-L (darker = lower attention) across layers (y-axis) and heads (x-axis).

*This work was done when the first author was an intern at AI2.

[2]attention ratios are normalized mean sentence-to-sentence attention from generation H to observations O_1 and O_2

learns the general characteristics of language processing through pre-training on large amounts of unlabeled data. For example, multiple analyses have suggested that attention patterns in pre-trained Transformers implicitly encode syntactic information (Raganato and Tiedemann, 2018; Michel et al., 2019; Vig and Belinkov, 2019). In sequence transduction tasks, these learned characteristics, embedded in attention, make pre-trained Transformers a powerful language model (Radford et al., 2019).

A final task-specific step is typically required for adapting a task-agnostic language model to perform the desired task[3]. However, these task-specific characteristics might not sufficiently coincide with general characteristics even after fine-tuning. For example, task-specific characteristics embedded in attention patterns – such as word alignments for machine translation – are often noisy and imperfect for generalization (Kobayashi et al., 2020).

We show that insufficient learning of task-specific characteristics, reflected in sentence-level attention patterns[4] often being out of focus, may be associated with neural text degeneration (§3). Based on this observation, we propose a simple attention modulation framework that can dynamically redistribute sentence-level attention weights by injecting task-specific priors in Transformer blocks for different downstream tasks (§4). Remarkably, in long-range narrative story generation, abductive reasoning generation and constrained commonsense text generation, both automatic and human evaluation have shown improved quality in fluency, dullness, repetition, and commonsense reasoning with attention modulation (§6).

2 Background

We briefly discuss how vanilla attention works, as well as Transformer architecture used in this paper.

Single-headed attention Given a sequence of d-dimensional input vectors $\boldsymbol{x} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$, attention mechanism computes a set of weights based on a query vector $\boldsymbol{y}_i \in \mathbb{R}^d$:

$$\text{Attn}(\boldsymbol{x}, \boldsymbol{y}_i) = (\alpha_{i,1}(\boldsymbol{x}, \boldsymbol{y}_i), \ldots, \alpha_{i,n}(\boldsymbol{x}, \boldsymbol{y}_i)) \quad (1)$$

where $\alpha_{i,j}$ is the attention weight that $\boldsymbol{y}_i$ pays to $\boldsymbol{x}_j$. One formulation of attention — scaled dot product attention — is computed as:

$$\alpha_{i,j} := \underset{\boldsymbol{x}_j \in \boldsymbol{x}}{\text{softmax}}\Big(\frac{\boldsymbol{q}(\boldsymbol{y}_i)\boldsymbol{k}(\boldsymbol{x}_j)^\top}{\sqrt{d}}\Big) \in \mathbb{R} \quad (2)$$

where query $\boldsymbol{q}(\cdot)$ and key $\boldsymbol{k}(\cdot)$ functions are linear transformations. In self attention, every $\boldsymbol{x}_i$ is used as the query vector ($\boldsymbol{y}_i$). An updated representation $\tilde{\boldsymbol{x}}_i$ is computed as a weighted sum of value vectors that are linearly transformed by $\boldsymbol{v}(\cdot)$:

$$\tilde{\boldsymbol{x}}_i = \sum_{\boldsymbol{x}_j \in \boldsymbol{x}} \alpha_{i,j} \boldsymbol{v}(\boldsymbol{x}_j). \quad (3)$$

Multi-head attention In *multi-headed* attention (MHA), N_h attention heads are computed independently to obtain the updated $\tilde{\boldsymbol{x}}_i$:

$$\tilde{\boldsymbol{x}}_i = W_o \Big\|_{h=1}^{N_h} \Big(\sum_{\boldsymbol{x}_j \in \boldsymbol{x}} \alpha_{i,j}^h \boldsymbol{v}^{\boldsymbol{h}}(\boldsymbol{x}_j) \Big). \quad (4)$$

$\alpha_{i,j}^h$ follows Eqn. 2 except the model dimension in each head h is often reduced to $d_h = \frac{d}{N_h}$. $\tilde{\boldsymbol{x}}_i$ is obtained by the concatenation of lower-rank representations from all heads and $W_o \in \mathbb{R}^{d \times h \cdot d_h}$.

GPT2-L GPT2 (Radford et al., 2019) is a family of Transformer-based language models (LMs) that follows the architecture of stacked decoder. As GPT2 follows a multi-layer and multi-headed setting, $\alpha_{i,j}$ is specific to a layer l and head h, noted as $\alpha_{i,j}^{l,h}$. We use the GPT2-L model that has 36 layers with 20 heads per layer (762M total parameters).

3 Neural text degeneration vs. attention

As researchers have sought to understand the internal mechanisms of Transformers, the attention patterns exhibited by these heads have drawn considerable study (Vig and Belinkov, 2019; Jain and Wallace, 2019; Wiegreffe and Pinter, 2019). We perform sentence-level attention analysis to explore whether aggregated attention patterns are associated with neural text degeneration.

3.1 Sentence-level attention

We first define the sentence-to-sentence attention of a language model $\mathcal{M}$ with L layers and H heads. Given two sentences p and g such that p precedes g,

[3] Brown et al. (2020) have shown that GPT3 greatly improves task-agnostic, few-shot performance, but still struggles on tasks with strong task-specific characteristics.

[4] We study the global context in the multi-sentence prompts and choose sentence-level attention (Eqn. 7) as the experiment unit, since sentences are linguistic units of complete meaning.

the mean $\bar{\alpha}^{l,h}_{g,p}$ and max $\hat{\alpha}^{l,h}_{g,p}$ sentence-to-sentence attentions from g to p for layer l and head h are:

$$\bar{\alpha}^{l,h}_{g,p} = \frac{\sum_{i=1}^{|g|}\sum_{j=1}^{|p|} \alpha^{l,h}_{i,j}(g_i, p_j)}{|g| \cdot |p|} \quad (5)$$

$$\hat{\alpha}^{l,h}_{g,p} = \max_{\substack{i\in\{1,\ldots,|g|\}\\ j\in\{1,\ldots,|p|\}}} \alpha^{l,h}_{i,j}(g_i, p_j). \quad (6)$$

The aggregated sentence-to-sentence attention over the Transformer architecture $\mathcal{M}$ is defined as:

$$\alpha^{\mathcal{M}}_{g,p} = \frac{\sum_{l=1}^{L}\sum_{h=1}^{H} \alpha^{l,h}_{g,p}}{L \cdot H} \quad (7)$$

where $\alpha \in \{\bar{\alpha}, \hat{\alpha}\}$ computes either the mean or the max sentence-level attention over $\mathcal{M}$.

