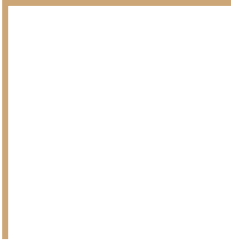



Generation Algorithms

11-711 Spring 2024





probability
distributions



Good news!

We have a great new model M !

7 billion parameters!

Pretrained on billions of tokens of text!



So what's in the box?



A model defines a *conditional probability distribution*

$$P(Y|X) = \prod_{j=1}^J P(y_j | X, y_1, \dots, y_{j-1})$$

A model defines a *conditional probability distribution*

Input X

English text

Question

Document

Utterance

Chess game state

Math problem

Output Y

Japanese

Answer

Short description

Response

Next chess move

Answer

Task

Translation

Question-answering

Summarization

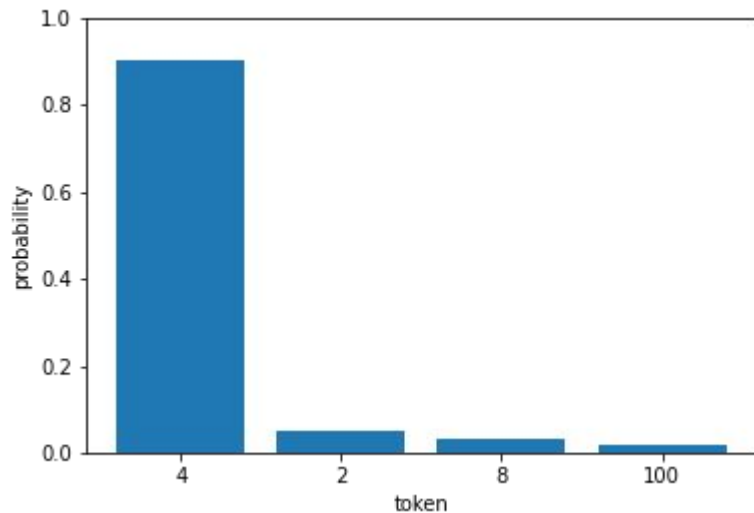
Response generation

Game-playing

Math reasoning

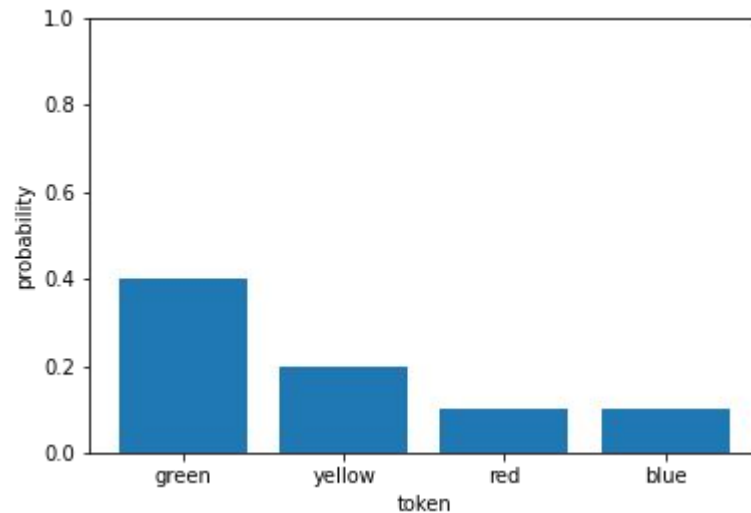
Probability distributions: confidence

M("2 + 2 = "):



4 (high confidence)

M("Graham's favorite color is "):



green (low confidence)

Calibration (quick reminder)

A model is **well-calibrated** if the confidence score is well-correlated with the probability of correctness

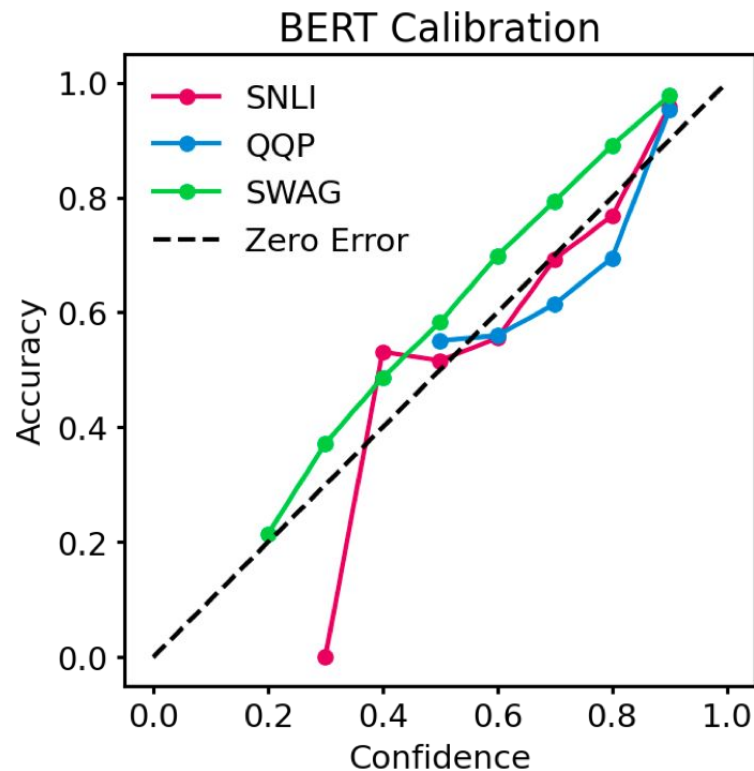


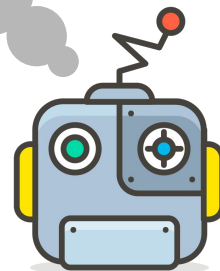
Figure from [Desai & Durrett \(2020\)](#)

Calibration can be hard for longer outputs...

