Sampling in NLP: A Beginner's Manual

Temperature, Top-k, Top-p

Compiled for learners

November 19, 2025

Abstract

This manual explains sampling techniques commonly used when decoding language models: **temperature sampling**, **top-k** sampling, and **top-p (nucleus)** sampling. It starts from first principles, motivates why sampling is needed in NLP, gives intuition and exact formulas, and provides clear diagrams and worked examples. The content is aimed at beginners and is intentionally verbose and explicit: no steps are skipped. Colors, boxes, and diagrams are used to guide the eye and emphasize definitions.

How to use this manual

- Read sections in order: the material builds from softmax basics to sampling heuristics.
- Diagrams are integrated into the text; examine the captions for numerical details.
- All math steps are shown; if you want a shorter summary, go to the final page "Cheat Sheet".

1 Why do we need sampling in NLP?

High-level goal: Given a trained language model, we want to generate text. The model assigns *scores* (called *logits*) to each possible next token. To produce actual tokens we must convert these scores into a distribution and pick tokens. **Sampling** is the process of converting model outputs to actual tokens.

1.1 What is the model predicting?

A language model predicts the conditional probability of the next token given the context: $P(\text{token}_t \mid \text{context})$. Internally, neural models output *logits* z_i for each token index i. These logits are unnormalized scores. To get probabilities we apply the *softmax* function (explained below).

1.2 Why not always pick the highest probability token?

- **Greedy decoding** (always pick the argmax) yields repetitive, deterministic, and often dull text.
- **Sampling** allows diversity — the ability to produce creative, varied outputs — while still favoring high-probability tokens.
- Different sampling methods trade off *creativity* and *coherence*.

2 Mathematical foundation: softmax and logits

2.1 Definition: softmax

Given logits $z_1, \ldots, z_n$, the softmax probability for token i is

$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}}.$$

This converts arbitrary real-valued scores into a valid probability distribution.

Important: Softmax is *invariant* to adding a constant to all logits: replacing z_i with $z_i + c$ produces the same distribution. This is because the factor e^c cancels.

2.2 Temperature in softmax

The softmax can be modified to include a *temperature* parameter $T > 0$:

$$P(i; T) = \frac{\exp\left(\frac{z_i}{T}\right)}{\sum_{j=1}^{n} \exp\left(\frac{z_j}{T}\right)}.$$

Equivalently people sometimes write using inverse temperature $\tau = 1/T$:

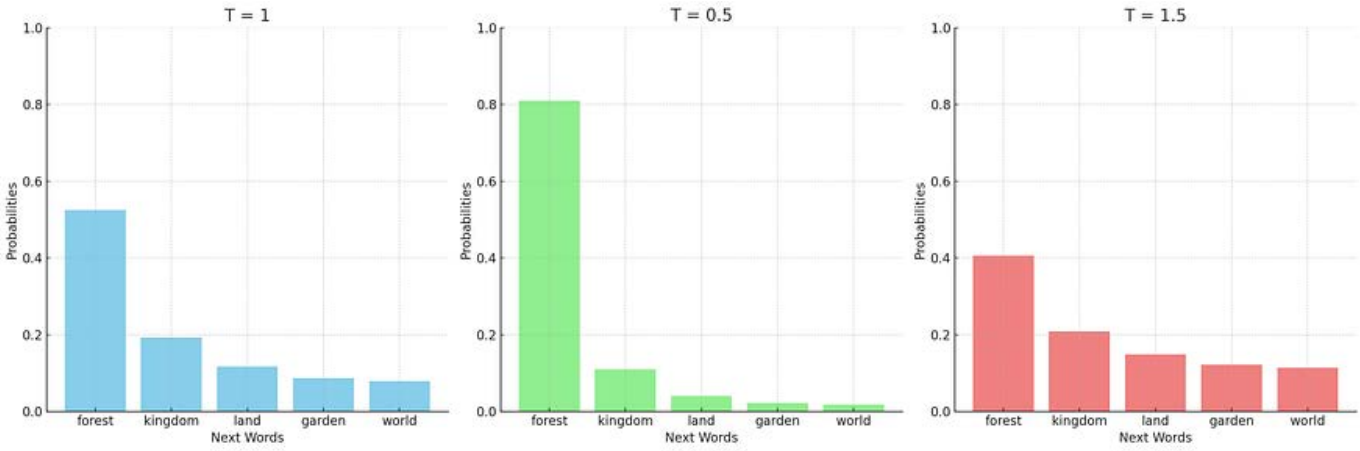
$$P(i;\tau) = \frac{\exp(\tau z_i)}{\sum_j \exp(\tau z_j)}.$$

Intuition (thermodynamics metaphor):

- $T \to 0$ (low temperature): the distribution becomes *sharp* — mass concentrates on the largest logit (greedy behavior).
- $T \to \infty$ (high temperature): the distribution approaches uniform — more random sampling.

2.3 Derivation: how temperature rescales logits

Let z_i be logits. Dividing by a small T (e.g. 0.5) amplifies differences between logits because z_i/T is larger in magnitude; exponentials accentuate these differences, so the top logit gains much more probability. Conversely, dividing by a large T shrinks differences.



Effect of Temperature (T) on Randomness:

When T>1:

Increasing T makes the distribution more uniform (flatter), increasing the randomness of the sampling. This happens because as T grows, the difference between the largest and smallest logits has less impact on the resulting probabilities, making less likely events more probable.

When T=1:

The model behaves normally, with no modification to the logits before applying the softmax function. The probabilities reflect the model's learned distribution.

When T<1:

Decreasing T makes the distribution sharper, reducing randomness. Lower temperatures increase the disparity between the higher and lower logits, making the model's predictions more deterministic (i.e., the highest logit value(s) dominate the probability distribution).

These effects can be seen in the following figure.

3 Visualizing temperature effects

We illustrate two intuitive visualizations:

1. The probability of a top token as a function of T for two logits.
2. Bar charts of full categorical distributions at several temperatures.

3.1 Plot: probability of top token vs temperature

Consider two logits $z_1 = 2,\ z_2 = 1$. Let $p_1(T) = \dfrac{e^{2/T}}{e^{2/T} + e^{1/T}}$. The plot below shows how p_1 varies with T.

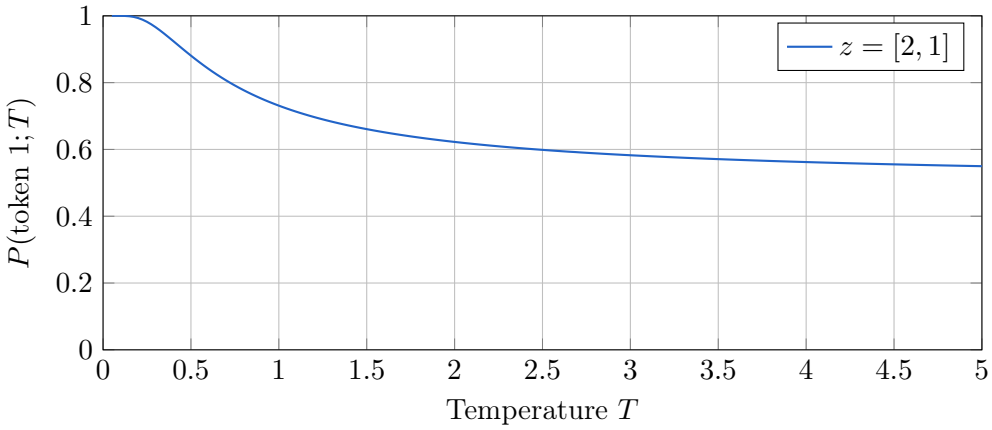


Figure 1: As temperature increases, the top-token probability moves toward 0.5 (more random). As $T \to 0$, it approaches 1 (greedy).

3.2 Example bar charts for logits [3,1,0,-1]

We compute exact probabilities for temperatures $T \in \{0.5, 1, 2\}$ and show bar charts. Numbers (rounded) are precomputed to avoid numerical instability.

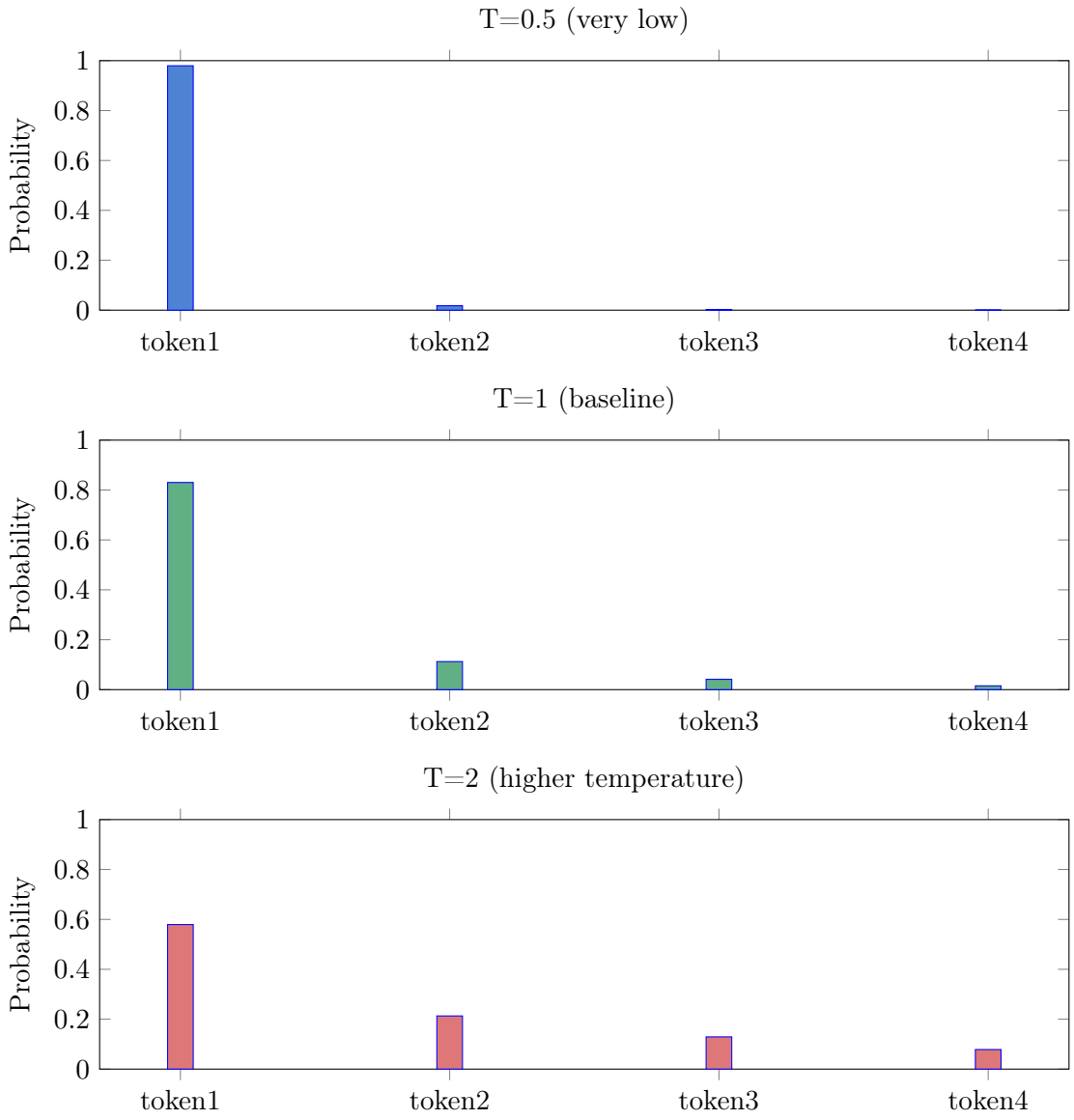


Figure 2: Bar charts showing how lower T concentrates mass on the top token, while higher T spreads mass. Values are rounded to 4 significant digits.

4 Practical sampling strategies

After converting logits into a distribution (with or without temperature), we need a rule for drawing a token. Popular options:

- **Greedy / Argmax**: pick token with highest probability. Deterministic.
- **Ancestral sampling**: sample directly from the distribution; randomness reflects the distribution.
- **Top-k sampling**: restrict distribution to the top-k tokens by probability, renormalize, and sample.
- **Top-p (nucleus) sampling**: take the minimal set of highest-probability tokens whose cumulative probability is at least p, renormalize, and sample.

4.1 Top-k: definition and intuition

Top-k (k is an integer):

1. Sort tokens by probability (descending).
2. Keep the first k tokens and set other probabilities to zero.
3. Renormalize the kept probabilities to sum to 1.
4. Sample from this reduced distribution.

Why use top-k? It prevents sampling from the long tail of extremely unlikely tokens (which can produce incoherent text), while still allowing diversity among the top k candidates.

4.2 Top-p (nucleus): definition and intuition

Top-p (nucleus) sampling, $0 < p \leq 1$:

1. Sort tokens by probability (descending).
2. Let S be the smallest set of top tokens such that the cumulative probability of S is at least p.
3. Keep tokens in S, set others to zero.
4. Renormalize the kept probabilities and sample.

Why top-p? Instead of fixing the number of candidates, top-p adapts to the shape of the distribution: when one token is dominant it keeps a small set; when the distribution is flat it keeps more tokens.

4.3 Diagrams: top-k vs top-p

We visualize the same baseline distribution (T=1, probabilities from earlier) and show the effect of top-k=2 and top-p=0.9.

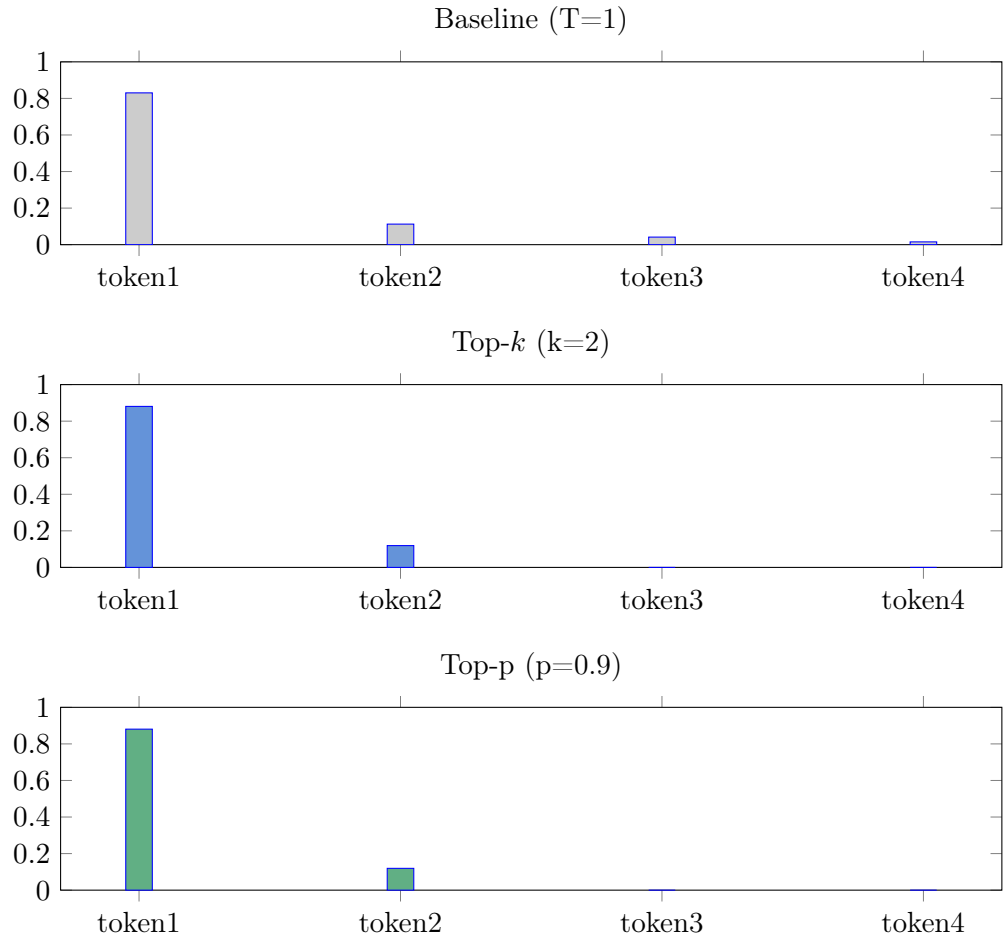


Figure 3: For this example baseline distribution, top-k=2 and top-p=0.9 select the same set: the first two tokens. The renormalized probabilities are shown.

