Accelerating Entropy-Based Transformer Calibration

Michael Hu

Advisor: Karthik Narasimhan

Motivation: Preliminaries

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

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.

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.

Motivation

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.

Related Work

- Analysis of language model sampling methods by Holtzman et al. [**?**]
- Calibration of neural networks via temperature scaling by Guo et al. [**?**]

Approach: Preliminaries

Conditional entropy:

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

Approach: Preliminaries

Conditional entropy:

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

Conditional 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 $\widehat{Pr}$.

Approach

Stepwise calibration:

1. Obtain the learned language model $\widehat{Pr}$.
2. Scale $\widehat{Pr}$ by $\exp\{-\alpha \cdot H(\widehat{W}_{t+1}|w_{\leq t})\}$. Renormalize to obtain a new probability distribution Pr_α.

Approach

Stepwise calibration:

1. Obtain the learned language model $\widehat{Pr}$.
2. Scale $\widehat{Pr}$ by $\exp\{-\alpha \cdot H(\widehat{W}_{t+1}|w_{\leq t})\}$. Renormalize to obtain a new probability distribution Pr_α.
3. Choose α such that it minimizes the cross-entropy loss between the context and Pr_α.

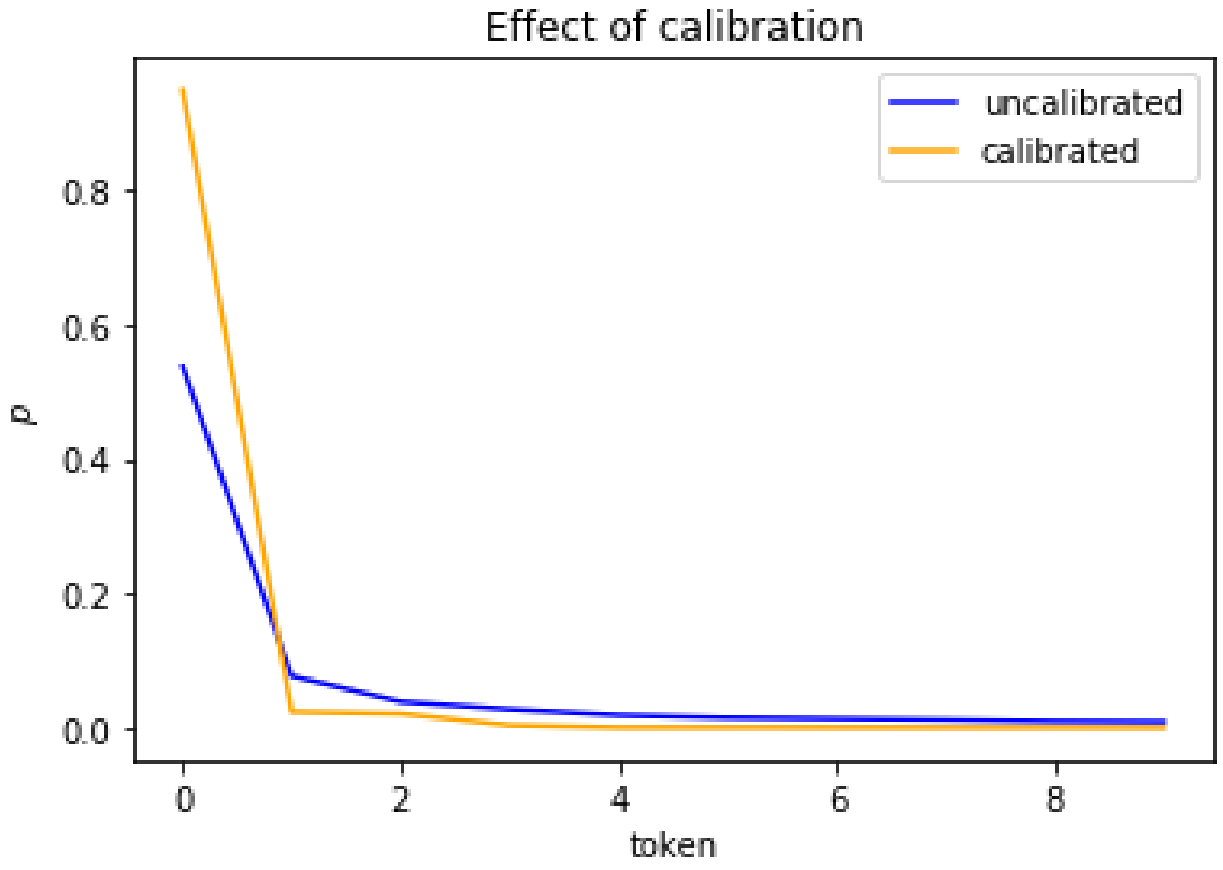
Approach

Stepwise calibration:

1. Obtain the learned language model $\widehat{Pr}$.
2. Scale $\widehat{Pr}$ by $\exp\{-\alpha \cdot H(\widehat{W}_{t+1}|w_{\leq t})\}$. Renormalize to obtain a new probability distribution Pr_α.
3. Choose α such that it minimizes the cross-entropy loss between the context and Pr_α.

Use Pr_α as your new language model.

Approach


Effect of calibration
uncalibrated
calibrated
p
0.0
0.2
0.4
0.6
0.8
0
2
4
6
8
token

Approach: How to improve?

Computational complexity:

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

Approach: How to improve?

Computational complexity:

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

The cross-entropy loss takes $O(V^2 C)$ time to compute.

Computing Pr_α takes $O(V^2)$ time $\Rightarrow$ sampling T words takes $O(V^2 T)$ time.

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

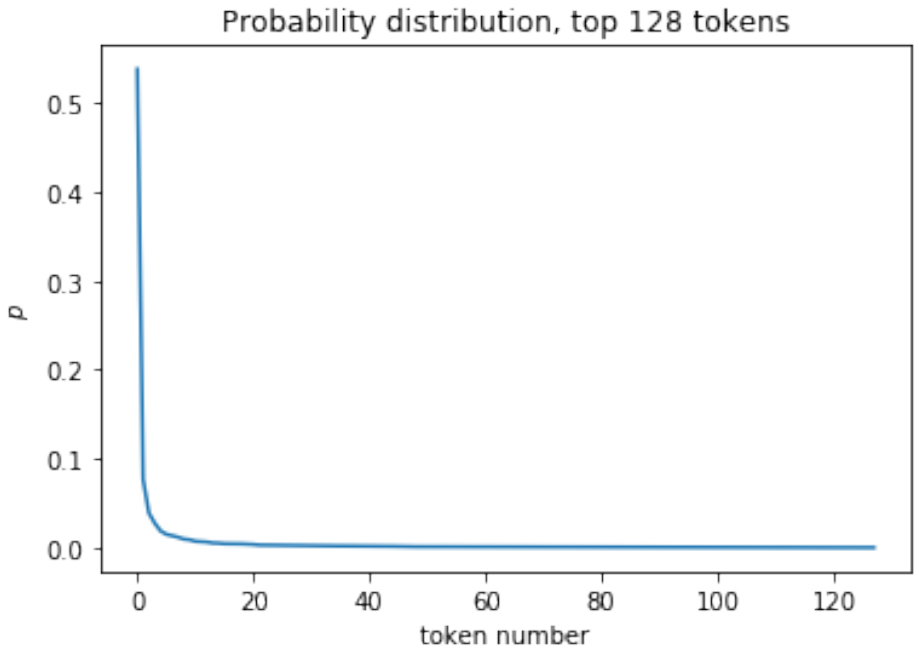
The major contribution of this IW: we can approximate the true one step lookahead entropy by calculating lookahead entropies for the top 100 or so words in the vocabulary.

Impact:

- $V = 50{,}000 \Rightarrow 100$. If the model used to take 8 hours to calibrate or generate T words, it now takes 1 minute.

