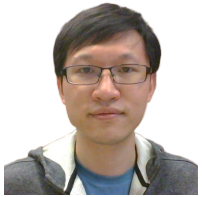
Google DeepMind

Why is prompting hard?

Understanding Prompts on Binary Sequence Predictors



Li **Kevin** Wenliang

Jordi Grau-Moya

Anian Ruoss

Marcus Hutter

Tim Genewein

Prompting isn't easy

We quite often find it hard to **prompt with intuition**

Optimized prompts may appear **unintuitive**

Instruction	Acc
Let's think step by step.	71.8
Let's work this out in a step by step way to be sure we have the right answer.	58.8
(empty string)	34.0
Take a deep breath and work on this problem step-by-step.	**80.2**
Break this down.	79.9

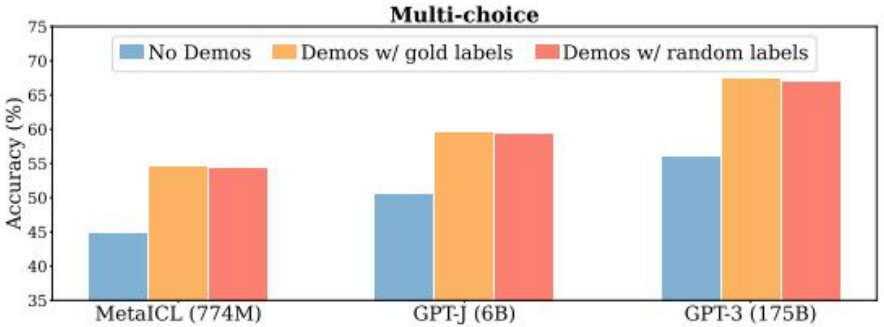
Yang+ 22

Apoploe vesrreaitais eating *Contarra ccetnxniams luryca tanniounons*

Daras+ 22

Task	Instruction
penguins_in_a_table	Here is my new text:
disambiguation_qa	Identifying Antecedents of Pronouns: A Comprehensive Guide
temporal_sequences	The answer is the time that is not mentioned in the given statements

Zhou+ 22



Min+ 22

Prompting as conditioning a sequence predictor

Latent:	$p(\tau)$
Conditional:	$p(x_{1:T}\|\tau)$
Prompt:	$s_{1:L}$
Sequence:	$x_{1:T}$

In theory: Bayes predictor is the optimal predictor given *p*

$$p_{\mathrm{B}}(x_{1:T}|s_{1:L}) = \int p(x_{1:T}|\tau, s_{1:L})\mathrm{d}p(\tau|s_{1:L})$$

In practice: Neural predictors trained from finite datasets from *p*

$$p_\theta(x_{1:T}|s_{1:L}), \quad \theta \leftarrow \theta + \gamma\nabla_\theta \log p_\theta(x_{1:T})$$

With enough training: $p_\theta \longrightarrow p_{\mathrm{B}}$

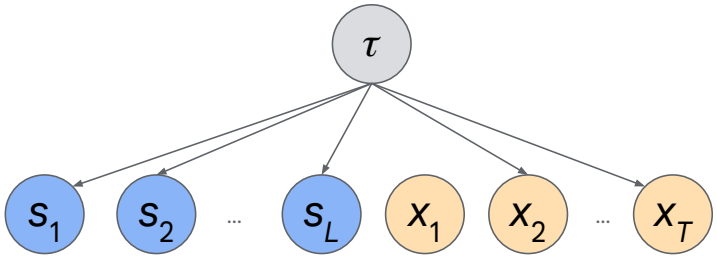
[Ortega+ 19; Mukulik+ 20; Genewein+ 23]

Downstream task distribution *q*, the optimal prompt top to length L_{max} is:

$$s^*_{L_{\max}} := \underset{s_{1:L}\in\mathcal{A}^L,\, L\leq L_{\max}}{\arg\max} \mathbb{E}_{q(x_{1:T})}\left[\log p_{\mathrm{B}}\left(x_{1:T}|s_{1:L}\right)\right]$$

Suppose you have a binary sequence predictor...

