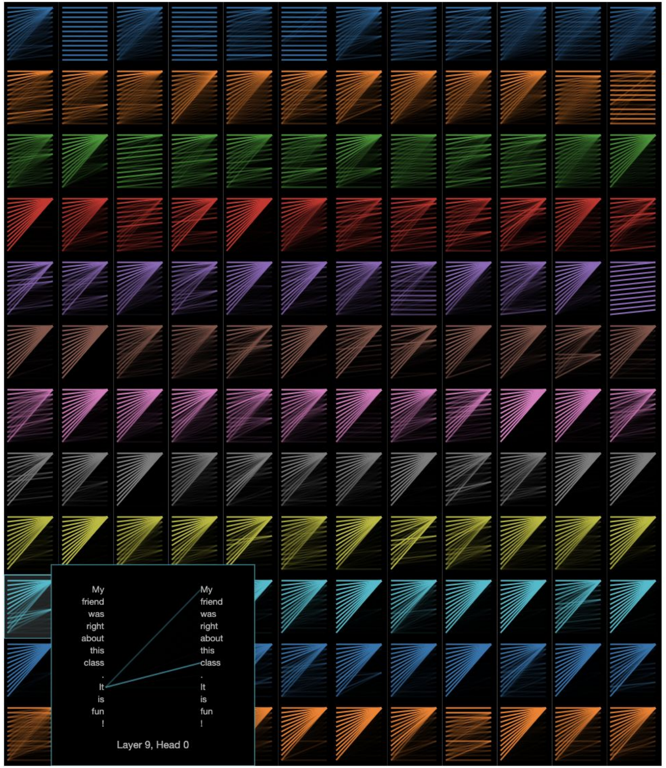
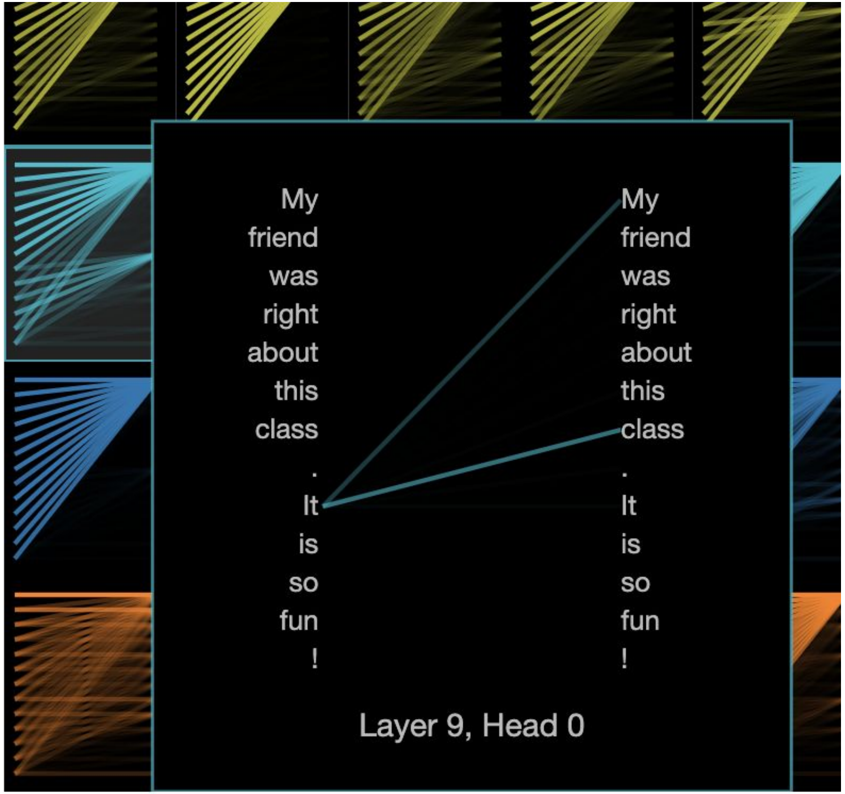
Visualizing GPT Attention

Multi-headed Self-Attention

“My friend was right about this class. It is so fun!”

