# Accelerating Entropy-Based Transformer Calibration

Michael Hu
Advisor: Karthik Narasimhan

# Motivation: Preliminaries



1. Language model: $\widehat{Pr}(w_t|w_{<t})$

1. Language model: $\widehat{Pr}(w_t|w_{<t})$
2. Entropy: a measure of the amount of information being produced by a source.

# Motivation: Preliminaries



1. Language model: $\widehat{Pr}(w_t|w_{<t})$
2. Entropy: a measure of the amount of information being produced by a source.
3. Entropy rate: the average amount of information per word. For a sentence, sum the entropies and divide by number of words.

# Motivation: Preliminaries



1. Language model: $\widehat{Pr}(w_t|w_{<t})$
2. Entropy: a measure of the amount of information being produced by a source.
3. Entropy rate: the average amount of information per word. For a sentence, sum the entropies and divide by number of words.
   - "A loud and boisterous carnival is outside my window." $\Rightarrow$ High entropy rate.

# Motivation: Preliminaries

1. Language model: $\widehat{Pr}(w_t|w_{<t})$
2. Entropy: a measure of the amount of information being produced by a source.
3. Entropy rate: the average amount of information per word. For a sentence, sum the entropies and divide by number of words.
   - "A loud and boisterous carnival is outside my window." $\Rightarrow$ High entropy rate.
   - "Nothing is outside my window." $\Rightarrow$ Low entropy rate.

# Motivation: Preliminaries

1. Language model: $\widehat{Pr}(w_t|w_{<t})$
2. Entropy: a measure of the amount of information being produced by a source.
3. Entropy rate: the average amount of information per word. For a sentence, sum the entropies and divide by number of words.
   - "A loud and boisterous carnival is outside my window." $\Rightarrow$ High entropy rate.
   - "Nothing is outside my window." $\Rightarrow$ Low entropy rate.
4. Entropy blowup: when the entropy rate of text produced by a source increases over time.

Entropy blowup does not occur in human-generated text.

- Intuition: people don't say increasingly weird things as you talk to them.

# Motivation



Entropy blowup does not occur in human-generated text.

- Intuition: people don't say increasingly weird things as you talk to them.

We observe entropy blowup on SOTA language models such as GPT-2.

# Goal

Develop a computationally efficient technique for minimizing entropy blowup in language models.

# Approach: Preliminaries



Entropy:

$$H(Pr) := \mathbb{E}_{w_{1:t} \sim Pr} \left[ \log \frac{1}{Pr(w_{1:t})} \right]$$

# Approach: Preliminaries

Entropy:

$$H(Pr) := \mathbb{E}_{w_{1:t} \sim Pr} \left[ \log \frac{1}{Pr(w_{1:t})} \right]$$

One-step lookahead entropy:

$$H(\widehat{W}_{t+1} | w_{\leq t}) = \mathbb{E}_{w_{t+1} \sim \widehat{Pr}} \left[ \log \frac{1}{\widehat{Pr}(w_{t+1} | w \leq t)} \right]$$

# Approach



Stepwise calibration:

1. Obtain the learned language model $\widehat{Pr}$.

Stepwise calibration:

1. Obtain the learned language model $\widehat{Pr}$.
2. Scale $\widehat{Pr}$ by $\alpha$ times the one-step lookahead entropy. Renormalize to obtain a new probability distribution $Pr_{\alpha}$.

Stepwise calibration:

1. Obtain the learned language model $\widehat{Pr}$.
2. Scale $\widehat{Pr}$ by $\alpha$ times the one-step lookahead entropy. Renormalize to obtain a new probability distribution $Pr_\alpha$.
3. Choose $\alpha$ such that it minimizes the cross-entropy loss between the context and $Pr_\alpha$.

Stepwise calibration:

1. Obtain the learned language model $\widehat{Pr}$.
2. Scale $\widehat{Pr}$ by $\alpha$ times the one-step lookahead entropy. Renormalize to obtain a new probability distribution $Pr_\alpha$.
3. Choose $\alpha$ such that it minimizes the cross-entropy loss between the context and $Pr_\alpha$.
4. Use $Pr_\alpha$ to predict the next word.

Computational complexity:

- $V$ = size of vocabulary = $\sim 50{,}000$ in practice
- $C$ = number of contexts = for a sentence, the number of words $\Rightarrow \sim 10$.

# Approach



Computational complexity:

- V = size of vocabulary = $\sim$ 50,000 in practice
- C = number of contexts = for a sentence, the number of words $\Rightarrow \sim$ 10.

The one-step lookahead entropy takes $O(VC)$ time to compute. The method, as proposed, is slow in practice. The factor we can control here is $V$. (If you want to feed the model more context, it is your right to do so.)

# Implementation



The major contribution of this IW: we can approximate the true one step lookahead entropy by calculating lookahead entropies for the top 100 or so words in the vocabulary.

# Implementation

The major contribution of this IW: we can approximate the true one step lookahead entropy by calculating lookahead entropies for the top 100 or so words in the vocabulary.

- Top 100 words capture about 90% of the probability mass!

The major contribution of this IW: we can approximate the true one step lookahead entropy by calculating lookahead entropies for the top 100 or so words in the vocabulary.

- Top 100 words capture about 90% of the probability mass!
- $V = 50{,}000 \Rightarrow 100$. If the model used to take 8 hours to run, it now takes 1 minute.