What if we have multiple ways to say the same thing? Probability is **split** between them

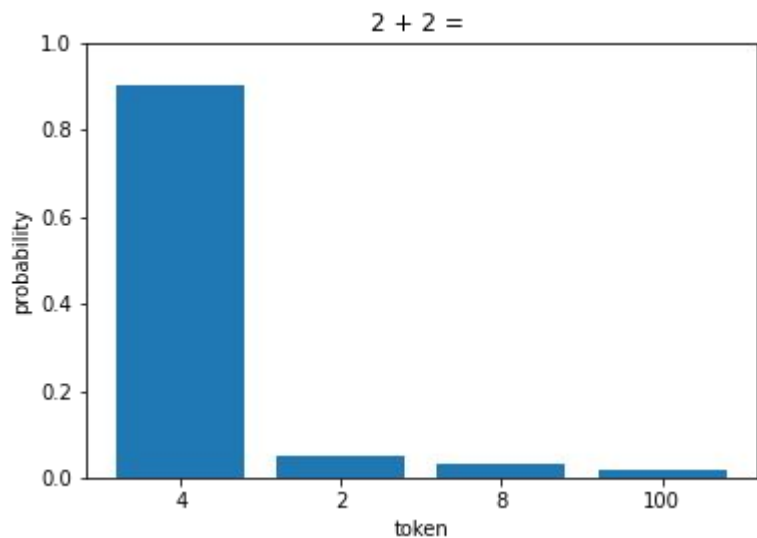
The U.S. president is...

- a) Joe Biden
- b) Joseph Biden
- c) Joseph Robinette Biden
- d) Biden



Probability distributions: hallucination

Models generally assign non-zero probability to some incorrect outputs



This is true *even if all
pretraining data is factual!*

Calibrated Language Models Must Hallucinate

Adam Tauman Kalai
Microsoft Research

Santosh S. Vempala
Georgia Tech

December 5, 2023

Reference: [Kalai & Vempala, 2023](#)

How do we get outputs from this model?

$$P(Y|X) = \prod_{j=1}^J P(y_j | X, y_1, \dots, y_{j-1})$$

We know:

- The model's distribution of likelihood over all vocabulary tokens V , for the next time step, given the input and previous generations

We want:

- a "good" output

Previous: probability distributions

up next:

sampling from LMs

Ancestral Sampling

$$y_j \sim P(y_j | X, y_1, \dots, y_{j-1})$$

- Exactly samples from model distribution!
- So we're done... right?

Issues with ancestral sampling: long tail

Llama has 32,000 vocabulary tokens!

Even if each individual token in the long tail has very little probability....
these small probabilities add up



What if we just ignore the long tail?

Top-k sampling: only sample from the most probable <k> next tokens

E.g., for k=6:

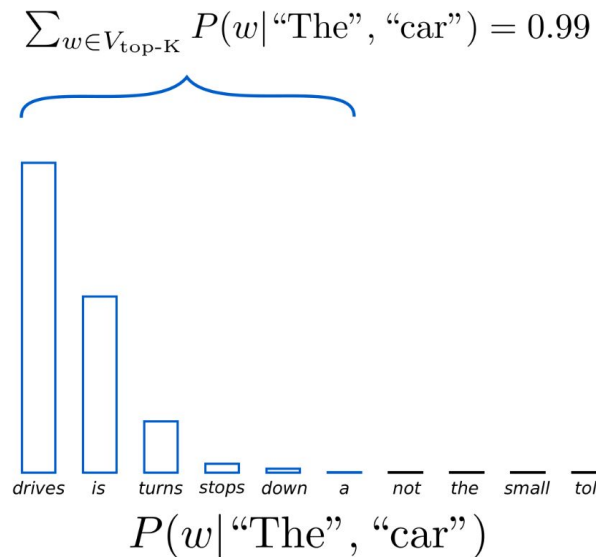
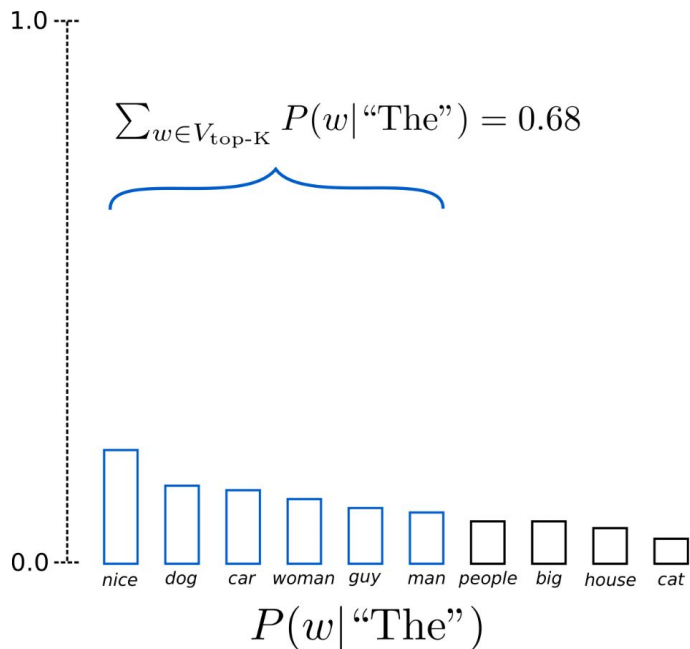


Figure from the [HuggingFace blog on text generation](#)

What if we just ignore the long tail?

Top-p (nucleus) sampling: only sample from the top <p> probability mass

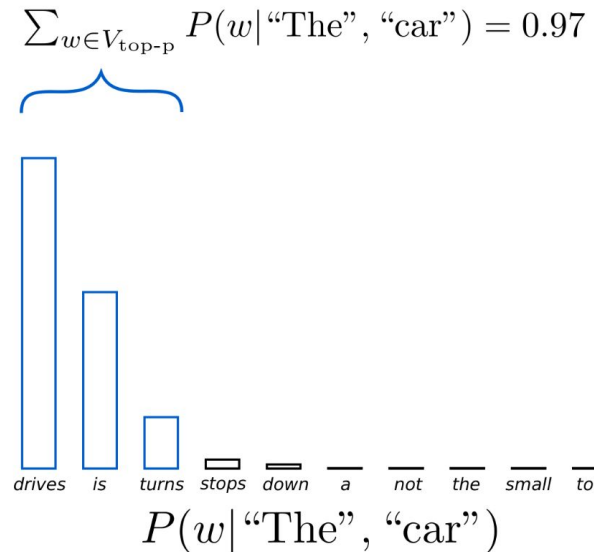
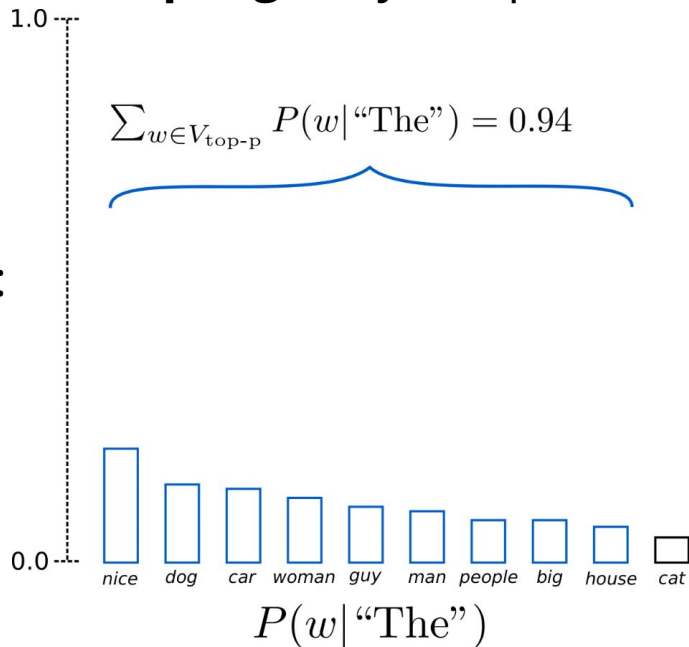


Figure from the [HuggingFace blog on text generation](#)

What if we just ignore the long tail?

Epsilon sampling: only sample tokens with probability of at least ϵ

E.g., for $\epsilon=0.05$:

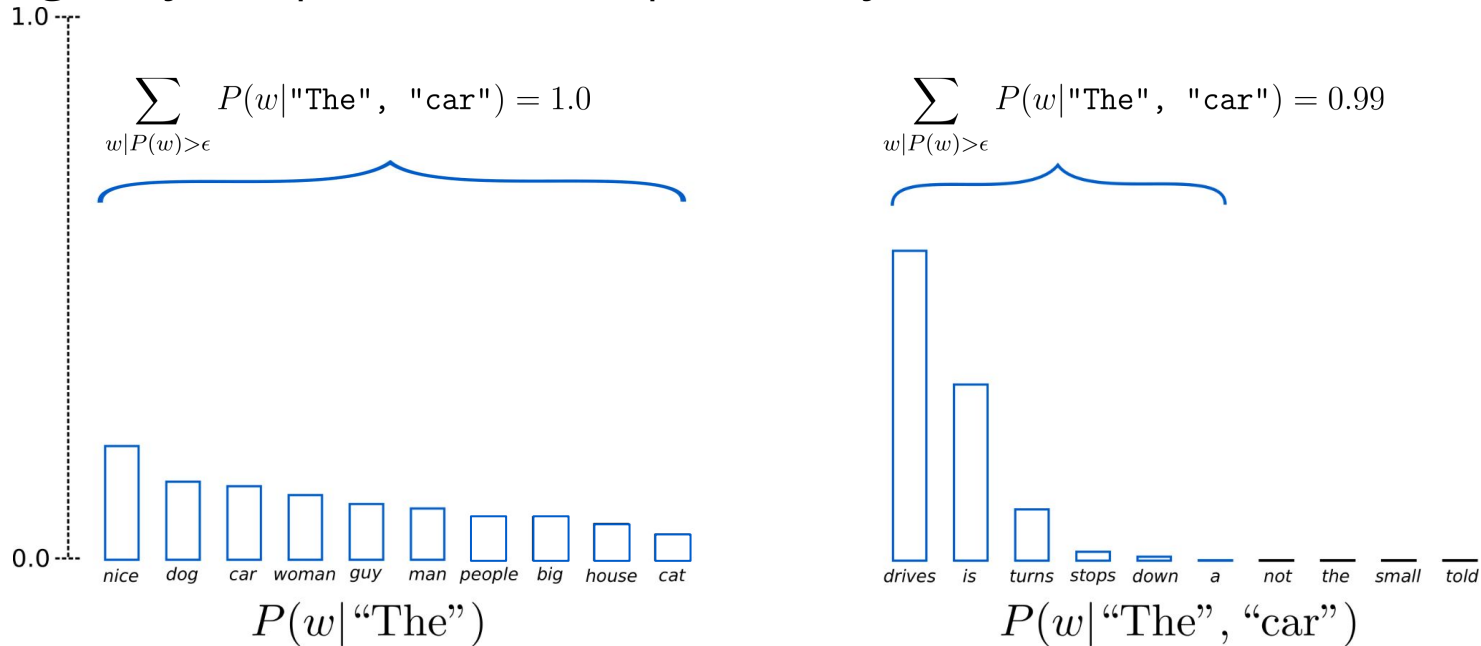
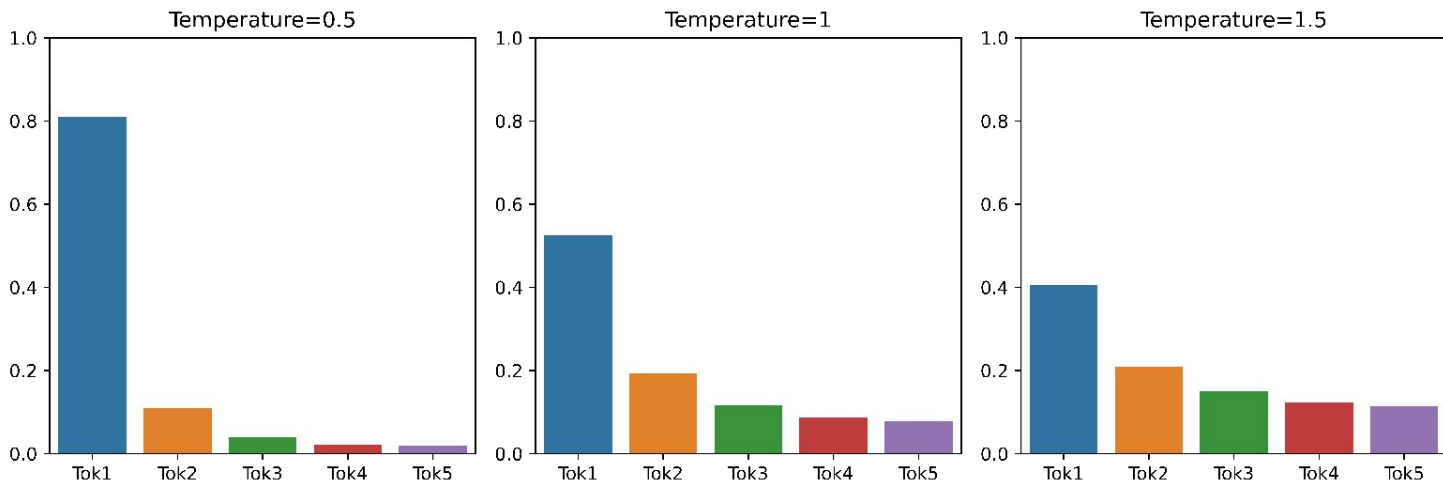


Figure modified from the [HuggingFace blog on text generation](#)

Distribution temperature

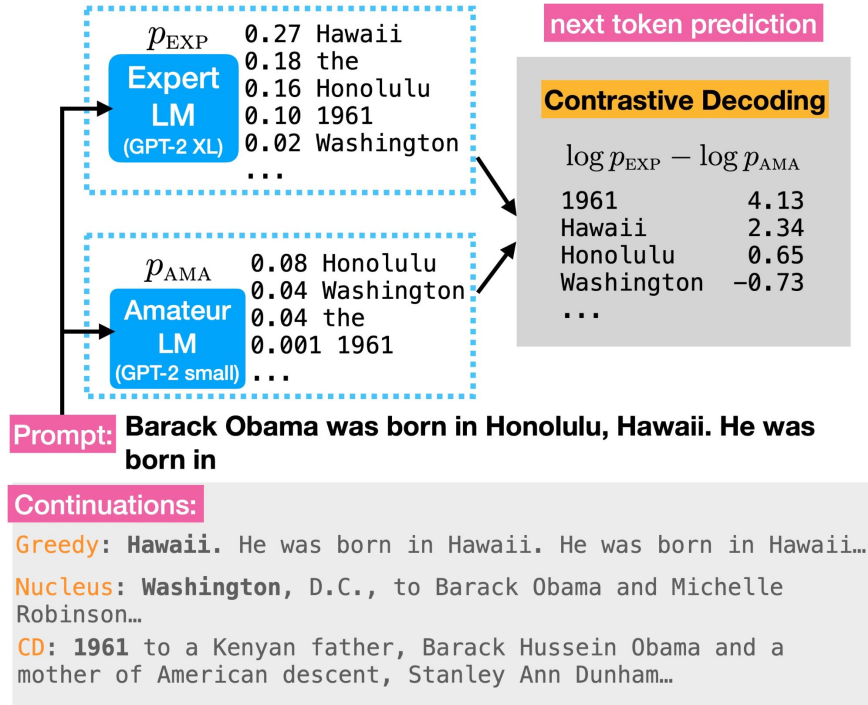
Idea: manipulate the distribution to have higher (or lower) probability on the top few tokens



Contrastive decoding

Smaller models make different mistakes– can we learn from these to improve our models?

Choose outputs that the “expert” finds much more likely than the “amateur”



Previous: sampling

up next:

mode-seeking search

Mode-seeking decoding methods

Given our inputs (evidence) and the model's parameters (prior), what's the *single most likely* output?

$$\hat{Y} = \underset{Y}{\operatorname{argmax}} P(Y|X)$$

(this is the mode of the distribution over outputs!)

Greedy decoding

Idea: choose the single most likely token at each step

$$y_j = \operatorname{argmax} P(y_j | X, y_1, \dots, y_{j-1})$$

Exactly what we want for a single-token output!

What about longer sequences? Doesn't always yield the highest-probability output :(

Beam search

Idea: maintain a few options, so we don't miss a high-probability completion "hidden" behind a lower-probability prefix

Breadth-first search: explore many options for each decoding step before generating candidates for the next step