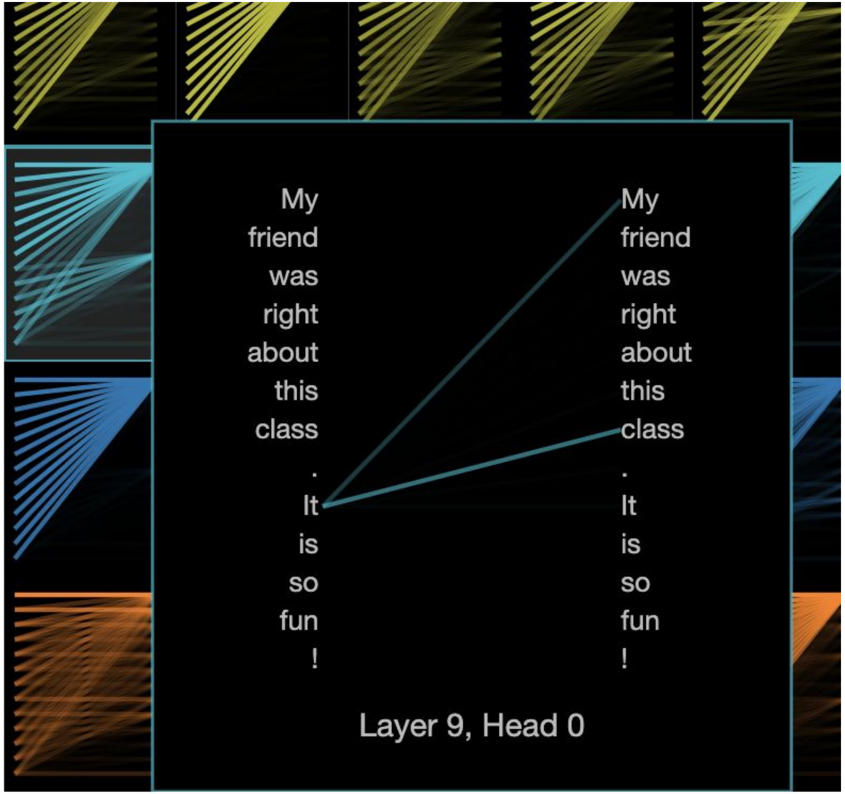




Multi-headed Self-Attention

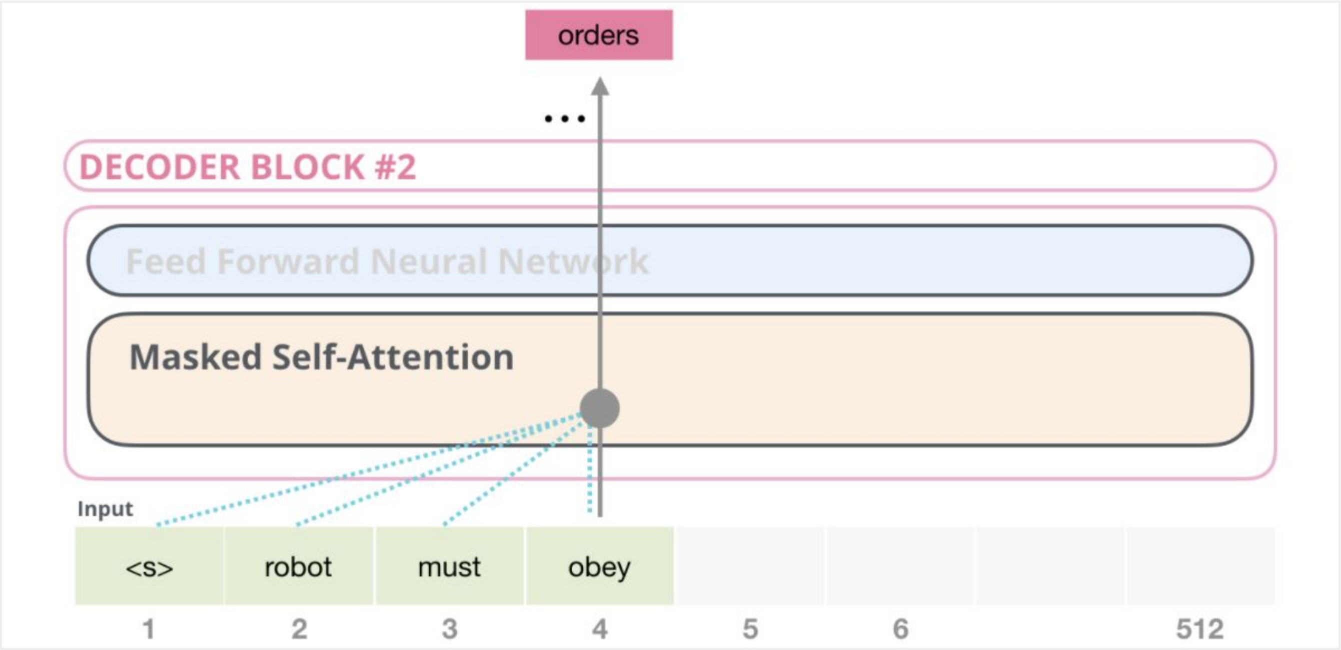
Notice how tokens cannot attend to tokens that came before. This is because of the masking.

Said another way, notice that no lines are drawn from tokens on the left to tokens that come afterwards on the right



How GPT predicts in real-time (inference)

Next token predictions happen one token at a time. This slows down GPT when predicting in real time



Source: https://jalammar.github.io/illustrated-gpt2/

GPT Inference Parameters

Parameters for inference

temperature (float) - Lower (below 1) makes the model more confident and less random. Higher values make generated text more random.

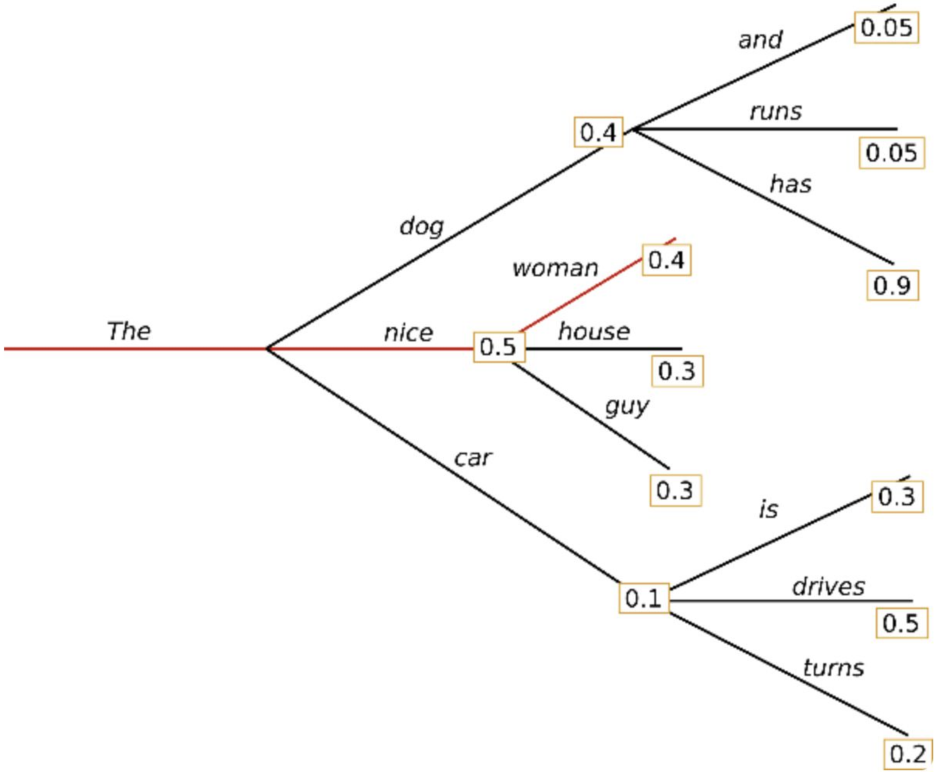
top_k (int) - How many tokens it considers when generating. 0 to deactivate

top_p (float) - only considers tokens from the top X% of confidences

beams (int) - How many tokens out should we consider

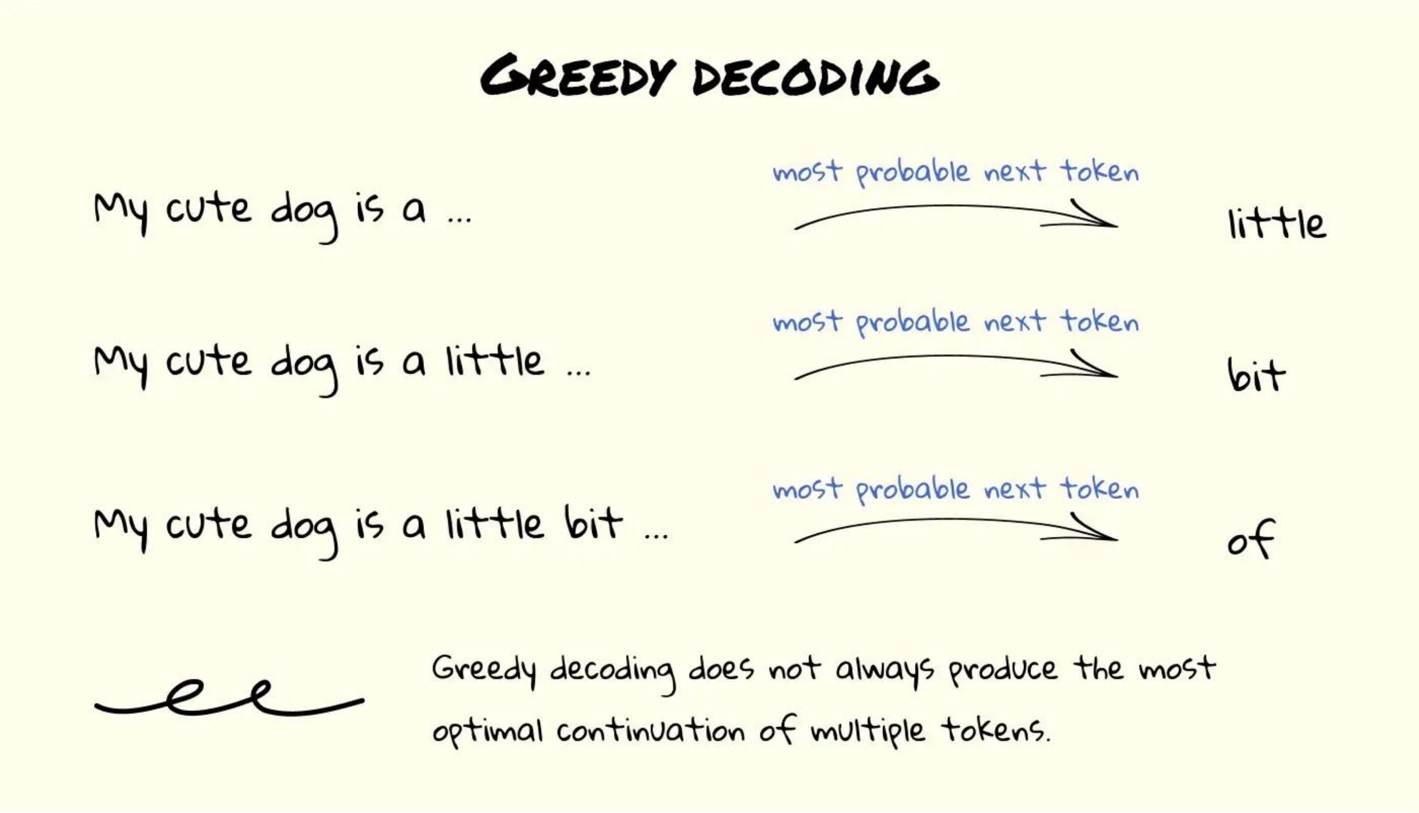
do_sample (bool) - If True, randomness is introduced in selection

Normal “Greedy” Search



Source: https://huggingface.co/blog/how-to-generate

Normal “Greedy” Search



GREEDY DECODING

My cute dog is a ... → most probable next token → little

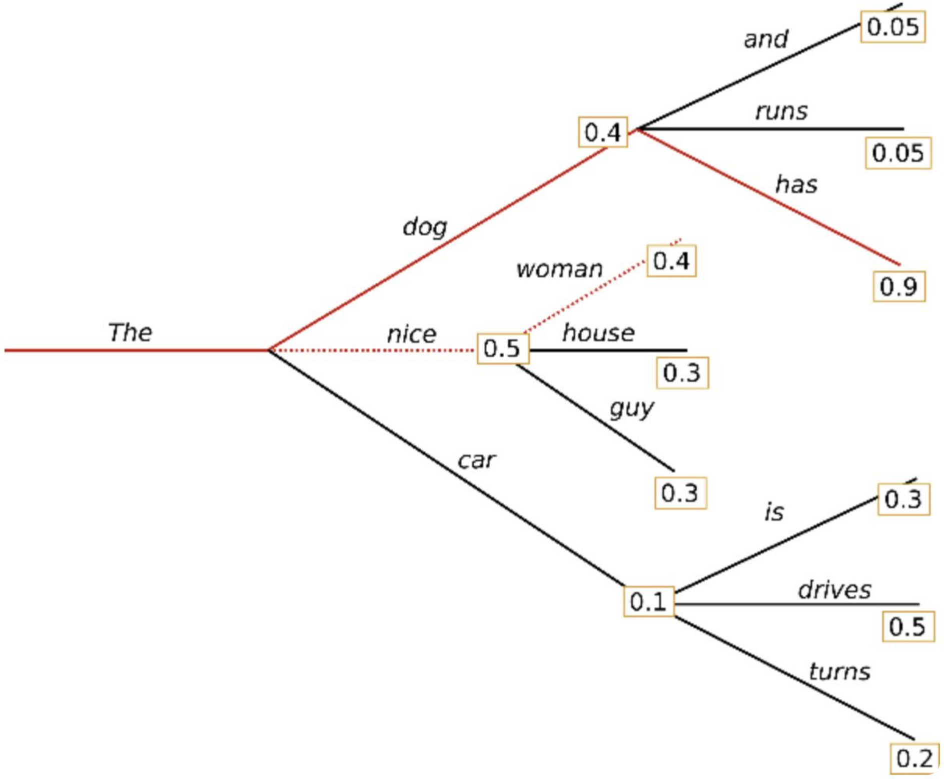
My cute dog is a little ... → most probable next token → bit

