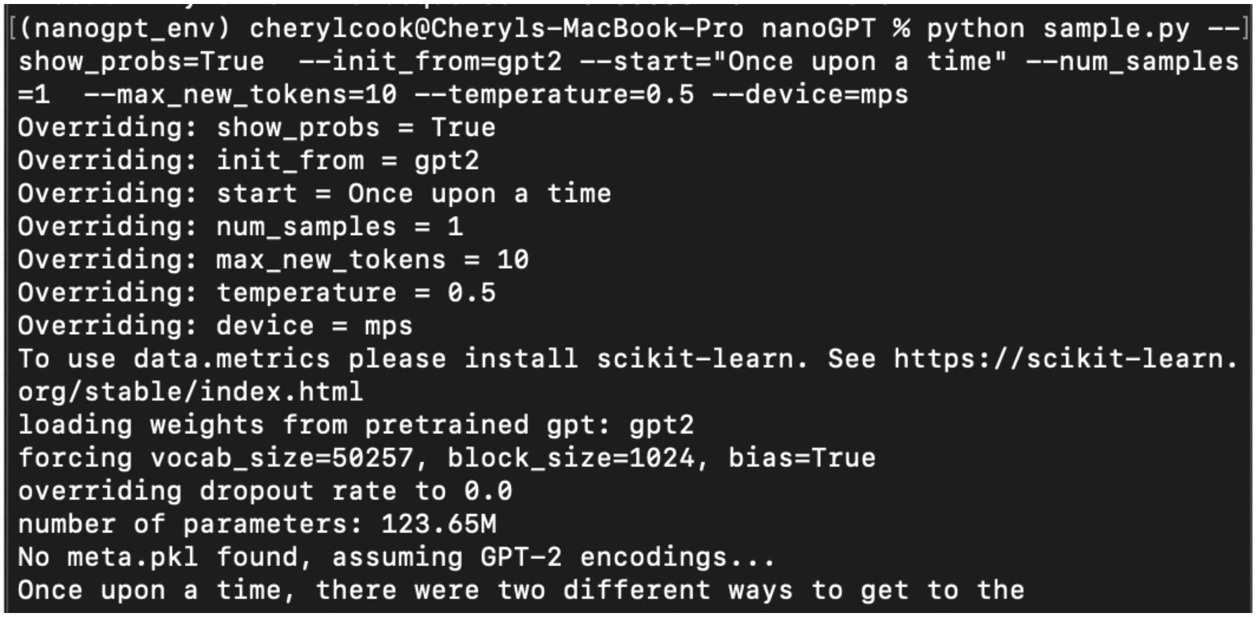
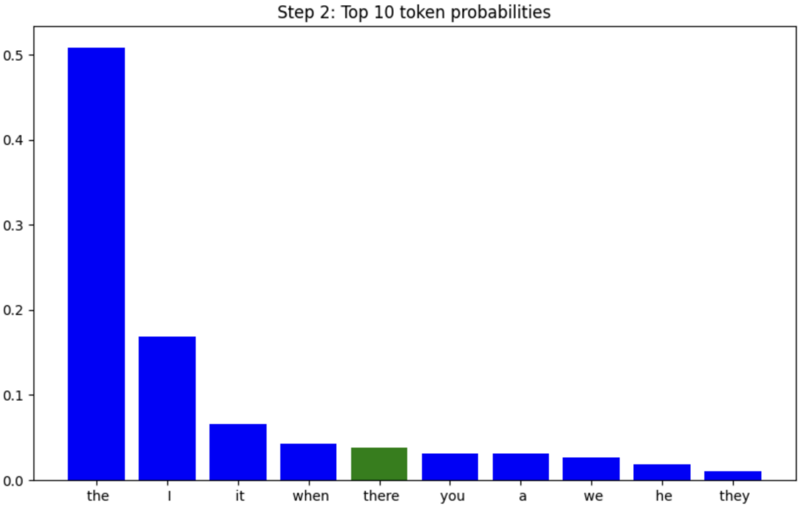
Effect of temperature on token selection:
All arguments remain the same across each run, with only the temperature value being changed.