5 Worked examples: putting it all together

5.1 Example 1: generating a short continuation

Suppose the model at a given timestep outputs logits for 6 tokens (indices A–F):

$$z = [3.0,\ 1.5,\ 1.0,\ 0.2,\ -1.0,\ -2.5].$$

We will compute probabilities for $T = 1$, then perform top-k and top-p selections.

5.1.1 Step 1: softmax (T=1)

Compute exponentials (rounded):

$$\begin{aligned} e^{3.0} &\approx 20.0855, \\ e^{1.5} &\approx 4.48169, \\ e^{1.0} &\approx 2.71828, \\ e^{0.2} &\approx 1.22140, \\ e^{-1.0} &\approx 0.36788, \\ e^{-2.5} &\approx 0.082085. \end{aligned}$$

Sum ≈ 29.9567. Probabilities:

$$p \approx [0.6704,\ 0.1496,\ 0.0907,\ 0.0408,\ 0.0123,\ 0.0027].$$

5.1.2 Step 2: top-k with k=3

Top 3 tokens by probability are A, B, C. Keep them and renormalize: Sum of kept = 0.6704 + 0.1496 + 0.0907 = 0.9107. Renormalized distribution:

$$p_{\text{top-3}} \approx [0.736,\ 0.164,\ 0.100,\ 0,\ 0,\ 0].$$

Then sample from this 3-way distribution.

5.1.3 Step 3: top-p with p=0.95

Sort and accumulate: A(0.6704) -> cumulative 0.6704. Add B -> cumulative 0.8200. Add C -> cumulative 0.9107. Add D -> cumulative 0.9515. At this point we reached $p = 0.95$, so we keep A,B,C,D. Renormalize and sample.

5.2 Example 2: effect of temperature on sampling

Using the logits from Example 1, compare sampling outcomes for $T = 0.7$ and $T = 1.5$ (qualitative description):

- At $T = 0.7$ differences are amplified; A's probability increases and sampling is more likely to pick A often.
- At $T = 1.5$ differences shrink; the distribution flattens, making it more likely to sample less probable tokens (B,C,D...), increasing diversity.

6 Practical recommendations and heuristics

Rules of thumb

- For coherent, safe outputs: use lower temperatures (e.g. $T \in [0.6, 1.0]$) with smallish top-k (like $k = 40$) or top-p (like $p = 0.9$).
- For creative writing: higher temperatures (e.g. $T \in [1.0, 1.5]$) or larger top-p (e.g. $p = 0.92$–0.98) produce more diversity.
- Avoid extremely low values (e.g. $T < 0.1$) which effectively yields deterministic outputs and can get stuck.
- Combine mechanisms: many systems use a temperature (to control sharpness) plus top-k or top-p (to clip the tail) simultaneously.

7 Common pitfalls and debugging

- Numerical instability: exponentiating large logits (or dividing by extremely small T) may overflow. A standard trick: subtract the max logit before exponentiating.
- Off-by-one in top-k: ensure $k \leq n$ and that you treat ties consistently.
- In top-p, sorting and cumulative sum must use the probabilities after applying temperature.

8 Exercises (for practice)

1. Given logits [2, 1, 0, -1, -2], compute softmax probabilities for T=0.5, 1.0, 2.0.
2. Implement top-p selection in code: sort, accumulate until threshold, renormalize. Test with several p values and distributions.
3. Experiment: on a small pretrained model, compare outputs with greedy, temperature-only sampling, top-k-only, and combined temperature+top-p. Record sample diversity and coherence.

Cheat sheet

- **Softmax:** $P(i) = \frac{e^{z_i}}{\sum_j e^{z_j}}$.

- **Temperature:** $P(i;T) = \frac{e^{z_i/T}}{\sum_j e^{z_j/T}}$. Low T = sharp, high T = flat.
- **Top-k:** keep top k probs, renormalize, sample.
- **Top-p:** keep smallest set with cumulative prob $\geq p$, renormalize, sample.

This manual was generated to be beginner-friendly and self-contained. If you want a PDF compiled from this LaTeX file or want adjustments (more diagrams, different colors, or additional exercises), tell me which parts to change.