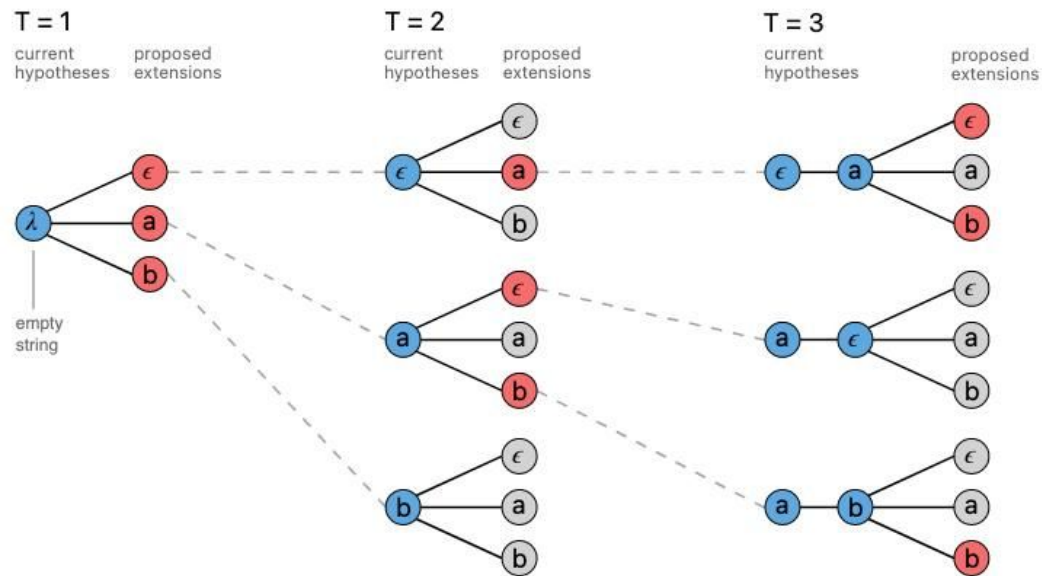


Figure from the [PyTorch blog on fast decoding](#)

Improving diversity: diverse and stochastic beam search

Idea: try to get some benefits of sampling *during* beam search

Diverse beam search: modify the **scoring** when pruning beams to avoid choosing overly similar beams

Stochastic beam search: modify the **next token selection** to sample instead of using the top greedy decodings

Previous: mode-seeking search

up next:

Minimum Bayes risk

Wait: is the highest-probability output best?

Outputs with *low probability* tend to be worse than those with *high probability*

Probability	Output
0.3	The cat sat down.
0.001	The cat grew wings.

But when you're *just* comparing the top outputs... it's less clear

Probability	Output
0.3	The cat sat down.
0.25	The cat ran away.

Is the highest-probability output best?

6 outputs:

Probability	Output
0.3	The cat sat down.
0.149	The cat got out of there.
0.2	The cat sprinted off.
0.25	The cat ran away.
0.1	The cat is very small.
0.001	The cat grew wings.

The single most probable output is that the cat sat down...

But 60% of the probability mass says something meaning "the cat left"!

The probability of this is spread over multiple similar generations

Then what makes an output good?

5 outputs:

Probability	Output
0.3	The cat sat down.
0.149	The cat got out of there.
0.2	The cat sprinted off.
0.25	The cat ran away.
0.1	The cat is very small.
0.001	The cat grew wings.

“The cat sat down” is *high probability* but different from the other generations...

“The cat ran away” is *high probability* and similar to other generations, so it's *lower risk*

We want an output that is *high probability* and *low risk*

Minimum Bayes Risk (MBR)

$$\hat{y} = \operatorname{argmax}_{y' \in \mathcal{Y}_h} \sum_{y \in \mathcal{Y}_e} G(y, y')$$

We want an output that is high probability and low risk

This is a useful framing for other methods, too...

It's MBR All the Way Down: Modern Generation Techniques Through the Lens of Minimum Bayes Risk

Amanda Bertsch* and **Alex Xie*** and **Graham Neubig** and **Matthew R. Gormley**
Carnegie Mellon University
[abertsch, alexx]`@cs.cmu.edu`

Reference: [Bertsch, Xie et al \(2023\)](#)

MBR variants: output ensembling

We have outputs from multiple models... how do we choose the best output?

Post-Ensemble (Kobayshi 2018): compare pairwise embedding similarity between outputs across models, choose the output with greatest average similarity

$$\hat{y} = \operatorname{argmax}_{y' \in \mathcal{Y}_h} \sum_{y \in \mathcal{Y}_e} G(y, y')$$

We want an output that is high probability and low risk

MBR variants: self-consistency (Wang et al., 2023)

1. Prompt for an answer using chain of thought
2. Sample multiple outputs
3. Extract the *answer* from each (ignore the explanations)
4. Return the most frequently generated answer

$$\hat{y} = \operatorname{argmax}_{y' \in \mathcal{Y}_h} \sum_{y \in \mathcal{Y}_e} G(y, y')$$

We want an output that is high probability and low risk



Previous: MBR

up next:

constrained
generation



Imposing global constraints

M("Describe a few hobbies I could try to stay in shape.")



I don't want to try
climbing!

How do we prevent the model from suggesting climbing?

Putting instructions in the input isn't enough



What are a few hobbies I could try to stay in shape? I don't want to try climbing. 

[View other drafts](#) 







There are tons of fun hobbies you can try to stay in shape without scaling any cliffs! Here are a few ideas based on different preferences:

For the outdoor enthusiast:

- **Hiking or backpacking:** Immerse yourself in nature and get a great workout on scenic trails. Choose beginner-friendly paths at first and gradually increase your difficulty as you build fitness.
- **Kayaking or canoeing:** Enjoy a peaceful paddle on lakes or rivers while getting your upper body and core engaged. Can be social with friends or enjoyed solo.
- **Cycling:** Explore your city, countryside, or trails on two wheels. Choose a road bike for speed, a mountain bike for off-road adventures, or a cruiser for leisurely rides.
- **Rock climbing (indoor):** Okay, so technically we're not going high, but indoor rock climbing offers a thrilling workout challenging your strength, coordination, and problem-solving skills.

Constrained decoding: logit manipulation

What if we set $P(y_j = \text{"climbing"} \mid X, y_1, \dots, y_{j-1})$ to be 0?