3.2 Is neural text degeneration related to attention patterns?

We conduct experiments to evaluate whether neural text degeneration is associated with sentence-level attention patterns. Empirical results on two types of neural degeneration that are easy to detect – repetition and lacking commonsense reasoning under constraints – reveal their association.

Repetition vs. attention One common form of neural text degeneration is sentence-level repetition (Welleck et al., 2019). This type of degeneration happens frequently in our experiment on ROCStories test set (§6.1): given a five-sentence prompt, 35.4% of the consecutive sentences from the next five greedily generated sentences by the fine-tuned GPT2-L are exact repetitions. We check whether sentence-level attention patterns behave differently when generating repeated or different consecutive sentences.

We inspect the attention behavior by measuring the change of sentence-level attention when generating two consecutive sentences. The generations of fine-tuned GPT2-L on ROCStories test set are separated into two subsets $\{\mathcal{D}_{\text{repeated}}, \mathcal{D}_{\text{different}}\}$: in which consecutive sentences that are either repeated (*i.e.*, degenerate) or different. Given the fine-tuned GPT2-L language model $\mathcal{M}$, we measure the sentence-level attention change Δ on the prompt sentence $p_{j\in\{1,\ldots,5\}}$ while generating the consecutive sentence pair $(g_i, g_{i+1}), i \in \{1, \ldots, 4\}$, aggregated over the subset $\mathcal{D} \in \{\mathcal{D}_{\text{different}}, \mathcal{D}_{\text{repeated}}\}$:

$$\Delta(j, \mathcal{D}) = \sum_{d\in\mathcal{D}} |\bar{\alpha}^{\mathcal{M}}_{g^d_{i+1}, p^d_j} - \bar{\alpha}^{\mathcal{M}}_{g^d_i, p^d_j}|/|\mathcal{D}| \quad (8)$$

where $\bar{\alpha}^{\mathcal{M}}_{g_i,p_j}$ is the mean sentence-level attention from sentence g_i to sentence p_j defined in Eqn. 7.

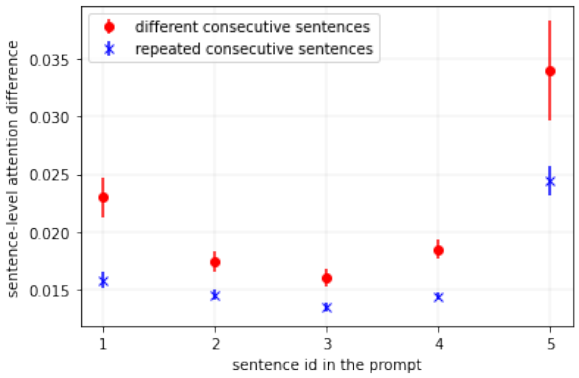

Figure 2: Mean sentence-level attention change of GPT2-L on ROCStories test set, while generating different (red) or repeated (blue) consecutive sentences.

Figure 2 plots the aggregated mean sentence-level attention change over prompt sentences when GPT2-L generates repeated (red) or different (blue) consecutive sentences. The sentence-level attention changes are vastly lower on all prompt sentences when generating repeated consecutive sentences. Thus, sentence-level repetition may be correlated with the insufficient change of sentence-level attention. In §4 and §6, we show that generation quality can be vastly improved by injecting the prior – attention should look at the prompt differently when generating different sentences – through our proposed attention modulation.

Lack of commonsense reasoning vs. attention Text generated by neural language models is also observed to be lacking commonsense reasoning (Mao et al., 2019). We check whether this type of neural degeneration is associated with attention patterns. A benchmark dataset for generative commonsense reasoning – CommonGen (Lin et al., 2020) – is used as our test bed. CommonGen is designed for constrained commonsense reasoning: given a set of common concepts (e.g., use, tool, piece, metal); the task is to generate a coherent and plausible sentence covering all these concepts (e.g., "a piece of metal is used for making tools"). Covering the concepts in generation requires relational reasoning with background commonsense knowledge.

	agg. max attn.	SD	#
covered	0.434	0.0040	4515
uncovered	0.376	0.0068	1473

Table 1: Aggregated max sentence-level attention of the fine-tuned GPT2-L; the results are aggregated from the generation to covered or uncovered concepts on the CommonGen test set. agg. max attn., SD, # refer to aggregated max sentence-level attention, standard deviation, and the number of instances.

Each concept is represented as a prompt sentence in our experiments.[5] During generation, a concept (*e.g.*swim) is covered if its reflected form (*e.g.*{swim, swimming, swam, swum}) is generated in the CommonGen test set. We use a fine-tuned GPT2-L for the generation. Among the 5988 concepts in the prompt, about 75% of them are covered in the generation of GPT2-L. We can then easily separate sentence-level attention from the generation to the concept into two subsets: concept in the prompt that is covered or uncovered by the generated sentence.

Table 1 shows the results of max sentence-level attention (Eqn. 7) of the finetuned GPT2-L on the CommonGen test set[6]. We can observe that sentence-level attention from the generation to the concept is vastly higher when the concept is covered. Compared to that of uncovered concepts, the aggregated max sentence-level attention is 15.4% higher. Therefore, failing to generate a common concept through reasoning may be associated with insufficient attention to the concept.

In both cases, neural text degeneration is associated with insufficient attention to elements that are important for downstream generations. This motivates us to explore whether we can inject these priors in the language model by altering the attention mechanism to alleviate degeneration.