Result 1

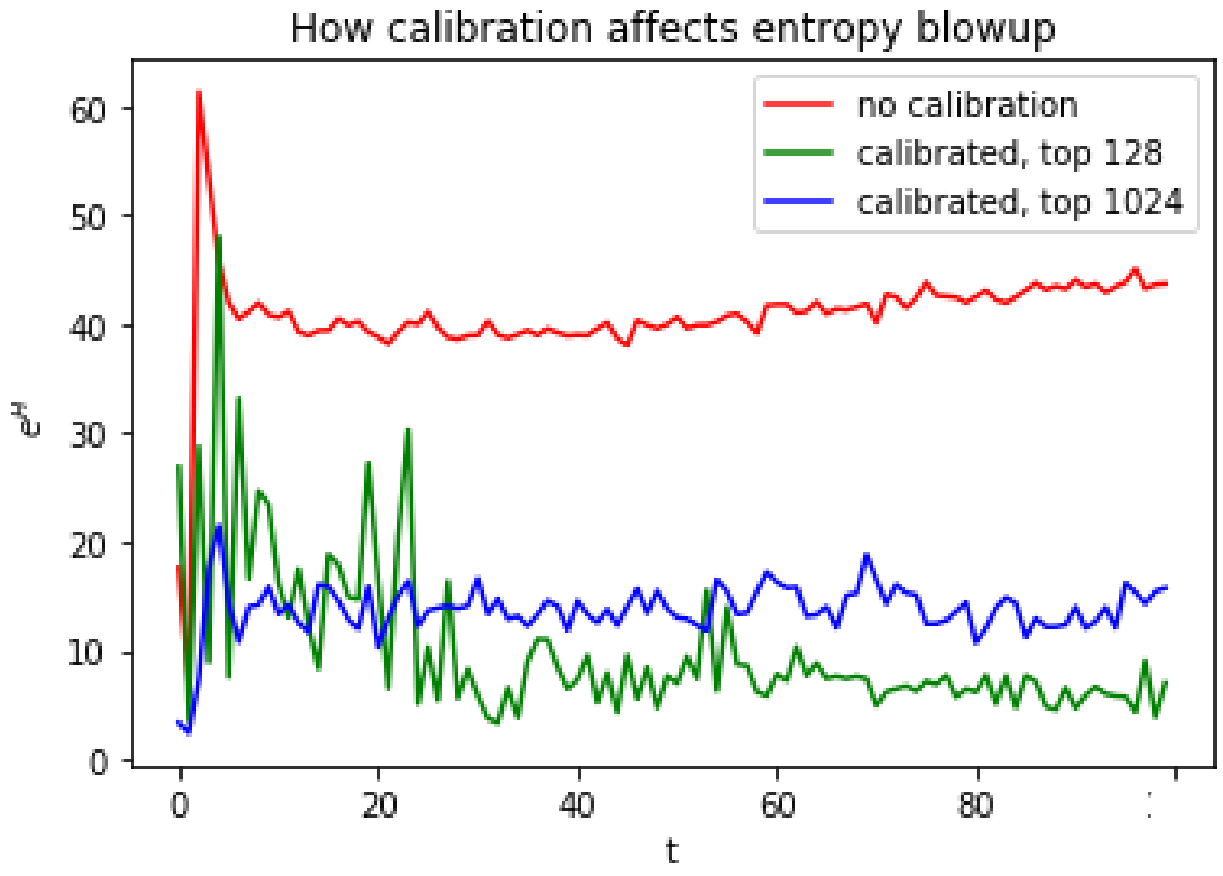
On average, top 128 words capture $> 90\%$ of the probability mass.



Here: 92.8%

Result 2

Generating using approximations of Pr_α minimizes entropy blowup.



Result 3

We observe no qualitative drawbacks when using approximations of Pr_α.

