Entropy-Based Calibration of GPT-2

Michael Hu (myhu), 2021
Advisor: Prof. Karthik Narasimhan

October 2, 2019

Motivation and Goal

Broadly speaking, the motivation is to develop techniques that will help us train better text-generating models. Here, "better" is generating text that is closer to normal language. (More on this later). The approach of choice here is step-wise calibration. With step-wise calibration, we adjust the model after every generated word, recalibrating it to match some property of the target distribution. Concretely, our goal is to develop an efficient algorithm that calibrates text-generating models. At each step, the algorithm should calibrate the model towards choosing words such that the entropy rate of the model's text is close to that of the underlying language.

Problem Background and Related Work

A challenge in text generation is creating metrics to evaluate text-generating models. Suppose that a human believes that one model produces better English than the rest. How can we measure this "goodness" quantitatively? One method is to calculate the entropy rate of the model's text output and compare it against the entropy rate of the underlying language itself. Informally, informational entropy is the average rate at which a data source produces information [4]. Knowing this, we can reason that our model should produce information at a rate close to that of the baseline language.

With this in mind, in 2019 Braverman et. al. proposed Algorithm 2, a step-wise "local entropy rate calibration" algorithm. Essentially, the algorithm calibrates models to choose words that minimize growth in the entropy rate one step in the future. Experimentally, Algorithm 2 decreases entropy growth compared to the baseline model after one iteration [1].

Approach

Algorithm 2 calculates entropy rates for all possible words one step in the future, which is to say that it calculates entropy rates for *all words in the English vocabulary.* This computation is slow. Instead of evaluating all possible outcomes, we propose using sampling

approaches or heuristic approaches to choose words instead.

One such heuristic approach is beam search. Beam search is a best-first search algorithm that keeps track of the k best candidates in every iteration. During each iteration, beam search considers all results from the k candidates. It then chooses the k best of these results for seeds in the next iteration [2]. We could use beam search to evaluate the entropy growth of the k words most probable to come next and pick the word that best minimizes the entropy growth. Concretely, our approach is to use bounds or search algorithms like beam search to decrease the computational complexity of Algorithm 2.

Plan

We will first attempt to implement Algorithm 2 using beam search instead of an explicit computation over the entire vocabulary. We will then measure the entropy growth of text generated by models trained under this new algorithm, and compare the growth to that of text generated solely by the models themselves. If beam search is unsuccessful, we will consider other techniques, such as other graph search algorithms or simply choosing a fixed number of most probable words to consider. We will then repeat the above process.

Evaluation

We will evaluate our new algorithm on two metrics: model training time and entropy growth of the generated text. Ideally, we want to develop an algorithm such that model training time is only increased by a small factor, but the entropy rate of the resulting text is closer to that of the underlying language.

We will be using our algorithm to calibrate GPT-2, a deep-learning-based NLP model that is very good at generating text, qualitatively speaking [3]. We will measure GPT-2 training times before and after calibration, and compare the entropy rates of the output texts to the underlying language in both cases. Hopefully, we will observe improvement after GPT-2 has been calibrated.

References

[1] Mark Braverman, Xinyi Chen, Sham M. Kakade, Karthik Narasimhan, Cyril Zhang, and Yi Zhang. Calibration, entropy rates, and memory in language models. 2019.

[2] Bruce T. Lowerre. The harpy speech understanding system. 1980.

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

[4] C. E. Shannon. Prediction and entropy of printed english. *Bell System Technical Journal*, 30(1):50–64, 1951.