4 Method

This section describes our method – attention modulation – that can alleviate neural text degeneration. In §4.1, we describe the general attention modulation framework. In §4.2, §4.3, and §4.4, we discuss the priors injected through attention modulation for three different tasks: narrative story generation, abductive reasoning generation, and constrained commonsense reasoning.

4.1 Attention Modulation

Attention modulation aims to change the attention weights of a Transformer-based language model during inference, so that the generation can reflect priors that alleviate neural text degeneration. This additional signal is added to the self-attention computation in the Transformer blocks.

We reformulate the attention computation of Eqn. 2 by adding an attention reweighting function f, where priors can be injected. Given a sequence of input tokens $\boldsymbol{x}$, the self-attention from $\boldsymbol{x}_i$ to $\boldsymbol{x}_j$ ($i \geq j$) while generating the t-token is reformulated to:

$$\alpha^t_{i,j} := \underset{\boldsymbol{x}_j \in \boldsymbol{x}}{\text{softmax}}\Big(\frac{\boldsymbol{q}(\boldsymbol{x}_i)\boldsymbol{k}(\boldsymbol{x}_j)^\top}{\sqrt{d}} + f_{i,j}(\boldsymbol{x}, \boldsymbol{\alpha}^{t-1})\Big) \tag{9}$$

where $f(\boldsymbol{x}, \boldsymbol{\alpha}^{t-1})$ is the attention reweighting function and $\boldsymbol{\alpha}^{t-1}$ is the attention weight matrix for all layers and heads in the Transformer architecture at time step $t-1$. The attention reweighting function f can be either pre-defined or learned. In our experiments, we inject pre-defined sentence-level priors (heuristics) through f and show that this injection alleviates neural text degeneration. We leave the learning of better reweighting functions automatically to future work.

In the following sections, we describe sentence-level attention reweighting functions that are used for three different text generation tasks.

4.2 ROCStories: narrative generation

As shown in §3, sentence repetition in long-form generation may be associated with insufficient attention change while generating consecutive sentences. To amplify the attention changes, we can redistribute sentence-level attention with some priors while generating consecutive sentences.

We choose the prior that language model should consider long-range context during generation, as we observed that attention mostly focuses on the near history in many cases (Appendix A.1). Note this prior also increases the sentence-level attention change while generating consecutive sentences: the sentence-level attention for all previous sentences is always re-balanced based on the newly-generated sentence. To balance the attention of tokens in

[5] We can obtain a clear boundary for each concept with this design choice of adding a period as the separator, as concepts can be tokenized into multiple subwords by GPT2 tokenizers.

[6] We measure the max sentence-level attention rather than the mean sentence-level attention on CommonGen, as the attention to a concept is usually reduced once it is generated.

each sentence received while generating the next sentence, we define the attention reweight function in Eqn. 9 with the aforementioned prior for ROCStories as:

$$f_{i,j}(\boldsymbol{x}, \boldsymbol{\alpha}_{i,j}^{t-1}) = \frac{1}{\alpha_{g_i,p_j}^{t-1}}. \quad (10)$$

As later sentences in the prompt usually receive larger sentence-level attention weights (Appendix A.1), attention reweighting function defined in Eqn. 10 will add a large weight to tokens in the early sentences and a small weight to tokens in the late context sentences. The simple heuristic of balancing context sentences to be considered relatively equal, namely more weights on early context sentences, might not be optimal prior. However, it improves the long-form story generation in multiple measures, including fluency, interesting, newness, and repetition (§6.1).

4.3 αNLG: abductive reasoning generation

The second benchmark dataset we tested with attention modulation is αNLG (Bhagavatula et al., 2020). This dataset is proposed for abductive reasoning generation: given two observations O_1 and O_2, the model needs to generate a valid hypothesis h that explains what happened between the two observations. For example, given O_1: "Today was the first day of school." and O_2: "I hate school.", the task is to generate h such as "The teacher made fun of me." as a plausible explanation.

Bhagavatula et al. (2020) has shown that the fine-tuned GPT2 performs far below human performance on the αNLG task. We hypothesis that this may be associated with insufficient learning of sentence-level attention to both observations; for example, the model might over-fit to one of the observations for generation. Thus, we inject the prior – the language model should consider both observations relative equally – while generating a plausible explanation. This prior can be injected with attention reweighting function defined in Eqn. 10.

4.4 CommonGen: constrained commonsense generation

The third benchmark is CommonGen – a constrained text generation challenge for generative commonsense reasoning. CommonGen requires machines to generate a realistic sentence using *all* concepts from a given concept set by conducting commonsense reasoning over the relations among the given concepts. To successfully generate a plausible and grammatical sentence that follows the commonsense, models need to conduct commonsense reasoning over the relations among the given concepts. Our experiment in §3.2 shows that the fine-tuned GPT2-L can only cover about 75% of concepts during generation. We infer from Table 1 that this may be associated with GPT2-L giving insufficient sentence-level attention to uncovered concepts. Thus, we propose a simple heuristic – model should pay more attention to concepts that are not covered yet – to be injected with attention modulation.

Consider the prompt with m concepts $\boldsymbol{c} = \{c_1, \ldots, c_m\}$ and a partially generated sentence $y_1, \ldots, y_{t-1}$. While generating the t-th token, the sentence-level reweighting function from the i-th token to the j-th token in c_k is defined as:

$$f_{i,j}(\boldsymbol{x}) = \begin{cases} 1/m & \text{if} \quad c_k \subset y_{1:t-1} \\ 1 & \text{else} \end{cases} \quad (11)$$

Intuitively, if a concept c_k is covered in the partial generation, attention modulation with Eqn. 11 will reduce the attention weights of the tokens in concept c_k.

5 Experimental Setups

This section describes the experiment setups, including the baselines, decoding algorithms, datasets, and evaluation metrics.

