Language Models over Canonical Byte-Pair Encodings

Tim Vieira Tianyu Liu Clemente Pasti Yahya Emara Brian DuSell Benjamin LeBrun
Mario Giulianelli Juan Luis Gastaldi Timothy J. O'Donnell Ryan Cotterell

ETH zürich
McGill UNIVERSITY

Background

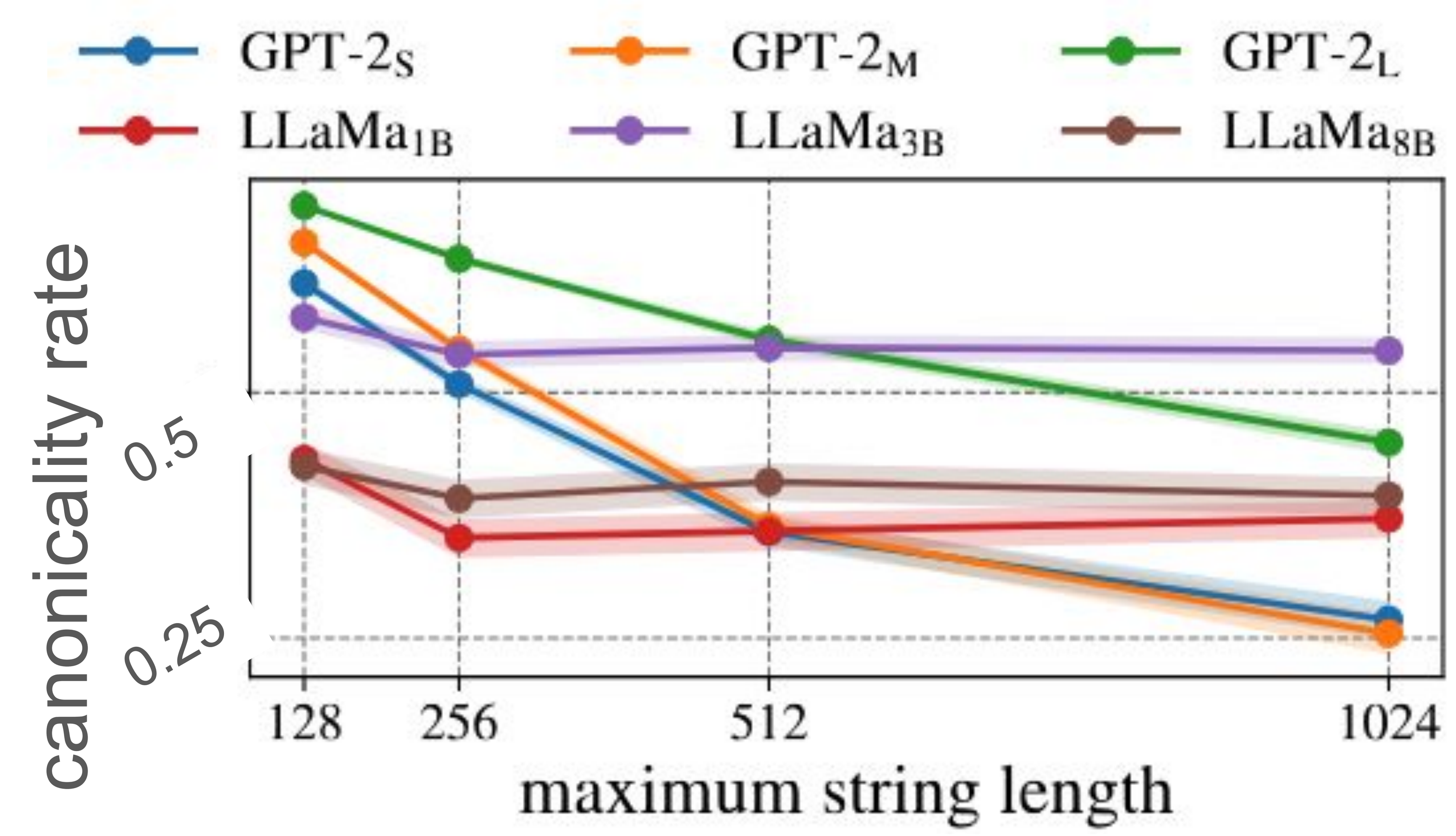
In byte-pair encodings (BPE), some token combinations never appear during training, but might appear during decoding.

Example: "`Hello world`"

BPE		Canonical
`Hello`	`␣world`	✅
`Hello`	`␣` `world`	❌

This misallocation is both erroneous, as noncanonical strings never appear in training data, and wasteful!

Much of the probability mass is misallocated to noncanonical sequences!



Most sequences sampled from most language models are noncanonical!
Longer sequences are more vulnerable to this probability mass leakage..

Theoretical Guarantees

Enforcing canonicality is guaranteed to make language models better (closer to the ground-truth distribution of sentences).

Reduction in KL divergence to the ground-truth LM

p_Δ	LM before canonicalization
$p^\star_\Delta$	Ground-truth LM
g	LM after canonicalization

$$\mathrm{KL}(p^\star_\Delta \,\|\, p_\Delta) - \mathrm{KL}(p^\star_\Delta \,\|\, g) = \underbrace{-\log Z}_{\geq 0}$$

Where Z is the **canonicality rate** of the LM before canonicalization

Enforcing Canonicality

We propose two ways to enforce canonicality in LMs.

1. **Canonicality by conditioning**

Without retraining the language model, we develop an efficient algorithm that forces only canonical sequences to be generated.

2. **Canonicality by construction**

We finetune a language model to get a parameterization that guarantees canonical outputs.

The Algorithm

For two any two tokens δ and δ' in the vocabulary Δ, we compute find_conflict(δ,δ') to check if it causes a noncanonicality. The find_conflict function can be pre-computed for all tokens as masks. And we use the masks in a logits processor to avoid generating any tokens that lead to a noncanonical sequence.

Results

Improvement = log(canonicality rate)

		Model	Baseline	Local	Global
PTB	GPT-2	small	201.0	200.7	**199.1**
		medium	195.1	194.5	**193.1**
		large	189.4	188.9	**188.2**
	Llama	1B	171.2	171.1	**169.7**
		3B	165.0	165.0	**164.2**
		8B	161.5	161.5	**160.1**
WikiText	GPT-2	small	369.2	**367.0**	367.3
		medium	334.1	333.2	**332.2**
		large	320.8	**319.1**	319.6
	Llama	1B	286.7	**284.4**	285.2
		3B	264.6	**262.0**	263.7
		8B	248.2	**245.8**	246.8

We observe improvements in language modeling log-likelihood on all models on PTB and WikiText, without any finetuning.

The gain is large when the LM has lower canonicality rate. Also, the gain is larger when modeling longer sequences.

paper: