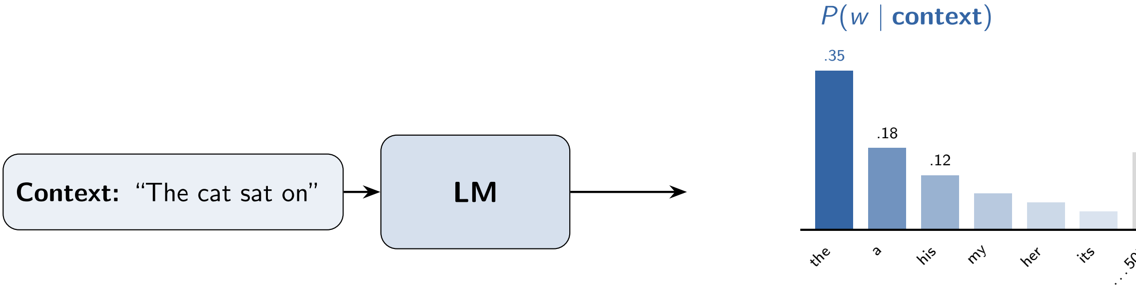
Decoding Strategies

Greedy · Beam Search · Temperature · Top-k · Top-p

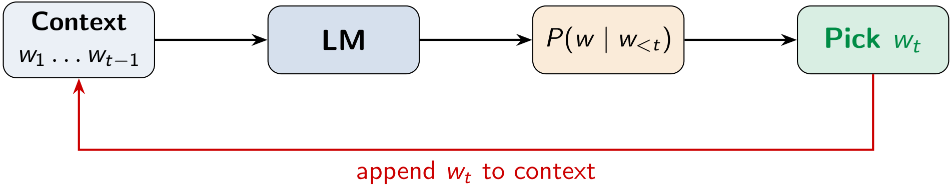
The generation problem



How do we pick the next token?

Autoregressive generation

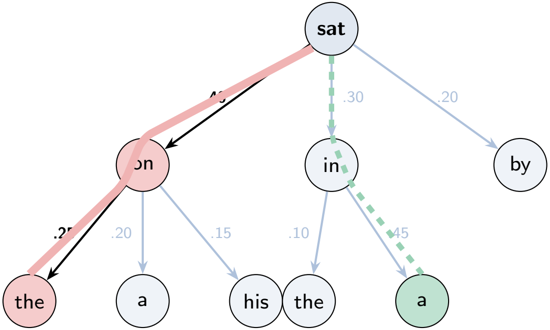
$$P(w_1, w_2, \ldots, w_T) = \prod_{t=1}^{T} P\big(w_t \mid w_1, \ldots, w_{t-1}\big)$$



Every decoding method differs **only** in the “Pick w_t” step.
The model and the loop are always the same.

Greedy search

$$w_t = \arg\max_{w} \; P(w \mid w_1, \ldots, w_{t-1})$$



Greedy picks the best token at each step, but can miss the **globally** best sequence.

Greedy path

$.40 \times .25 = .10$

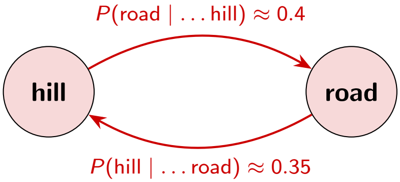
Better path

$.30 \times .45 = .135$

Greedy search: the repetition trap

Prompt: "Once upon a time, there was a little cat who"

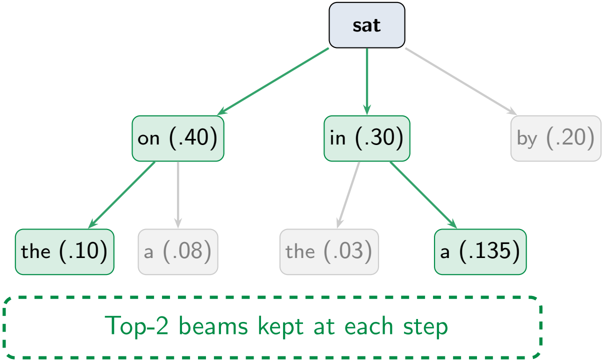
Greedy output: lived in a small house. The house was on the hill. The hill was on the road. The road was on the hill. The hill was on the road. The road was on the hill. The hill was on the . . .



Once in a high-probabil
greedy search **never e**

Each token is locally c
but the sequence is deg

Beam search ($B = 2$)



Beam search keeps the B most probable *partial sequences*.

Length-normalized score:

$$\text{score} = \frac{1}{T^{\alpha}} \sum_{t=1}^{T} \log P(w_t \mid w_{<t})$$

$\alpha \approx 0.6$–0.7 prevents preference for short sequences.

Beam search: trade-offs

Works well for:

- Machine translation
- Summarization
- Any task with a "right answer"
- Constrained generation

These tasks want the **most likely** sequence — beam search finds it.

Struggles with:

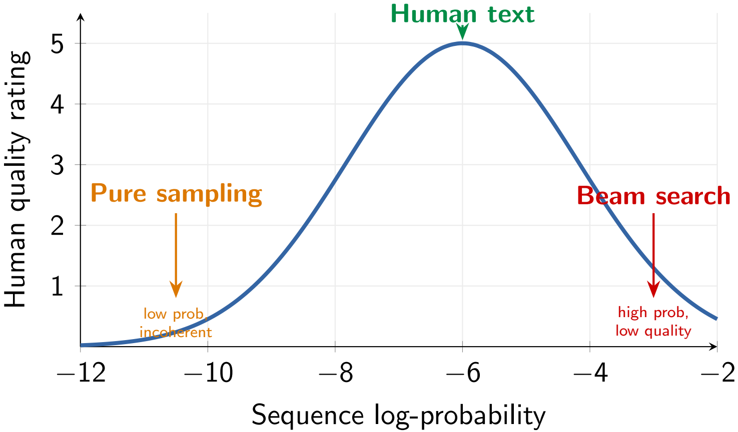
- Open-ended text / stories
- Dialogue / chat
- Creative writing

Output is "safe" but **boring**: generic, repetitive, lacks surprise.

Humans rate beam search text as less natural than sampling.

Repetition fix: n-gram penalty — if generating token w_t would create an n-gram that already appeared, set $P(w_t) = 0$.

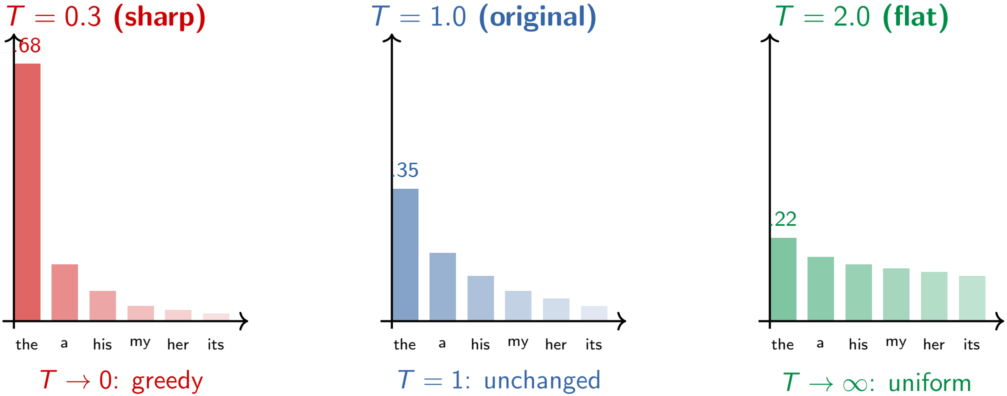
The probability trap



The highest-probability text is **not** the best text.
Human language occupies a **sweet spot** of moderate probability.
— Holtzman et al., "The Curious Case of Neural Text Degeneration" (2019)

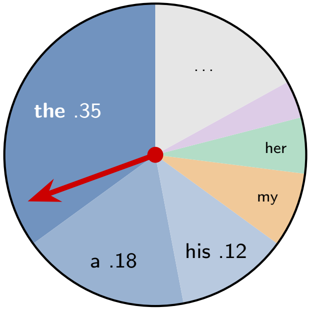
Temperature scaling

$$p_i = \frac{\exp(z_i \,/\, T)}{\sum_j \exp(z_j \,/\, T)}$$



Sampling from the distribution

$$w_t \sim P(\cdot \mid w_1, \ldots, w_{t-1})$$



Sample = "spin the wheel"

The good: introduces diversity, avoids repetition loops, more human-like

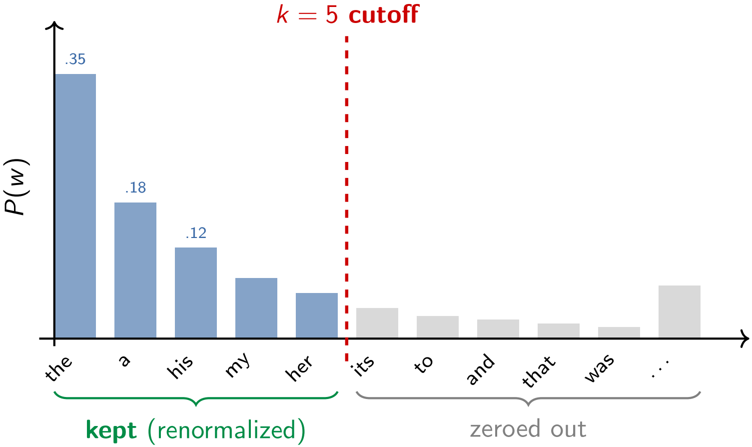
The bad: low-probability tokens ("banana", "⟨unk⟩") occasionally get sampled, **derailing** the entire sequence

"The cat sat on quantum and then refrigerator began to oscillate the ..."

We need to sample, but from
a **truncated** distribution.
This motivates top-k and top-p.

Top-k sampling

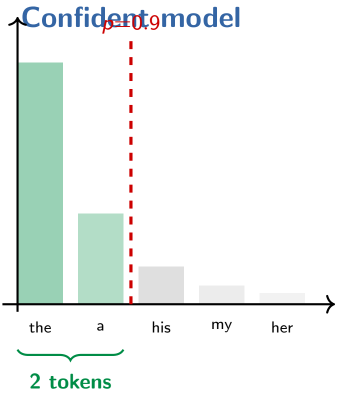
Idea: Keep only the k most probable tokens, zero out the rest, renormalize.

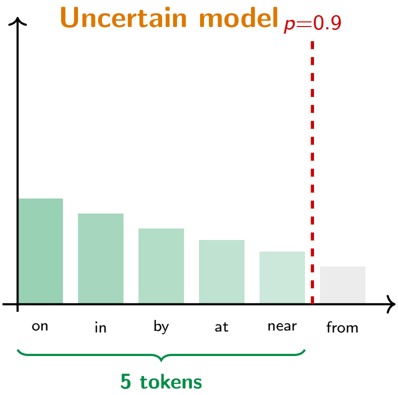


Problem: Fixed k doesn't adapt. When the model is **confident**, k=50 includes garbage. When the model is **uncertain**, k=10 cuts off good

Top-p (nucleus) sampling

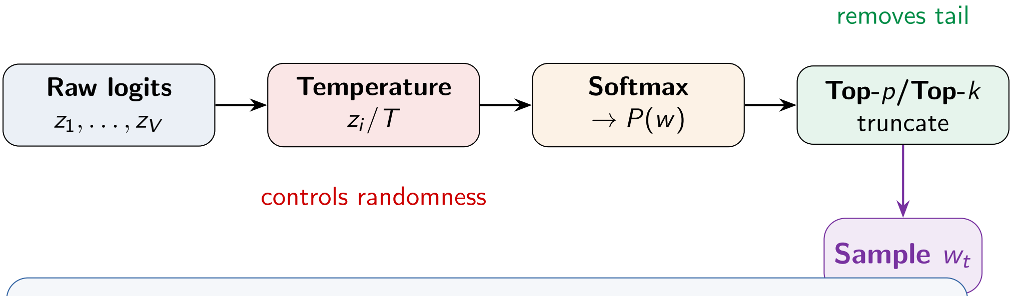
Idea: Find the smallest set V_p such that $\sum_{w \in V_p} P(w) \geq p$.





Top-p **adapts** the number of candidates to the model's confidence.

Combining the knobs



Common combinations:

- Temperature T reshapes the distribution *before* truncation
- Top-k and top-p can be used *together* (intersection of both filters)
- Greedy $=$ $T \to 0$, or equivalently top-k with k=1

Comparison of decoding strategies

Method	Deterministic?	Best for	Main weakness
Greedy	Yes	Quick baseline	Repetitive, local optima
Beam search	Yes	Translation, summary	"Boring" open-ended text
Temperature	—	*Modifier, not standalone*	Just a knob
Top-k	No	Open-ended generation	Fixed k doesn't adapt
Top-p	No	Open-ended generation	Slightly slower than top-k

In practice, most LLM APIs use **temperature + top-**p together.
Beam search is reserved for structured tasks (translation, summarization).
Greedy ($T=0$) is used when you want fully deterministic output.

Practical: typical API parameters

```
response = client.generate(
  prompt = "...",
  temperature = 0.7,
  top_p = 0.9,
  top_k = 50,   # optional
  max_tokens = 256
)
```

Creative writing

$T = 0.8$–1.0
top-p = 0.9–0.95

Factual / Code

$T = 0.0$–0.3
or greedy ($T{=}0$)

Translation

Beam search, $B = 4$–6
length penalty $\alpha \approx 0.6$

Questions?

Next: Evaluation — Perplexity, BLEU, ROUGE