-  Easy to implement: just add a big negative to the logit before the softmax!
-  Bad if there are a lot of synonyms
-  Bad if the tokens we restrict could be used in "allowed" ways
-  Bad if we generate other related terms before the restricted term

Constrained decoding: sample-then-rank (or reject)

Generate a set of sequences S

```
for  $s_i$  in  $S$ :  
    if  $s_i.is\_about\_climbing()$ :  
        discard( $s_i$ )
```

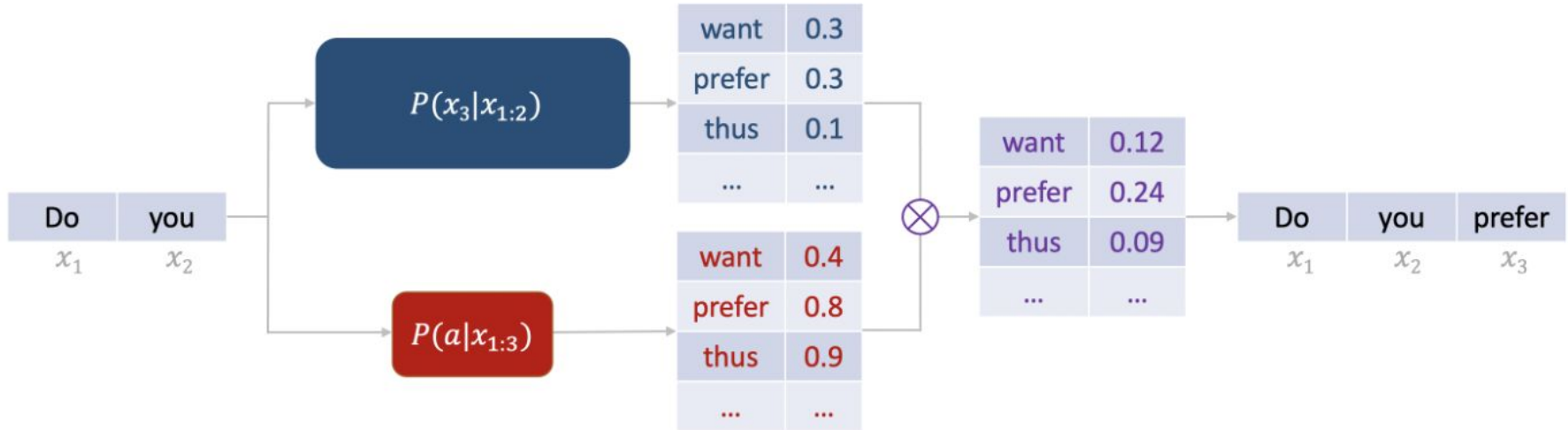


Easier to check if the full sequence violates the constraint



Expensive (i.e. slow), might even need to re-generate

Constrained decoding: FUDGE (Yang & Klein, 2021)



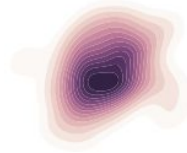
Constrained decoding via... RLHF?

Aligning an LM with human preferences is Bayesian inference



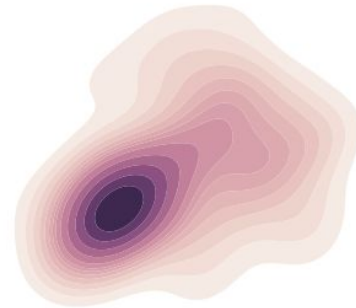
Prior: original LM

$$\pi_0(x)$$



Evidence: reward model

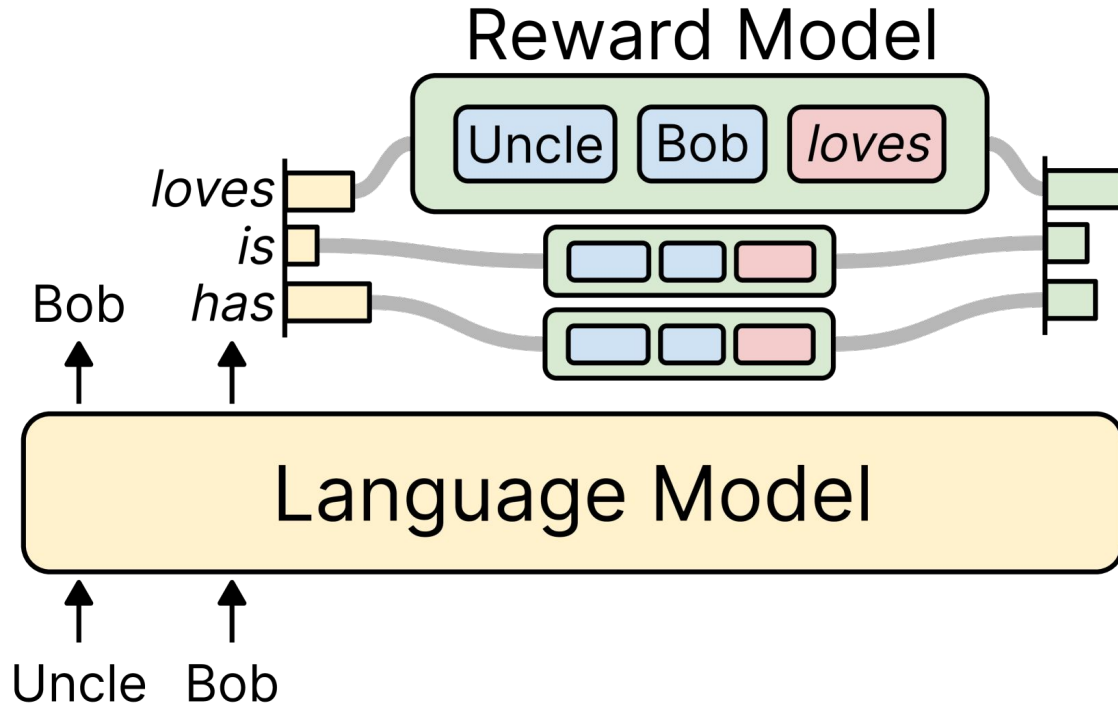
$$\exp(r(x))$$



Posterior: aligned LM

$$\pi^*(x) \propto \pi_0(x) \exp(r(x))$$

Reward-augmented decoding



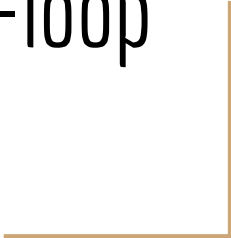
Modify probabilities by factoring in the estimated final reward of each sequence



Previous: constrained generation

up next:

Human-in-the-loop
decoding



Human-in-the-loop decoding: interleaved text

Choose when to insert model-generated text versus human continuation

Optionally, edit model-generated text before continuing

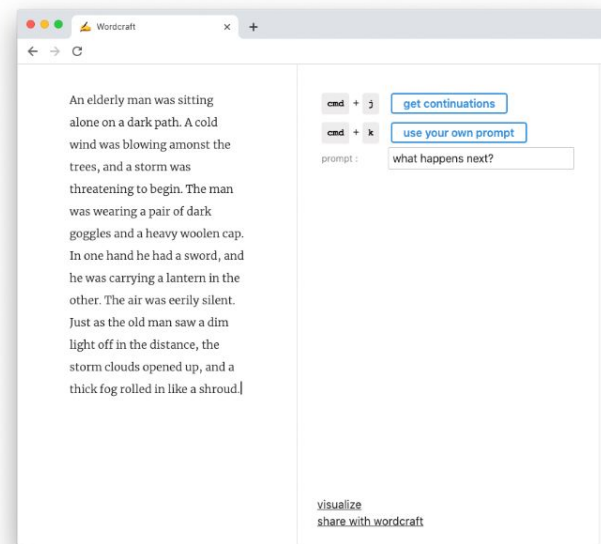


Figure from [Yuan et al \(2022\)](#)

Human-in-the-loop decoding: fine-grained replacement

User chooses the point to intervene, adds additional constraints (e.g. “more descriptive”, “four words”)

This can be accomplished with

- **input manipulation:** instructions in a prompt
- **modeling changes:** training an infilling model
- **decoding changes:** adding constraints to decoding to be more consistent with following text

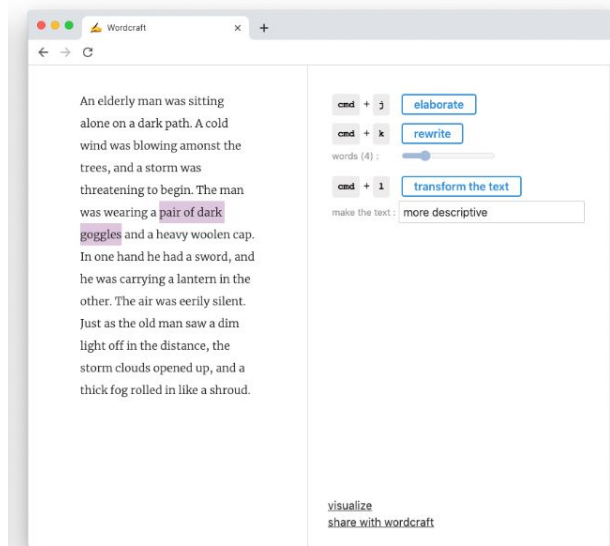


Figure from [Yuan et al \(2022\)](#)

Human-in-the-loop decoding: choosing outputs

Provide multiple options...

Cade hated the quiet. It bothered him to a certain degree that no matter how much he tried to break out of his shell, he couldn't force himself to make friends. He was sure at least one person in the world was similar to him, maybe even more than one, but where were they? He'd been looking for them most of his life after all.

He could not find a way to connect to other people: no way to make conversation. Even during their only shared, brief, conversation he could only stutter and stumble.

He was sure at least one person in the world was similar to him, maybe even more than one, but where were they? He'd been looking for them most of his life after all.

So when he finally meets someone like himself, he is excited about sharing secrets, dreams and fears.

He didn't know if they just couldn't be understood or if he just didn't find them interesting. Either way, it didn't matter anymore. His best friend passed away last night [show prompt details](#).

31 words

(continuation)

or the option to regenerate

You
Write a 2-sentence horror story.

ChatGPT
In the pitch-black silence of the night, I heard my own voice whispering my name from the depths of the closet. As I turned to flee, the door slammed shut, and a chilling laughter echoed, revealing that I was not alone.

Regenerate

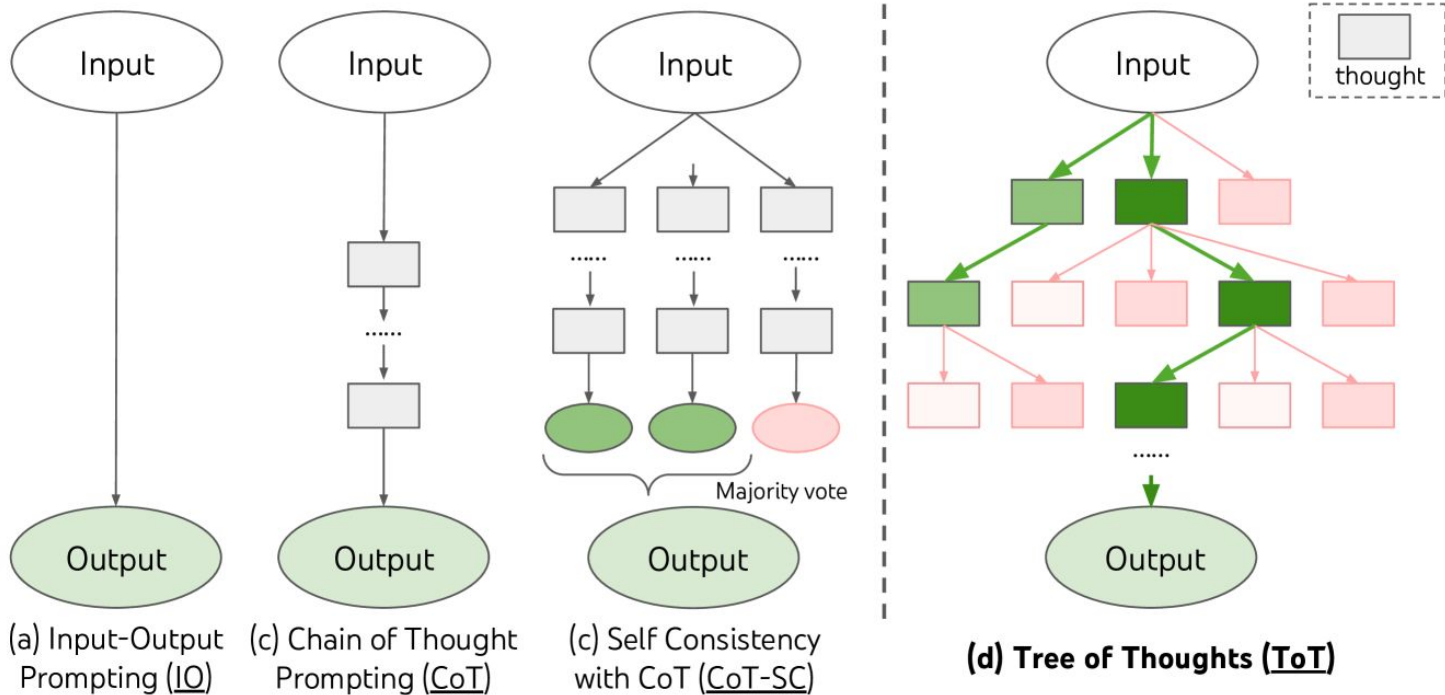
You
Write a 2-sentence horror story.