Model architecture & baseline Attention modulation is architecture-agnostic and can be applied to any Transformer-based models that contain self-attention computation. We choose GPT2-L (Radford et al., 2019) for our experiments, which has achieved state-of-the-art performance on a variety of generation tasks (Vig and Belinkov, 2019). Attention modulation can be applied to any range of layers in the Transformer. To compare models with and without attention modulation on each of the three generation tasks, we use the best fine-tuned GPT2-L based on the validation set after fine-tuning for 4 epochs with the default settings.

Decoding Attention modulation directly changes the attention weights of the context tokens during inference. It is orthogonal to different decoding algorithms that change the searching strategies based on the softmax distribution emitted by Transform-

	next 1 sent.		next 2 sent.		next 3 sent.		next 4 sent.		next 5 sent.		
	uniq.	**% rel.**	**uniq.**	**% rel.**	**uniq.**	**% rel.**	**uniq.**	**% rel.**	**uniq.↑**	**% rel.↑**	**% rep.↓**
w/o AttnM	9.08k	48.59	13.58k	48.78	15.71k	49.20	16.01k	49.97	17.03k	50.52	35.43
w/ AttnM (ours)	**10.39k**	**52.92**	**15.12k**	**54.23**	**18.33k**	**55.35**	**20.41k**	**55.78**	**22.69k**	**55.78**	**17.49**
ratio	1.14		1.12		1.16		1.24		1.34		

Table 2: Test results of the fine-tuned GPT2-L w/ and w/o attention modulation on ROCstories with the greedy decoding algorithm. uniq. represent the unique number of tokens generated in the whole test corpus, which measures the number of new unique tokens generated. rel. represent relevancy, which measures the percentage of tokens generated appears in the prompt. rep. measures the sentence-level repetition – whether two sentences generated are identical.

ers.[7] We present the results with non-stochastic decoding algorithms (*i.e.* greedy decoding and beam search), as generations based on them truly reflect the token-level probabilities predicted by the model (Holtzman et al., 2018).

Datasets We use three different generation datasets – ROCStories (Mostafazadeh et al., 2016), αNLG (Bhagavatula et al., 2020), and CommonGen (Lin et al., 2020). For ROCStories, we used the 2017 version and split the data into 75/10/15 for train/val/test.

Evaluation On ROCStories, we measure dullness, relevancy and repetition similar to Welleck et al. (2019). We report the number of unique tokens generated, where the generation is less dull if more unique tokens are generated. For repetition, we directly measure sentence-level repetition: two generated sentences are repeated if their strings are the same. For relevancy, we measure the percentage of generated tokens that appear in the prompt. Besides, we perform a human evaluation, where three annotators are asked to rate the generations based on fluency, interestingness, newness, relevancy, and repetition.

On αNLG, we score the generated explanation with respect to the reference using the following automatic metrics: BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), METEOR (Banerjee and Lavie, 2005), and CIDEr (Vedantam et al., 2015). In addition, we ask annotators to compare the generated explanations without and with attention modulation. Human judges are asked to decide which system provides a more plausible explanation of the observations.

On CommonGen, we report SPICE (Anderson et al., 2016) – a measure that evaluates semantic propositional content, in addtion to BLEU, ROUGE, METEOR, CIDEr. We also report Coverage (Lin et al., 2020), which computes the average percentage of input concepts that appear in the lemmatized outputs. We conduct a human evaluation following the protocol of Lu et al. (2020). Human judges are asked to compare two systems in terms of fluency, coverage (covers the concept), and overall quality (covers the concepts and follows commonsense).

[7] These search-based decoding algorithms do not resolve the poorly generated token-level probabilities.

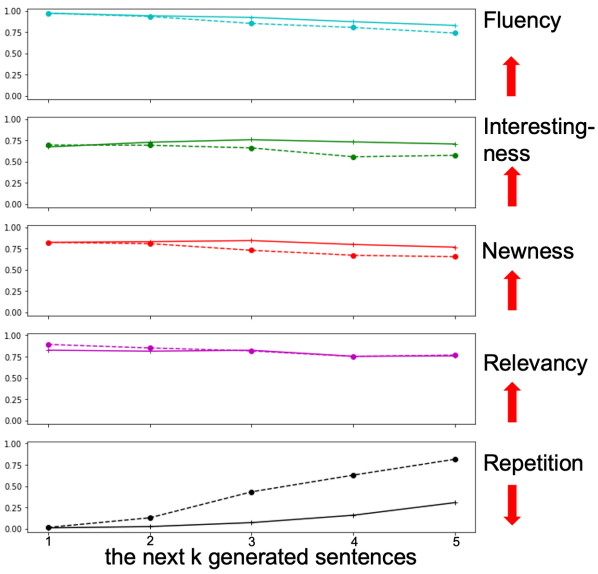

Figure 3: Human evaluation results on the next 1 to 5 sentences generated without (dashed lines) and with (solid lines) attention modulation (1000 samples).

6 Result

In this section, we present the vast improvements of the fine-tuned GPT2-L with attention modulation on three narrative generation and generative reasoning tasks: ROCStories, αNLG, and CommonGen.

6.1 ROCstories

Table 2 indicates that attention modulation significantly reduces repetition in narrative generation,

		R-2	R-L	B-3	B-4	Meteor	CIDEr	SPICE	Coverage
greedy	w/o AttnM	14.06	34.13	26.19	17.92	25.82	10.81	22.14	76.50
	w/ AttnM (ours)	**14.71**	**35.15**	**27.53**	**18.91**	**26.23**	**11.61**	**23.22**	**79.18**
beam=5	w/o AttnM	16.68	36.92	32.39	23.36	26.87	12.24	22.83	76.99
	w/ AttnM (ours)	**17.14**	**38.23**	**33.92**	**24.03**	**27.48**	**12.88**	**24.24**	**81.27**
beam=10	w/o AttnM	17.25	37.37	33.81	24.39	27.51	12.58	23.24	78.68
	w/ AttnM (ours)	**17.59**	**38.71**	**35.72**	**25.93**	**27.71**	**13.32**	**24.36**	**81.24**
beam=20	Lin et al. (2020)	16.85	39.01	33.92	23.73	26.83	12.19	23.57	79.09
	w/o AttnM	17.98	38.07	35.14	25.61	27.63	12.90	23.28	79.62
	w/ AttnM (ours)	**18.11**	**39.32**	**36.69**	**26.80**	**28.02**	**13.71**	**23.94**	**81.85**

Table 3: CommonGen test results of the fine-tuned GPT2-L w/ or w/o attention modulation based on different decoding algorithms.