Minimalist approach: consider coin flip sequences



Prompt (length L) Sequence (length T)

τ=0.2: 01010100101000001…

τ=0.7: 10111111010111111…

Pretrain with $p(\tau)$
Test predictive log-likelihood with $q(\tau)$

Advantages

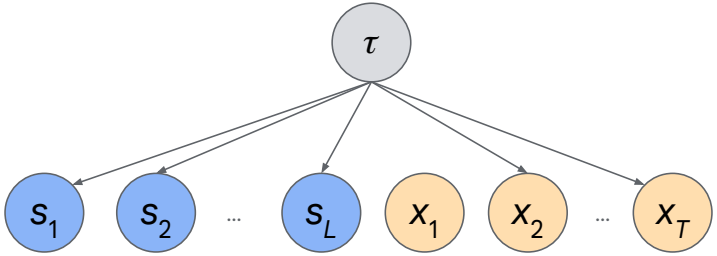
- Bayes predictor in closed-form

$$p_{\mathrm{B}}(x_{1:T}|s_{1:L}) = \int p(x_{1:T}|\tau, s_{1:L})\mathrm{d}p(\tau|s_{1:L})$$

- Prompts in counts: (#0s, #1s)

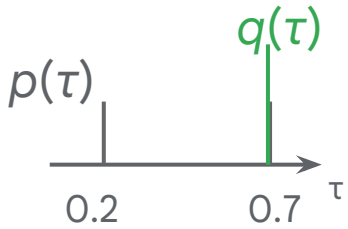
Question: What's the best prompt to generate 70% 1s in the sequence?

Counter-intuitive prompts in simplest setup



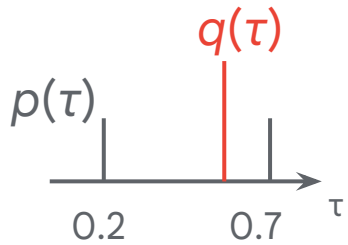
Task: generate a prompt that will induce 70% 1s in the sequence

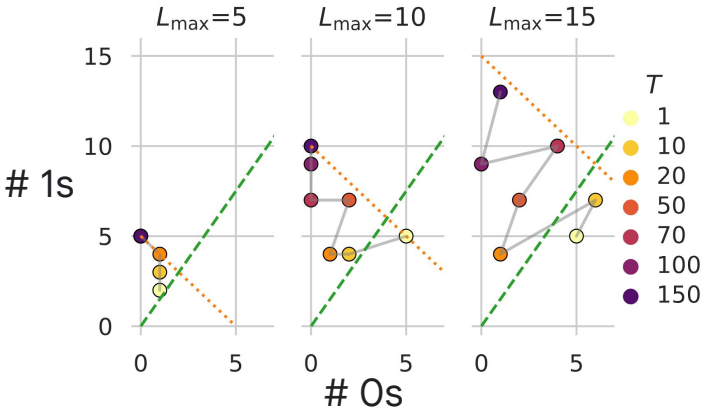
In-meta-distribution



Best prompt: 11111111111... all 1s!
Reason: need to move mass $p(\tau|s_{1:L})$ to 0.7