My cute dog is a little bit ... → most probable next token → of

Greedy decoding does not always produce the most optimal continuation of multiple tokens.

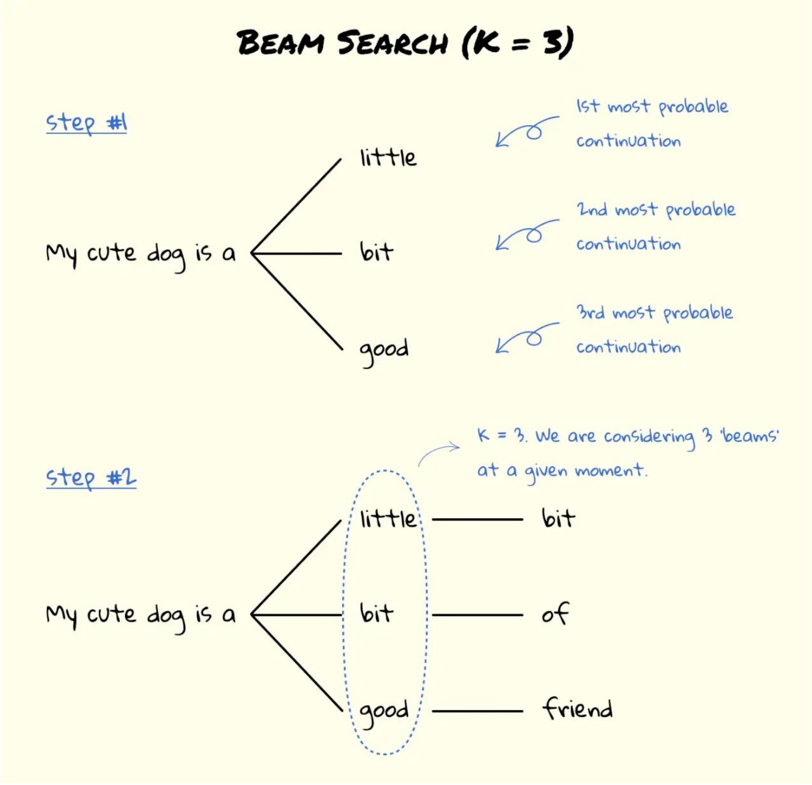
Source:
https://towardsdatascience.com/the-power-of-constrained-language-models-cf63b65a035d

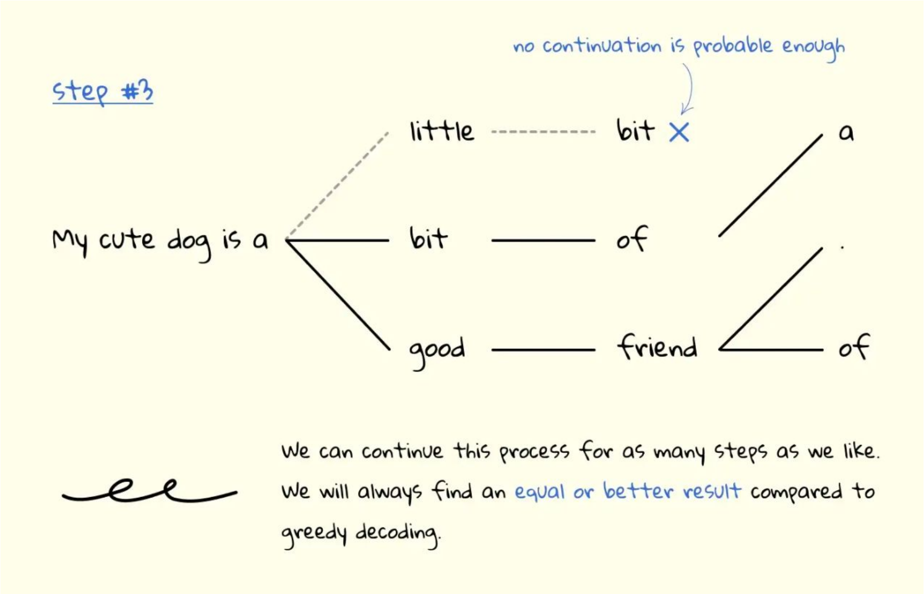
Beam Search



Source: https://huggingface.co/blog/how-to-generate

Beam Search

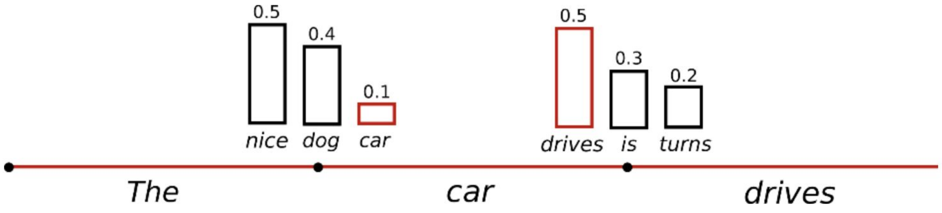

BEAM SEARCH (K = 3)
step #1
My cute dog is a
little
bit
good
1st most probable continuation
2nd most probable continuation
3rd most probable continuation
step #2
K = 3. We are considering 3 'beams' at a given moment.
My cute dog is a
little
bit
good
bit
of
friend



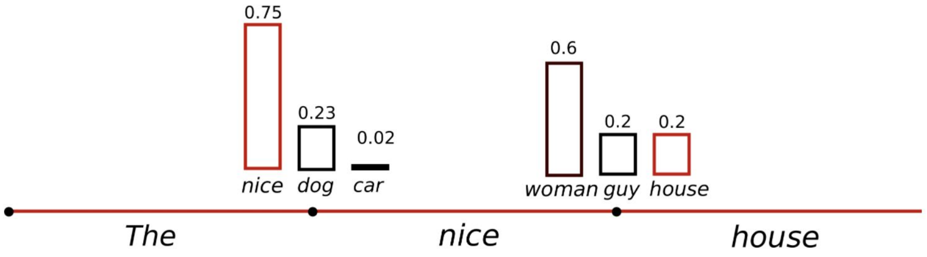
We can continue this process for as many steps as we like. We will always find an equal or better result compared to greedy decoding.

Source: https://towardsdatascience.com/the-power-of-constrained-language-models-cf63b65a035d

Temperature



Normal probability distribution

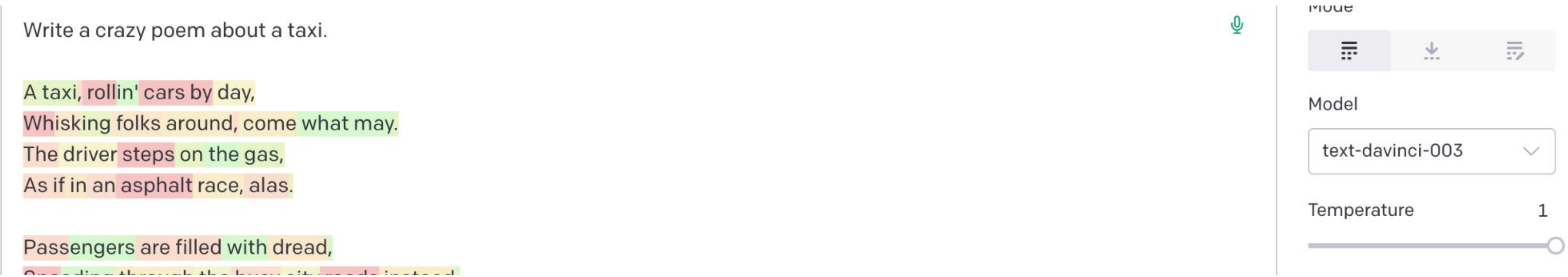


With temperature < 1, probabilities are “sharper”

Source: https://huggingface.co/blog/how-to-generate

Temperature

More randomness



Write a crazy poem about a taxi.