	B-4	R-L	Meteor	CIDEr	Human
w/o AttnM	13.51	**18.29**	13.18	47.69	14%
w/ AttnM (ours)	**13.52**	18.01	13.18	**48.20**	**33%**

Table 4: Evaluations of the fine-tune GPT2-L on αNLG using greedy decoding.

while increasing the relevancy of generated sentences to the original story. We can observe a vast improvement in the number of unique generated tokens using attention modulation, indicating a reduced repetition rate (confirmed by the % number of repeated sentences in the next five sentence generated – 35.43 vs. 17.49 for our approach). This intuition is confirmed by our human evaluation in Figure 3, where the GPT2-L with attention modulation produces sentences that are more fluent, more interesting, more novel, and less repetitive than the original decoder. Furthermore, we note that the difference in performance across these evaluation categories generally increases as the number of generated sentences increases, indicating less sensitivity to long-form degeneration.

6.2 αNLG

Table 4 presents the automatic and human evaluation results on αNLG. We can see that our model performs similarly with and without attention modulation in terms of automatic evaluation. However, our human evaluation results in the last column show that overall, the human judges prefer the explanations produced using attention modulation significantly more than those of the original model. With 100 samples generated, 33% of the time, human judges select explanations generated with attention modulation as more plausible. In contrast, explanations from the original model are only preferred 14% of the time.

6.3 CommonGen

Table 3 shows the automatic evaluation results on the CommonGen dataset. We separate different settings of decoding algorithms in blocks. By injecting the prior – the model should put more attention on uncovered concepts – into the GPT2-L with attention modulation, we can improve the text generated in every automatic measure significantly. Interestingly, despite our attention-reweighted decoder only encouraging coverage, we see all the other measures such as ROUGE, BLEU, METEOR, CIDEr, SPICE improve, as well.

These improvements also hold when we use a different base decoding algorithm, such as beam search. Again, the performance improvement for using attention modulation is significant over all measures. Thus, unlike decoding algorithms that improve downstream tasks through truncation of the sampling distribution, we directly re-calibrate the token-level probabilities predicted by the model by altering attention patterns in the Transformer blocks during inference.

We also conduct a human evaluation to check whether this improvement in the automatic metrics transfers to human judgments. In Table 6, we see that our attention modulation algorithm significantly outperforms the original inference model on every measure – from fluency, quality, and overall performance.

6.4 Vast improvements on few-shot learning

Table 5 presents the results of attention modulation on GPT2-L that are fine-tuned on different numbers of training examples from CommonGen. We observe the improvements on all measures are

Training size	Method	R-2	R-L	B-3	B-4	Meteor	CIDEr	SPICE	Coverage	
10	w/o AttnM	2.15	16.07	6.39	3.63	10.02	1.23	5.07	27.12	
	w/ AttnM (ours)	**3.38**	**18.89**	**7.24**	**3.48**	**12.47**	**1.63**	**7.21**	**36.55**	↑ 9.43
1000	w/o AttnM	7.61	27.03	14.01	7.38	20.67	6.40	16.79	62.33	
	w/ AttnM (ours)	**8.54**	**27.97**	**15.51**	**8.78**	**22.13**	**6.68**	**18.02**	**69.31**	↑ 6.98
10000	w/o AttnM	10.70	30.06	17.40	10.02	23.40	7.15	20.20	73.39	
	w/ AttnM (ours)	**11.53**	**30.70**	**18.43**	**11.12**	**24.42**	**7.29**	**21.33**	**78.74**	↑ 5.35
full (∼39k)	w/o AttnM	14.06	34.13	26.19	17.92	25.82	10.81	22.14	76.50	
	w/ AttnM (ours)	**14.71**	**35.15**	**27.53**	**18.91**	**26.23**	**11.61**	**23.22**	**79.18**	↑ 2.68

Table 5: CommonGen test results of the fine-tuned GPT2-L w/ or w/o attention modulation trained on different size of training examples (greedy decoding).

	Fluency	Quality	Overall
w/o AttnM	85.07	39.30	44.77
w/ AttnM (ours)	**89.55**	**48.76**	**52.73**

Table 6: Human evaluations of the fine-tuned GP2-L w/o or w/ attention modulation on CommonGen.

more prominent when the fine-tuning data size is small. For example, adding attention modulation can improve coverage by 9.43% on the GPT2-L fine-tuned with only 10 examples. This not only validates that priors we injected into the model are suitable for improving the downstream task performance, but also shed lights to use attention modulation on different few-shot learning scenarios where the number of training examples is limited.

7 Related Work

We propose to use attention modulation to heuristically re-balance sentence-level attention for neural text degeneration. At least three domains of work are closely related to our proposal, namely, attention pattern analysis, work that focuses on changing or approximating learned attention patterns, and work for countering neural text degeneration.

Attention analysis: Previous work has investigated the attention patterns within the local context of a sentence. These works highlighted that attention patterns in Transformers implicitly encode syntactic information such as dependency relations (Htut et al., 2019), and part-of-speech tags (Vig and Belinkov, 2019; Raganato and Tiedemann, 2018). Other works observed that attention patterns can provide explanations (Wiegreffe and Pinter, 2019) or coarse word alignments in machine translation (Zenkel et al., 2019; Kobayashi et al., 2020). In contrast to these works, we analyze sentence-level attention patterns for neural text degeneration, and propose to directly modify the attention computation to reduce it.

Alternative attention: Many works have been proposed to change attention mechanisms to optimize their O(n^2) complexity. Some promising directions in this space include sparse attention mechanisms (Beltagy et al., 2020; Zaheer et al., 2020) and linearized attention (Choromanski et al., 2021). These alternative attention mechanisms require training the model and are used as replacements to the original attention mechanism for fast training or reduced computation. Our work is fundamentally different as we seek to inject priors into the standard attention mechanism during inference (without re-training the model).