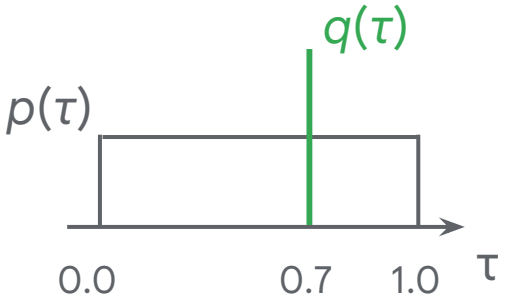
Out-of-meta-distribution

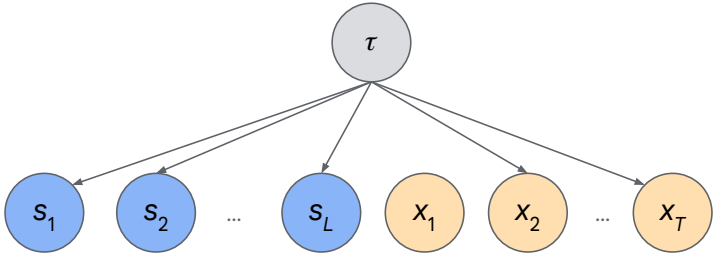




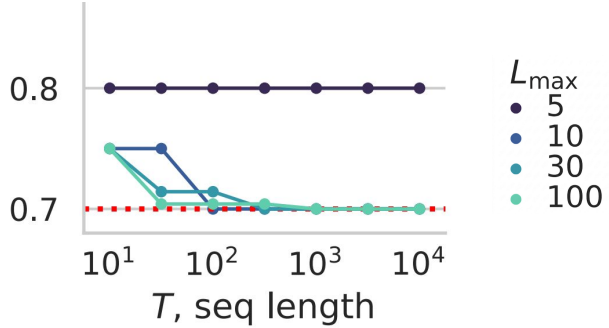
Optimal prompts can be intuitive

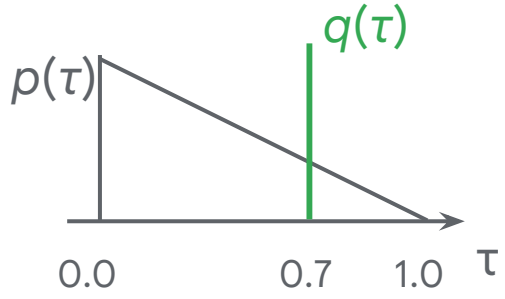
Task: generate a prompt that will induce 70% 1s in the sequence



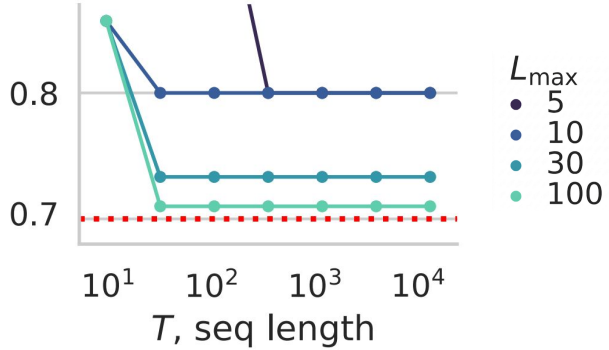


Prop. of 1s in optimal prompt



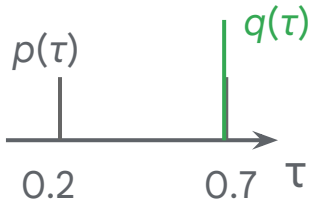


Prop. of 1s in optimal prompt



Pretraining distribution affects the optimal prompt

Optimizing on **neural predictors p_θ**



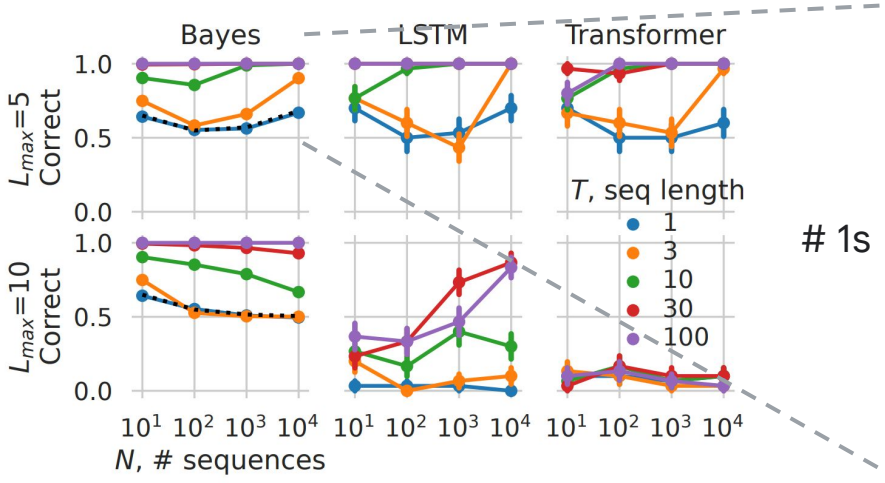
Give N sequences from q, how often can we find the 1111… prompt?