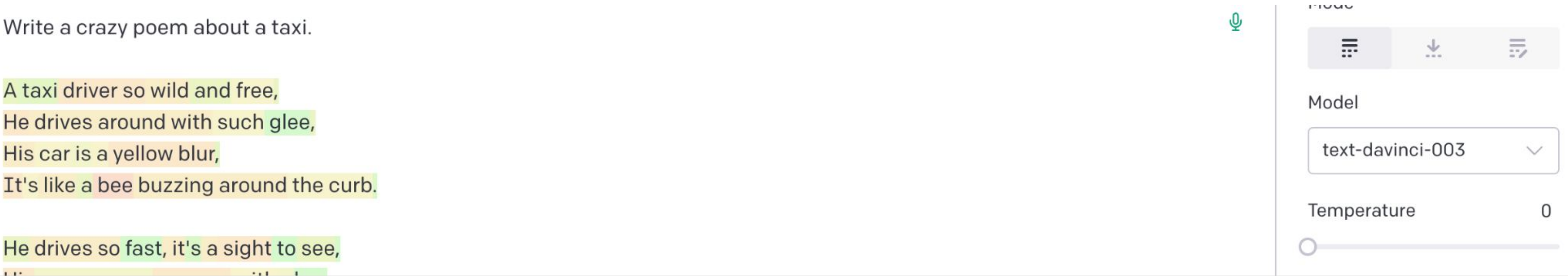
A taxi, rollin' cars by day,
Whisking folks around, come what may.
The driver steps on the gas,
As if in an asphalt race, alas.

Passengers are filled with dread,
Speeding through the busy city roads instead,

Model: text-davinci-003

Temperature: 1

Less randomness



Write a crazy poem about a taxi.

A taxi driver so wild and free,
He drives around with such glee,
His car is a yellow blur,
It's like a bee buzzing around the curb.