Neural text degeneration: Previous works seek to solve neural text degeneration by changing the training objective to reduce the likelihood of common tokens (Welleck et al., 2019), or modifying the decoding algorithm by truncating the sampling distribution (Holtzman et al., 2018, 2020). Specifically, Welleck et al. (2019) introduce an additional training loss that reduces the likelihood of common tokens. Holtzman et al. (2018, 2020) propose stochastic decoding algorithms with truncation of the sampling distribution. Our work is orthogonal to these methods by injecting priors into the model's attention computation during inference.

8 Conclusions and future work

Neural language models often exhibit degeneration: the output texts are repeated, bland, and inconsistent. Our empirical analyses show that neural text degeneration may be associated with insufficient

learning of task-specific characteristics by the attention mechanism. We propose a simple but effective module – attention modulation – that can inject priors for better generation through re-balancing the attention weights during inference. Results on three different narrative and commonsense generation tasks indicate that attention modulation can reduce repetition and enhance commonsense reasoning while maintaining fluency and coherence.

Acknowledgements

This research was supported in part by NSF (IIS-1714566), DARPA MCS program through NIWC Pacific (N66001-19-2-4031), the Canada CIFAR AI Chair program, the Natural Sciences and Engineering Research Council of Canada (NSERC), Intel Labs Cognitive Computing Research, and the Allen Institute for AI. Computations on beaker.org were supported in part by credits from Google Cloud.

References

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. Spice: Semantic propositional image caption evaluation. In *European Conference on Computer Vision*, pages 382–398. Springer.

S. Banerjee and A. Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *IEEvaluation@ACL*.

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

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Wen-tau Yih, and Yejin Choi. 2020. Abductive commonsense reasoning. In *International Conference on Learn-ing Representations*.

T. Brown, B. Mann, Nick Ryder, Melanie Subbiah, J. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, G. Krüger, T. Henighan, R. Child, Aditya Ramesh, D. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, E. Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, J. Clark, Christopher Berner, Sam McCandlish, A. Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. 2021. Rethinking attention with performers.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Ari Holtzman, Jan Buys, Maxwell Forbes, Antoine Bosselut, David Golub, and Yejin Choi. 2018. Learning to write with cooperative discriminators. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1638–1649, Melbourne, Australia. Association for Computational Linguistics.

Phu Mon Htut, Jason Phang, Shikha Bordia, and Samuel R Bowman. 2019. Do attention heads in bert track syntactic dependencies? *arXiv preprint arXiv:1911.12246*.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. 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 3543–3556, Minneapolis, Minnesota. 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.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2020. Attention is not only a weight: Analyzing transformers with vector norms. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7057–7075, Online. Association for Computational Linguistics.

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2020. CommonGen: A constrained text generation challenge for generative commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1823–1840, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Annual Meeting of the As-sociation for Computational Linguistics (ACL)*.

Ximing Lu, Peter West, Rowan Zellers, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Neurologic decoding:(un) supervised neural text generation with predicate logic constraints. *arXiv preprint arXiv:2010.12884*.

Huanru Henry Mao, Bodhisattwa Prasad Majumder, Julian McAuley, and Garrison Cottrell. 2019. Improving neural story generation by targeted common sense grounding. 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 5990–5995.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems*, volume 32, pages 14014–14024. Curran Associates, Inc.

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.

Kishore Papineni, S. Roukos, T. Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Annual Meeting of the As-sociation for Computational Linguistics (ACL)*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog 1.8*.

Alessandro Raganato and Jörg Tiedemann. 2018. An analysis of encoder representations in transformer-based machine translation. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 287–297, Brussels, Belgium. Association for Computational Linguistics.

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

Ramakrishna Vedantam, C. L. Zitnick, and D. Parikh. 2015. Cider: Consensus-based image description evaluation. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4566–4575.

Jesse Vig and Yonatan Belinkov. 2019. Analyzing the structure of attention in a transformer language model. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 63–76, Florence, Italy. Association for Computational Linguistics.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2019. Neural text generation with unlikelihood training. In *International Conference on Learning Representations*.

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. 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 11–20, Hong Kong, China. Association for Computational Linguistics.

Manzil Zaheer, Guru Guruganesh, 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. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Thomas Zenkel, Joern Wuebker, and John DeNero. 2019. Adding interpretable attention to neural translation models improves word alignment. *arXiv preprint arXiv:1901.11359*.

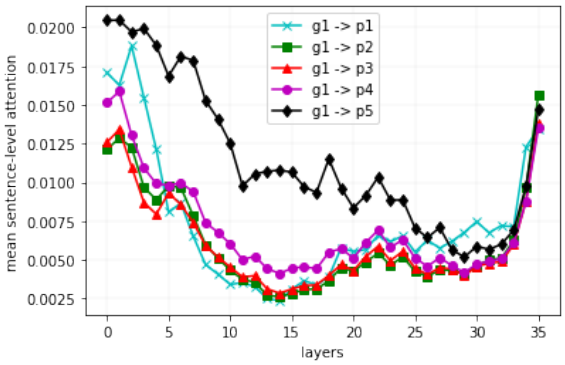

Figure 4: Sentence-level attention distribution across different layers in GPT2-L. The result is aggregated by computing the mean sentence-level attention from the next generated sentence to the five sentences in the prompt of ROCStories development set. Lower number represents lower layers in the Transformer.

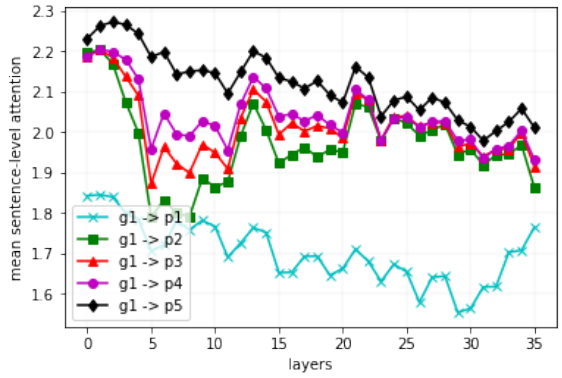

Figure 5: Attention entropy of each sentence in the prompt aggregated over the ROCStories dev. set. In the first sentences, there are particularly high-entropy attention heads that produce bag-of-vector-like representations.