Regenerate draft

I dug up the antique mirror, excited to add it to my collection. Its inscription promised "eternal beauty," but the reflection staring back had different plans.

Left figure from [Yuan et al \(2022\)](#)

Model-in-the-loop decoding: Tree of Thought





Previous: human-in-the-loop

up next:

practical
considerations



Practical considerations: speed (speculative decoding)

```

[START] japan ' s benchmark bond n
[START] japan ' s benchmark nikkei 22 75
[START] japan ' s benchmark nikkei 225 index rose 22 76
[START] japan ' s benchmark nikkei 225 index rose 226 . 69 7 points
[START] japan ' s benchmark nikkei 225 index rose 226 . 69 points , or 0 1
[START] japan ' s benchmark nikkei 225 index rose 226 . 69 points , or 1 . 5 percent , to 10 , 9859
[START] japan ' s benchmark nikkei 225 index rose 226 . 69 points , or 1 . 5 percent , to 10 , 989 . 79 in
[START] japan ' s benchmark nikkei 225 index rose 226 . 69 points , or 1 . 5 percent , to 10 , 989 . 79 in tokyo late
[START] japan ' s benchmark nikkei 225 index rose 226 . 69 points , or 1 . 5 percent , to 10 , 989 . 79 in late morning trading . [END]

```

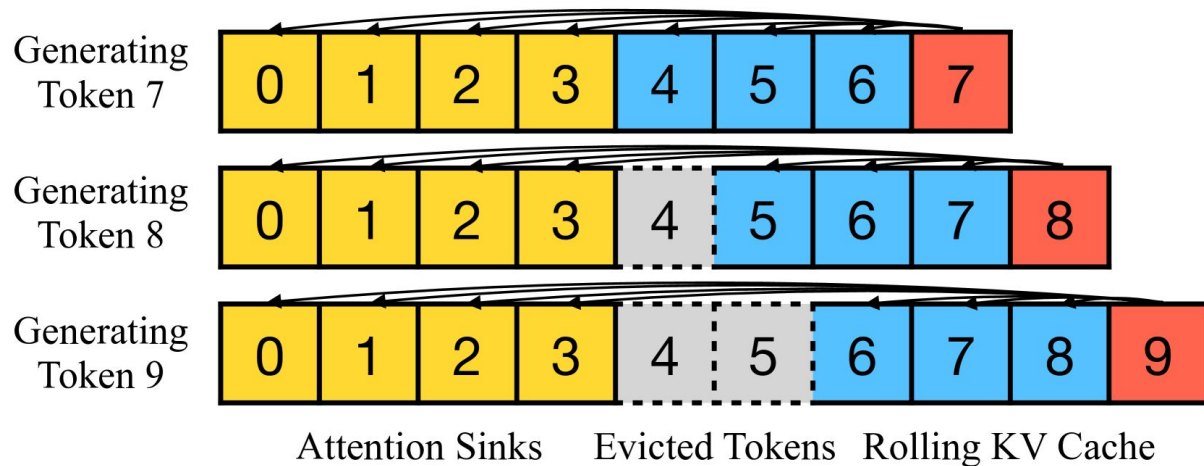
Propose candidates with small model, accept/reject candidates with larger model

Practical considerations: speed (attention sinks)

How do we keep generating quickly when we have more and more context to condition on?

Sliding windows: performance drops quickly

Alternative: attn sinks



Libraries for decoding (and fast inference)



disco

- + Many methods are implemented in HuggingFace, fairseq2, jax, etc

Outlines 



Summary: two levels of decoding

The model provides a distribution $P(y | X)$

1. **At each decoding step:** choose a function $f(P(y | X))$ to manipulate the next-token distribution
2. **Over the full decoding process:** choose a function $g(s)$ to choose between (full or partial) sequences generated from $f(P(y | X))$

Takeaways: decoding methods

You can use decoding methods to control features of the output

- Match certain constraints
- Factor in a reward function or data source
- You can do more expensive decoding to compensate for a worse model... up to a point

Different methods have tradeoffs in quality, diversity, and inference speed

- Sampling is fast and diverse but can be lower-quality
- More restricted sampling and MAP methods are higher-quality but less diverse
- MBR is high quality but slow

**Your responsibility to make design decisions doesn't stop when the model is trained!
Letting your libraries pick "sensible defaults" can leave performance on the table.**