He drives so fast, it's a sight to see,

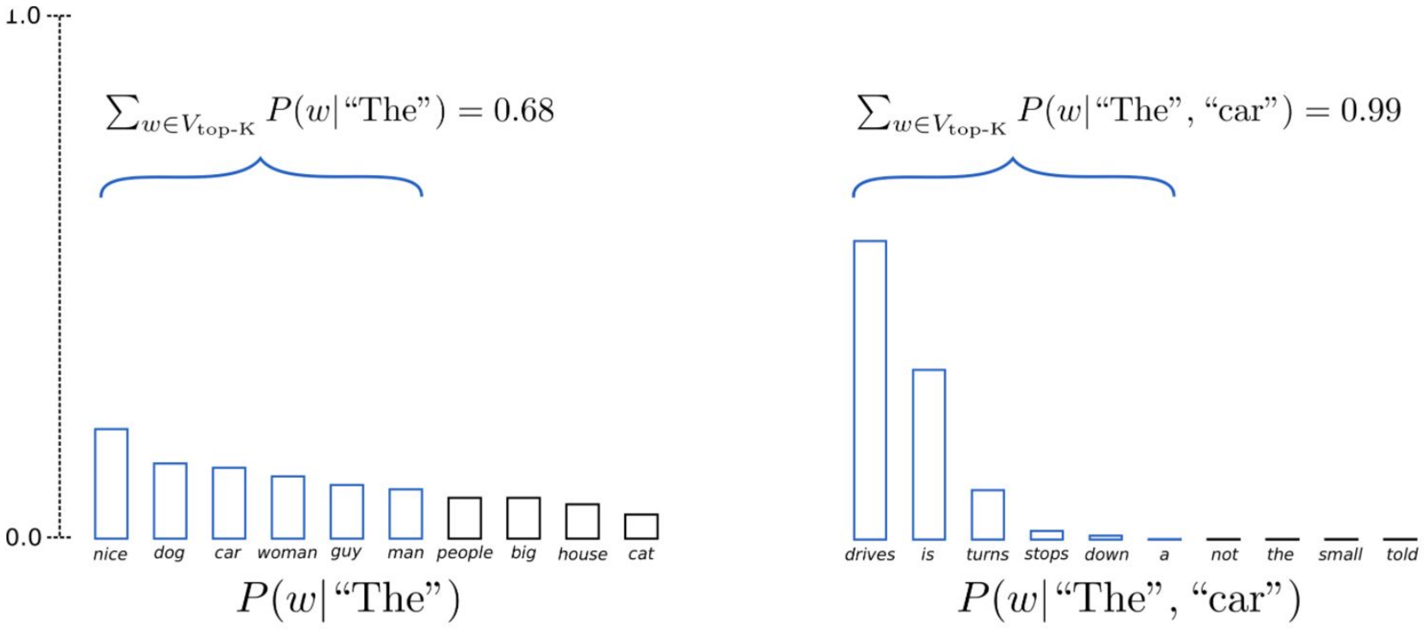
Model: text-davinci-003

Temperature: 0

Top-K Sampling

With top_k=6

we readjust probabilities to be sharper for the top 6 possible tokens

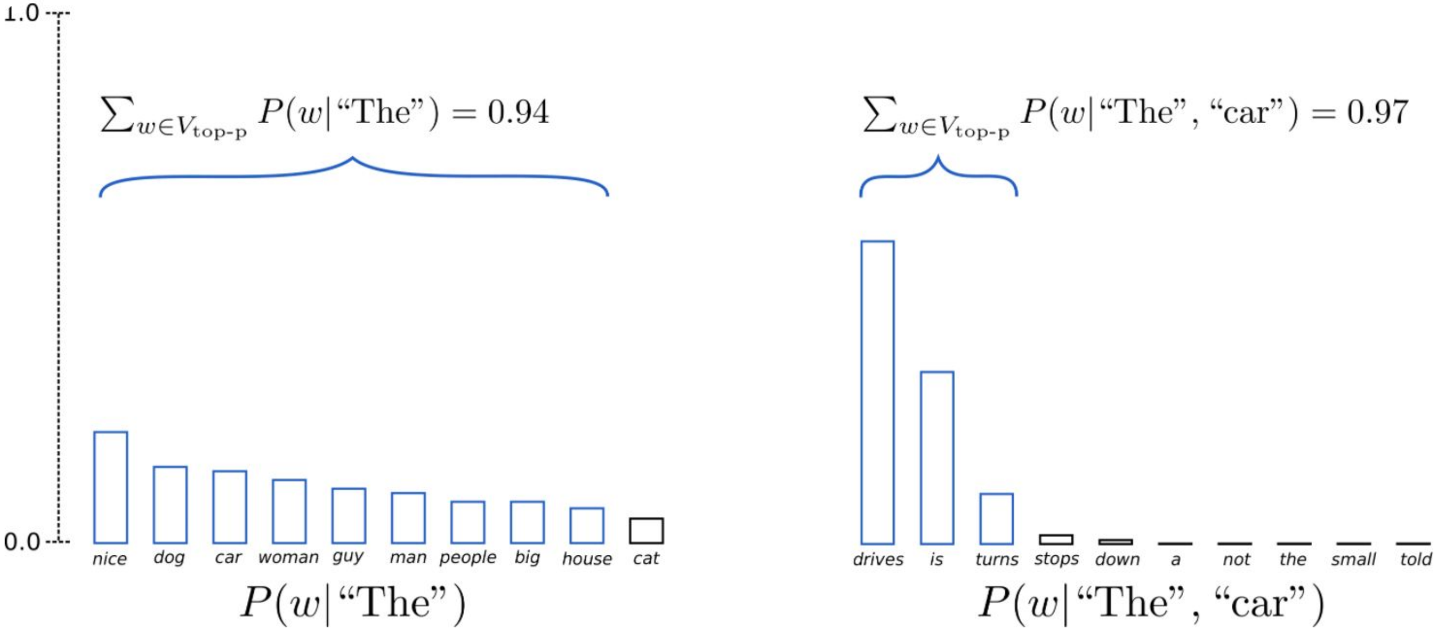


Source: https://huggingface.co/blog/how-to-generate

Top-P Sampling

With top_p=0.92

we readjust probabilities among the minimum number of tokens that **exceed** the given parameter



Source: https://huggingface.co/blog/how-to-generate

Narrow Generation Ability

If GPT fits too much to the training examples, it is very susceptible to becoming overfit. This means that it is effectively memorizing the text rigidly and does not know how to generalize to new examples.

This is called the **narrow generation ability** of a pre-trained deep learning model. The model lacks the ability to generalize broadly and cannot be assumed to be “reasoning” or “learning truth”

Source: https://arxiv.org/abs/1911.01547