A Appendices

A.1 How does a language model use attention to model a multi-sentence prompt?

Sentence-level attention portion To reveal which part of context – near or distant history – are important for context representation, we compute aggregated mean sentence-level attention (Eqn.5 in the main text) each prompt sentence p_i received, while generating the sentence g_1 after the prompt. We observe from Figure 4 that GPT2-L mostly attends to the nearest sentence (p_5) during the generation. This effect is especially prominent in the early and middle layers. In the late layers, the attention from different sentences evens out. This observation is consistent with previous analysis of attention patterns within sentences such that deeper layers focus on longer-range context (Vig and Belinkov, 2019).

	train	dev.	test
ROCStories	39498	5269	7899
αNLG	169,654	1,532	3,059
CommonGen	67,389	4,018	7,644

Table 7: Dataset Statistics

Sentence-level attention entropy Khandelwal et al. (2018) observed that LSTM represent distant context as topics; only a few token in the distant context are used to compute the context representation. We check whether this observation also holds on Transformer-based models by computing attention entropy. This sentence-level attention entropy of p_i based on the attention from g_1 to p_i at layer l_m over a corpus X is defined as:

$$E_A(g_1, p_i, l_m) = -\frac{\sum_{x \in X} \sum_{h \in l_m} \sum_{j=1}^{|p_i|} \sum_{k=1}^{|g_1|} \alpha_{j,k}^h \log(\alpha_{j,k}^h)}{|X| \cdot |H| \cdot |p_i| \cdot |g_1|} \quad (12)$$

where h is a head in layer l_m and $\alpha_{j,k}^h$ is the attention weight from $x_j \in p_i$ to $x_k \in g_1$ for h. Figure 5 shows a clear separation of entropy over different sentences in the prompt, where more distant sentences have lower entropy values. This suggests that LMs only modelling distant sentences as topics – attention over key words being a proxy.

A.2 Hyperparameters

Attention modulation can be applied to any layer and any head in the Transformer based on our implementation. However, the weights learned by different heads in a particular layer have a large variance (Vig and Belinkov, 2019) and are subject to change from different training sessions. Therefore, we only reweight attention on all heads in different layers, where what layers are re-weighted are hyperparameters. We choose to reweight the consecutive layers from a starting layer l_s to an end layer l_e and performed a grid search on different layer ranges. For the start layer, we experimented with $l_s \in \{0, 4, 8, 12, 16, 20, 24, 28, 32\}$; for the end layer, we experimented with $l_e \in \{4, 8, 12, 16, 20, 24, 28, 32, 36\}$. The reweighting layers are chosen based on the validation set performance. On ROCstories, the GPT2-L are reweighted with $l_s = 8$ and $l_e = 32$; On αNLG, the

propmt	attention reweight	Generated sentence with attention-decoding
field. stand. look. =	(1,3,2)	A man stands looking at a sign in a field.
field. stand. look. =	(1,2,3)	He looks up and sees a group of people standing in the field.
field. stand. look. =	(2,3,1)	He stands in the middle of the field, looking down at the stands.

Table 8: An example of attention modulation with different attention reweighting functions on CommonGen dev set. Only by redistributing the sentence-level attention during inference, we can generate sentences following the desired order specified in the attention reweighting function.

		R-2	R-L	B-3	B-4	Meteor	CIDEr	SPICE	Coverage
beam=20	w/o AttnM	17.98	38.07	35.14	25.61	27.63	12.90	23.28	79.62
	w/ AttnM	18.11	39.32	36.69	26.80	28.02	13.71	23.94	81.85
	w/ AttnM$_{pm}$	**19.78**	**41.17**	**38.28**	**28.36**	**30.97**	**15.38**	**28.24**	**91.89**

Table 9: Results on CommonGen test set with beam search and permutations defined in A.3.

GPT2-L are re-weighted with $l_s = 12$ and $l_e = 32$; On CommonGen, the GPT2-L are re-weighted with $l_s = 24$ and $l_e = 32$.

A.3 Attention modulation and generation order

CommonGen provides the concept set in a random order, where models need to perform a relational commonsense reasoning to find the optimal order of them for generating a plausible sentence. We found that attention modulation provide signals for generation order given different reweighting functions (examples in Table 8). In this experiment, we guide the generation order by providing different initialization weights in the reweighting functions. We enforce different attention modulation weights based on the order we want the concepts to be generated. For examples, row 1 in table 8 means the concepts of (FIELD, STAND, LOOK) are initialized to be re-weighted by scales of (1,3,2).

This interesting finding motivates us to conduct a permutation experiment on CommonGen. For a k concept set, we initialize the attention modulation weights based on the permutations of 1 to k ($k!$ permutations in total) and generate $k!$ sentences with attention modulation. We then select the generation that covers the most concepts[8] from these $k!$ generations as output. We call this method "attention modulation with permutation". Table 9 presents the results of attention modulation with permutation. We see that just by enforcing the order in attention modulation, the coverage can be improved by another 10%.

A.4 Human evaluation details

Figure 6 and 7 show the evaluation templates for tuckers for Rocstories and CommonGen[9], respectively. On ROCstories, the inter-annotator agreements is 0.743 and fleiss kappa is 0.229 averaged over the five dimensions on 200×5 samples. The inter-annotator agreements are 84.6%, 72.2%, 76.1%, 78.5%, and 59.9% for fluency, interestingness, newness, relevance, and repetition. On αNLG, the human evaluation results are 33%, 14%, 10%, 32%, 11% for preferring outputs with attention modulation, without attention modulation, both, neither and no agreement. The inter-annotator agreements is 0.835 and kappa is 0.439 over 100 samples. On CommonGen, the inter-annotator agreements are 77.94%, 71.64%, and 67.50% for fluency, quality, and overall on 200 samples.