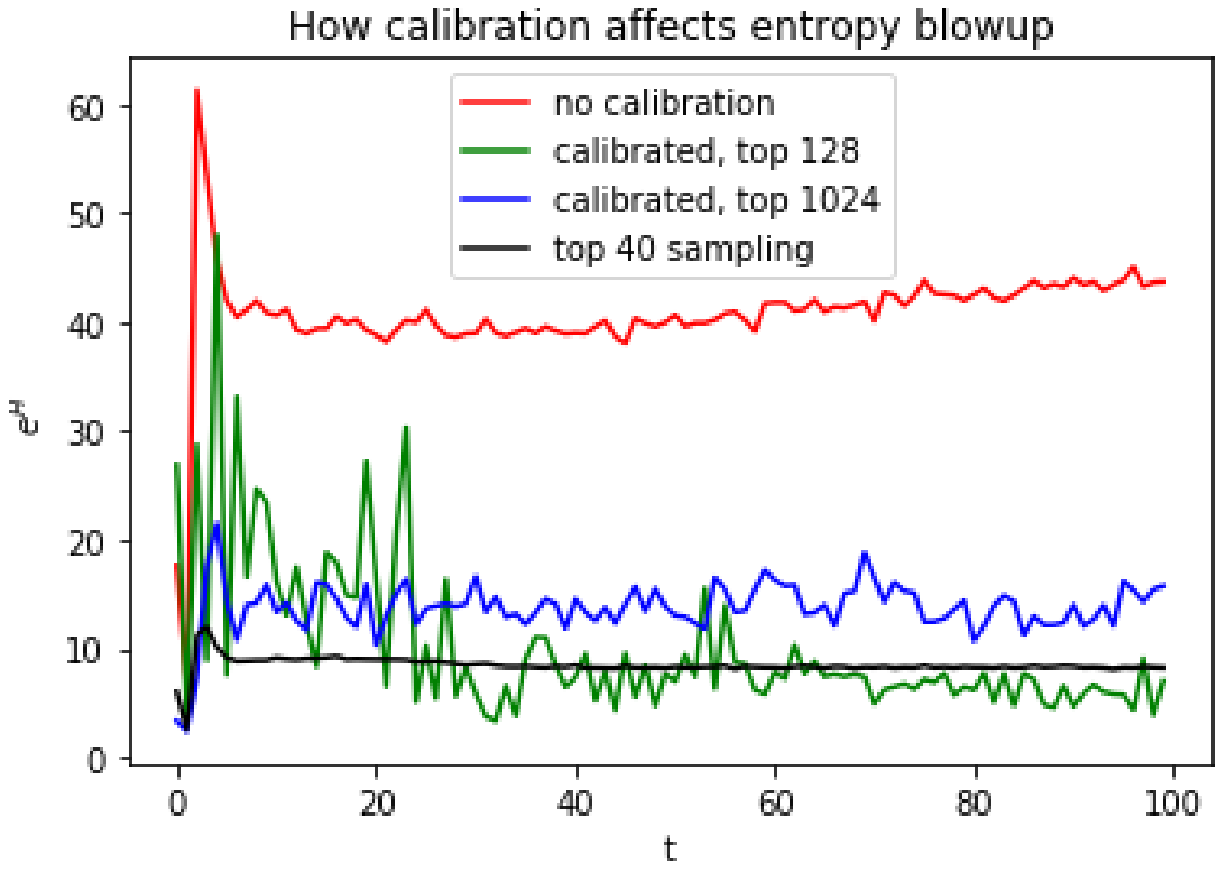
Result 4

A plug-and-play API for calibration, available at `https://github.com/mikkyhu/transformers/blob/master/examples/calibrate.py`

Conclusion

Approximating Pr_α is a viable way to speed up calibration and generating from Pr_α.

Future work


How calibration affects entropy blowup
no calibration
calibrated, top 128
calibrated, top 1024
top 40 sampling
e^H
t
0
10
20
30
40
50
60
0
20
40
60
80
100

Acknowledgements

- Karthik Narasimhan
- Cyril Zhang