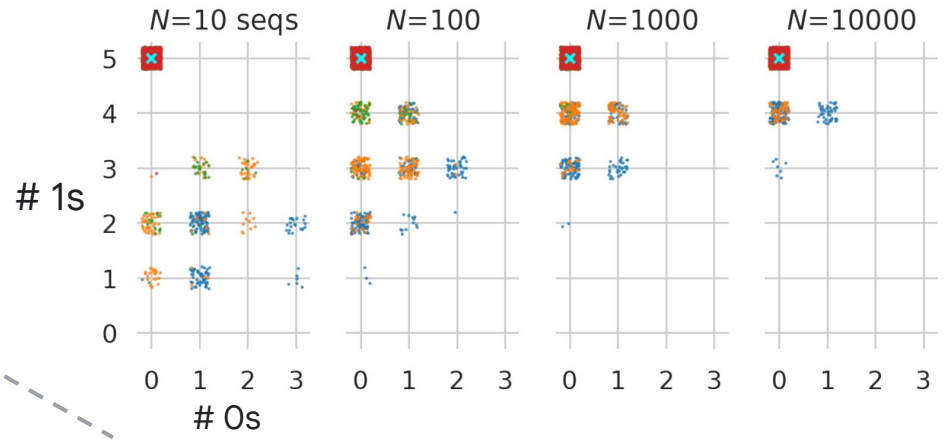
N sequences

1	0	1	1	1	0	0
0	1	1	0	1	0	1
1	1	1	1	0	1	1
0	1	1	1	1	1	0

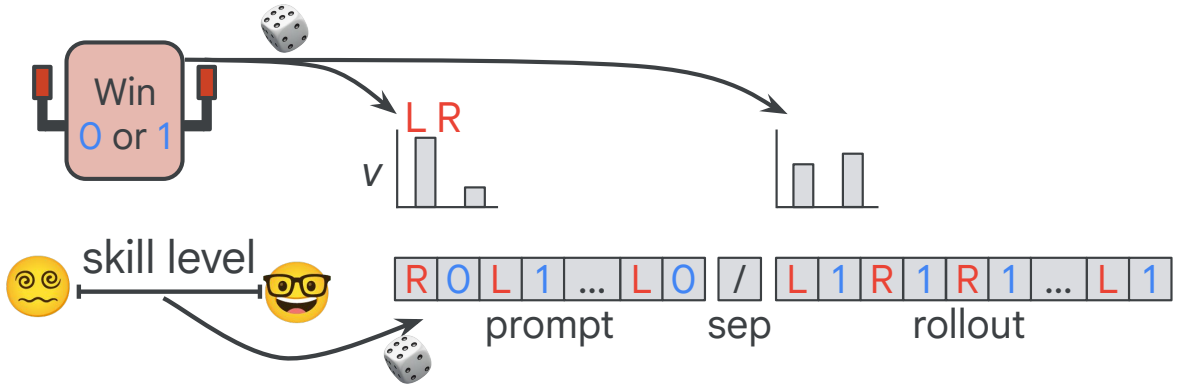
T, seq length

~ q, find s*





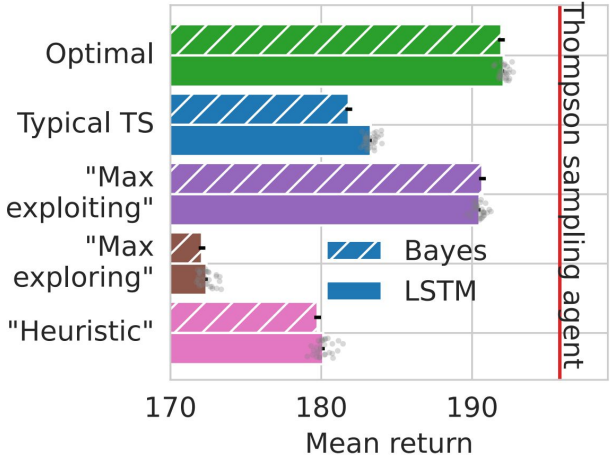
Binary in-context learning task



Theoretically optimal prompts (4 in total)

L	0	R	1	R	1	R	1	R	1	R	1	R	1	R	1
L	0	R	1	R	1	R	1	R	1	R	1	R	1	R	0

The optimal prompt appears very greedy!



Thank you

Why is prompting hard?

Unknown pretraining distribution => Unintuitive prompts

(from a Bayesian perspective)

Come to our poster #170 @ 3pm

- Information-theoretic interpretation of optimal prompts
- Non-convergence of prompt optimization
- Real LLMs and IMDB dataset

Google DeepMind