[8] If there is a tie, we choose the shorter generation.

[9] omitted αNLG template due to space limitation.

prompt	Jen was always afraid of flying. She needed to fly to Japan for an important meeting. Jen took some Dramamine and boarded her plane. Jen dozed off shortly after the plane took off.
w/o AttnM	Jen woke up to a huge headache. She had to cancel her trip to Japan. Jen was glad she had taken Dramamine. She was able to fly safely. Jen was glad she had taken Dramamine.
w/ AttnM	She woke up to a loud bang and felt sick. She had to sit in the back of the plane. She was so scared she couldn't even get out of her seat. Jen was so embarrassed she never flew again. She was so scared she never flew again.

Table 10: A generation example (greedy decoding) without and with attention modulation on ROCStories. Adding attention modulation during inference **improves the commonsense reasoning and consistency of the generation**.

prompt	Neil had been journeying through Asia. But he contracted malaria on a Thai island, and had to be flown home for treatment.
w/o AttnM	He was so sick that he couldn't go to the hospital. He was so sick that he couldn't go to the hospital. He was so sick that he couldn't go to the hospital. He was so sick that he couldn't go to the hospital. He was so sick that he couldn't go to the hospital.
w/ AttnM	He was very sick and had to be hospitalized. He was in the hospital for a week. He recovered and was released. Neil was very happy to be home. He was able to get better and was able to go back to his home country.

Table 11: Example (greedy decoding) of the fine-tuned GPT2-L without and with attention modulation on story completion. Adding attention modulation during inference significantly **reduces the sentence-level repetition**.

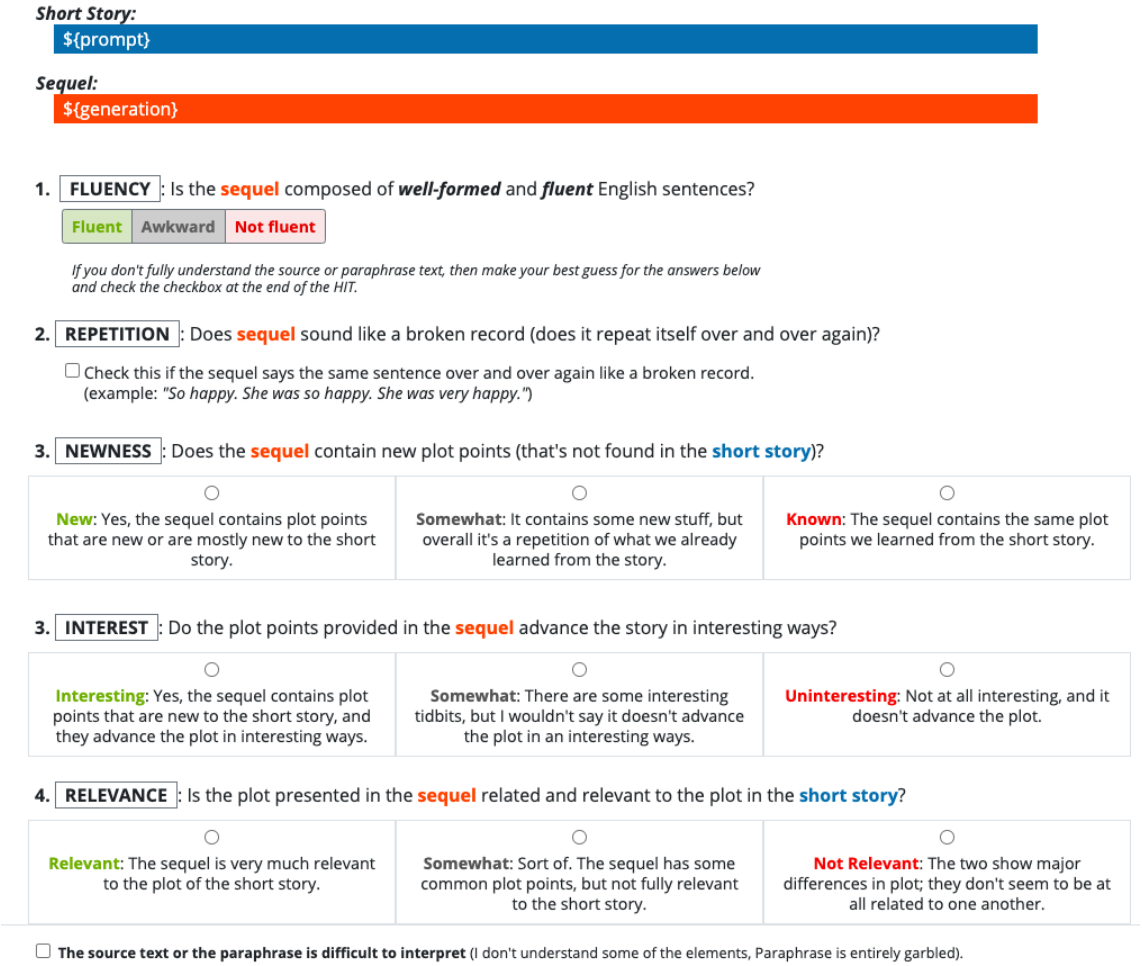

Figure 6: Mechanical Turk template used to evaluate ROCstories generations.

	propmt	Generated sentence with attention-decoding
w/o AttnM	run. team. field. drill. =	person runs a drill during a practice at training camp.
w/ AttnM.	run. team. field. drill. =	person runs a drill during a training session with his team.
w/o AttnM	use. tool. piece. metal. =	tool or piece of metal used in manufacturing.
w/ AttnM	use. tool. piece. metal. =	piece of metal used to make tools.

Table 12: Examples produced by GPT2-L without and with attention modulation. Use attention modulation would have higher concept coverage (details in Table 3 in the main text with 5% coverage improvements on all decoding algorithms we tested);

Figure 7: Mechanical Turk template used to evaluate CommonGen generations.