Temperature = 0.5:

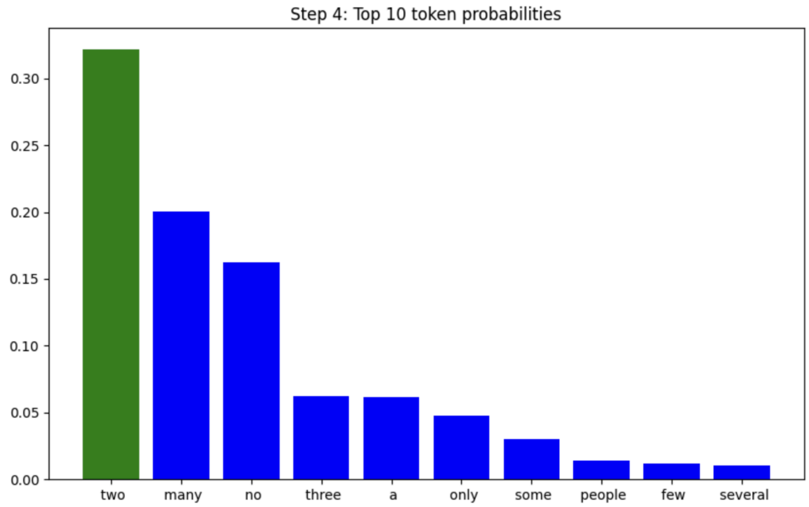
Once upon a time, there were two different ways to get to the

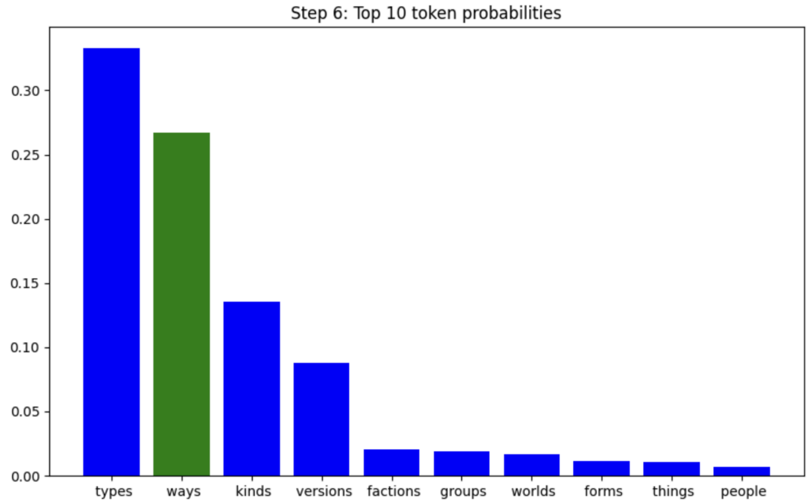


```
[(nanogpt_env) cherylcook@Cheryls-MacBook-Pro nanoGPT % python sample.py --]
show_probs=True  --init_from=gpt2 --start="Once upon a time" --num_samples
=1  --max_new_tokens=10 --temperature=0.5 --device=mps
Overriding: show_probs = True
Overriding: init_from = gpt2
Overriding: start = Once upon a time
Overriding: num_samples = 1
Overriding: max_new_tokens = 10
Overriding: temperature = 0.5
Overriding: device = mps
To use data.metrics please install scikit-learn. See https://scikit-learn.
org/stable/index.html
loading weights from pretrained gpt: gpt2
forcing vocab_size=50257, block_size=1024, bias=True
overriding dropout rate to 0.0
number of parameters: 123.65M
No meta.pkl found, assuming GPT-2 encodings...
Once upon a time, there were two different ways to get to the
```

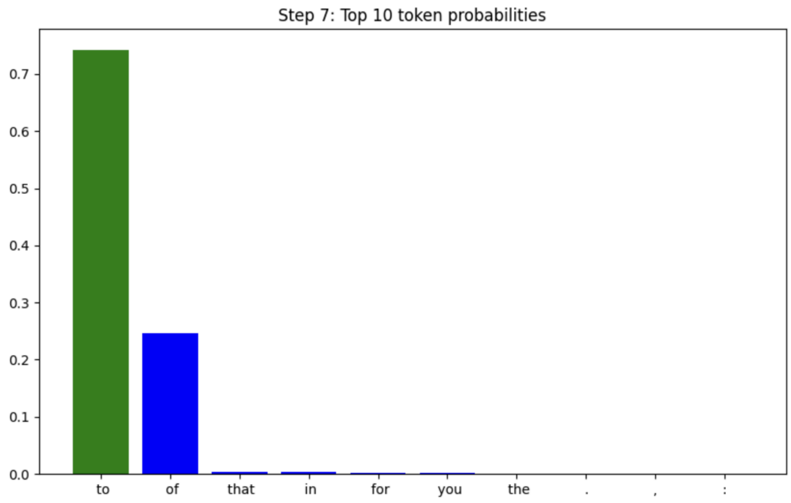


After step 2, all of the selected tokens were one of the top two highest probability options.

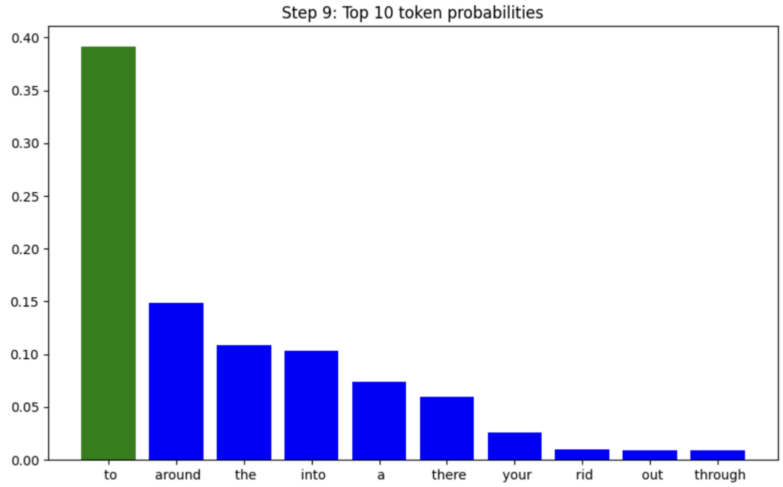




From Step 7-10, the probability distributions sharpened even more and the highest probability token was selected every time.



Step 7: Top 10 token probabilities



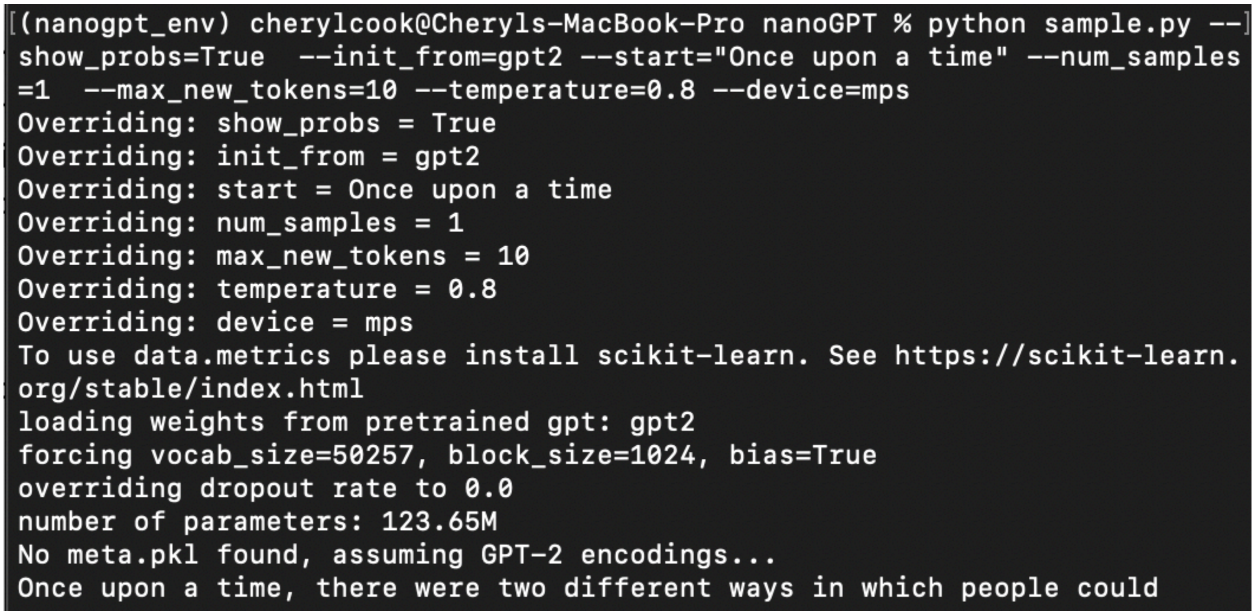
Step 9: Top 10 token probabilities

A lower temperature means a sharper probability distribution, resulting in the selected token almost always being one with a high-probability.

Based on these results, a low temperature of 0.5 results in sharp probability distributions, with the top token having a notably higher probability than the rest. The result is quite conservative and deterministic token selection that favors high-probability tokens. Of the 10 tokens generated in this example, there was only one step (step 2) where the selected token wasn’t one of the top two most probable.

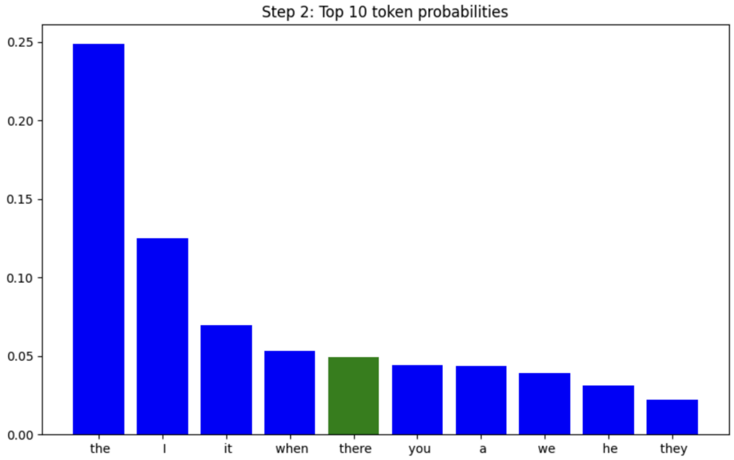
Temperature = 0.8:

Once upon a time, there were two different ways in which people could

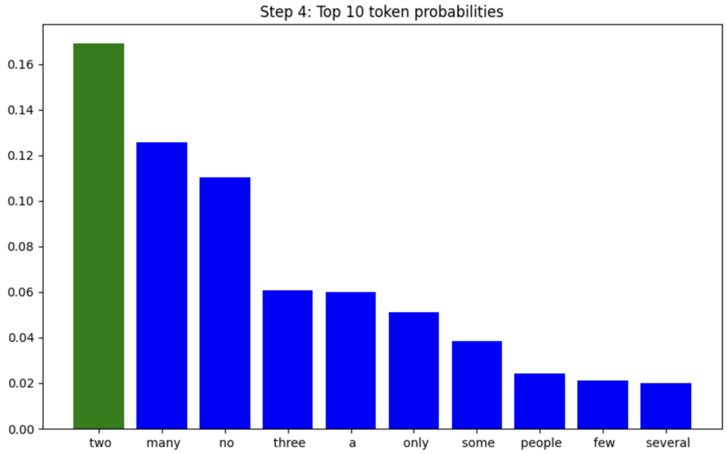


```
[(nanogpt_env) cherylcook@Cheryls-MacBook-Pro nanoGPT % python sample.py --]
show_probs=True  --init_from=gpt2 --start="Once upon a time" --num_samples
=1  --max_new_tokens=10 --temperature=0.8 --device=mps
Overriding: show_probs = True
Overriding: init_from = gpt2
Overriding: start = Once upon a time
Overriding: num_samples = 1
Overriding: max_new_tokens = 10
Overriding: temperature = 0.8
Overriding: device = mps
To use data.metrics please install scikit-learn. See https://scikit-learn.
org/stable/index.html
loading weights from pretrained gpt: gpt2
forcing vocab_size=50257, block_size=1024, bias=True
overriding dropout rate to 0.0
number of parameters: 123.65M
No meta.pkl found, assuming GPT-2 encodings...
Once upon a time, there were two different ways in which people could
```

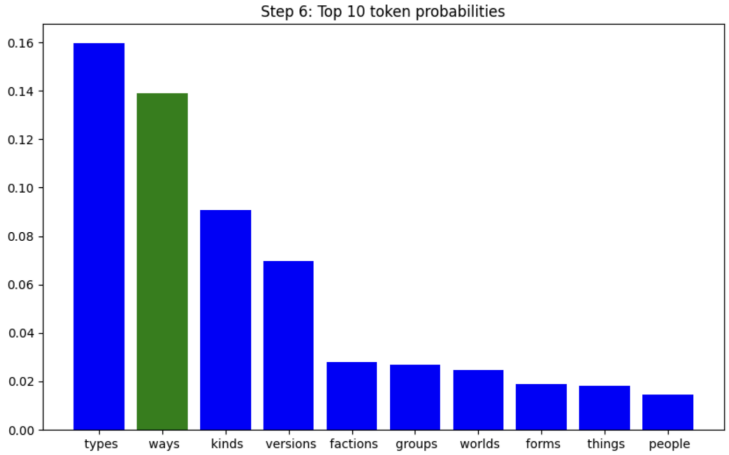
The token probabilities are more distributed in Step 2 compared to temperature 0.5's step 2 (where 'the' had 50% probability).



From Step 3-6, one of the top two highest probability tokens was always selected.



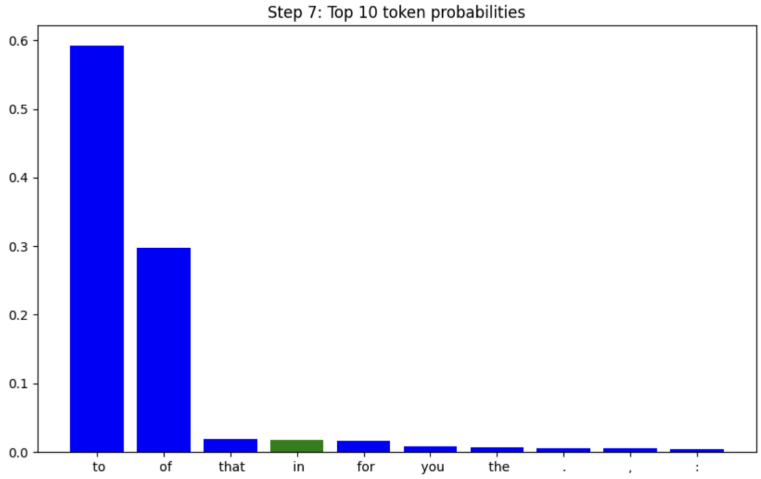
Step 4: Top 10 token probabilities



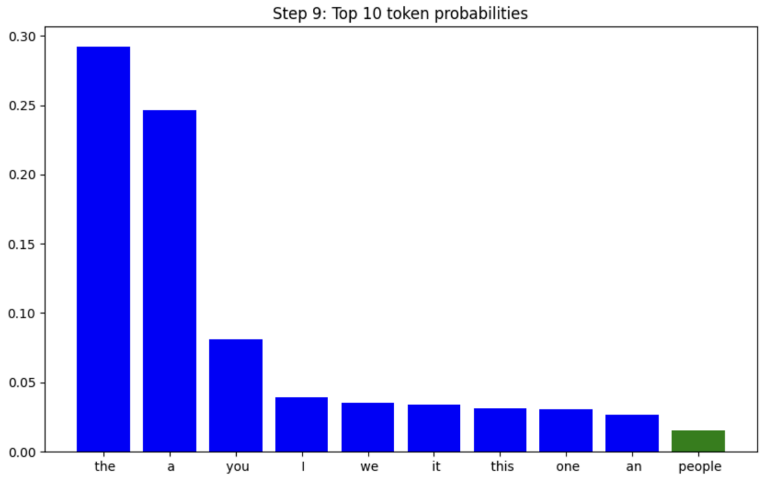
Step 6: Top 10 token probabilities

These steps had the same top ten tokens as temperature, but the probability distribution is more evenly spread (less sharp) compared to temperature 0.5's distributions. For example, Step 4's probability of 'two' probability was around 0.16, compared to temperature 0.5's 'two' being around 0.3. The same applies to 'types' in Step 6.

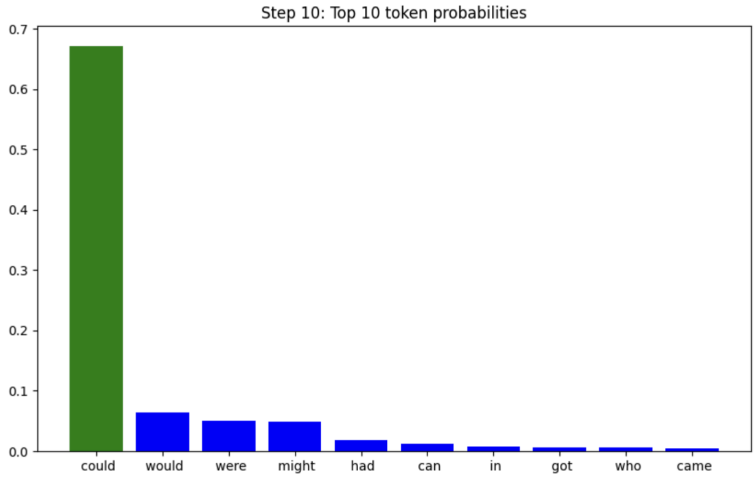
From Step 7-10, there was diverse token selection compared to temperature 0.5, as a mix of the highest-probability tokens and low probability tokens were selected.



Step 7: Top 10 token probabilities



Step 9: Top 10 token probabilities

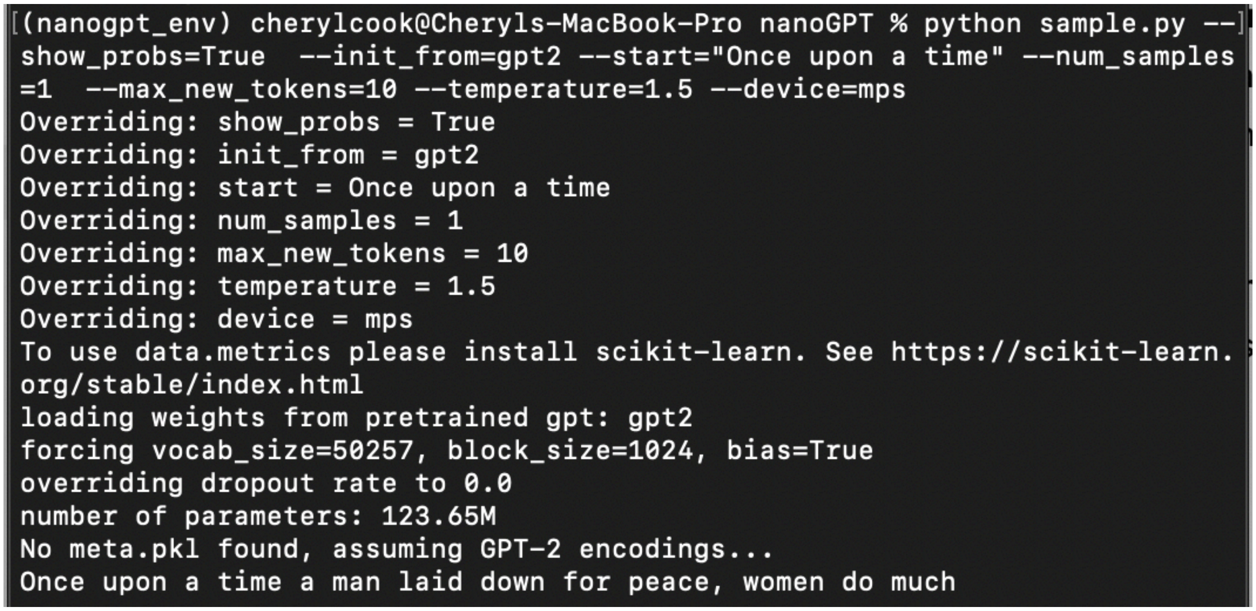


Step 10: Top 10 token probabilities

Based on these results, a moderate temperature of 0.8 still shows preference for high-probability tokens, but exhibits more diversity in token selection than temperature 0.5.

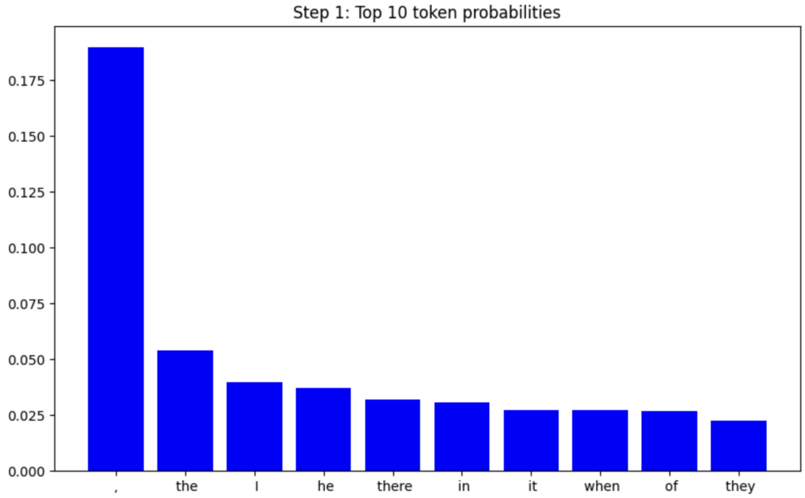
Temperature = 1.5:

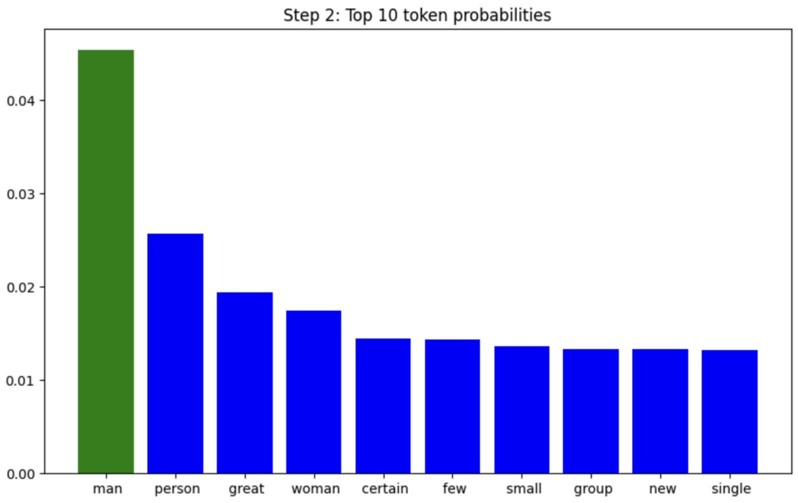
Once upon a time a man laid down for peace, women do much



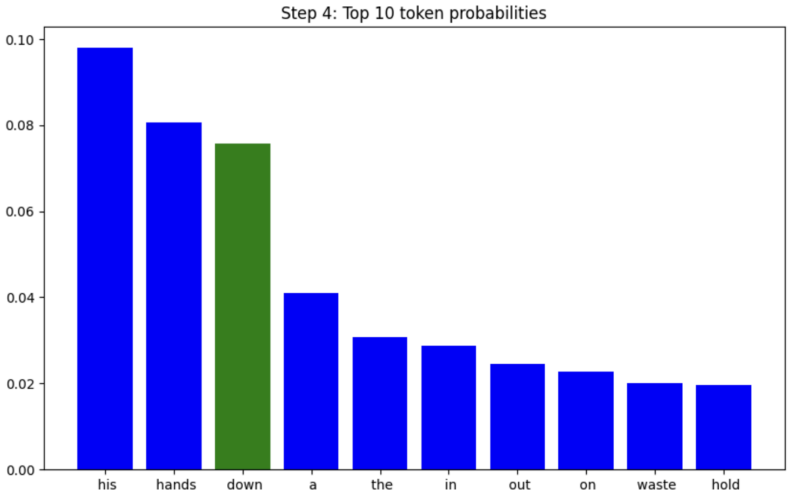
```
[(nanogpt_env) cherylcook@Cheryls-MacBook-Pro nanoGPT % python sample.py --]
show_probs=True  --init_from=gpt2 --start="Once upon a time" --num_samples
=1  --max_new_tokens=10 --temperature=1.5 --device=mps
Overriding: show_probs = True
Overriding: init_from = gpt2
Overriding: start = Once upon a time
Overriding: num_samples = 1
Overriding: max_new_tokens = 10
Overriding: temperature = 1.5
Overriding: device = mps
To use data.metrics please install scikit-learn. See https://scikit-learn.
org/stable/index.html
loading weights from pretrained gpt: gpt2
forcing vocab_size=50257, block_size=1024, bias=True
overriding dropout rate to 0.0
number of parameters: 123.65M
No meta.pkl found, assuming GPT-2 encodings...
Once upon a time a man laid down for peace, women do much
```

While the top one or two tokens still have noticeably higher probabilities than the rest, the distribution is much flatter compared to temperatures 0.5 and 0.8. This gives lower-probability tokens a higher chance of being chosen, since the difference in probability isn't as drastic. This is evidenced by certain steps of this example, where several of the selected tokens were not even in the top ten most probable. This makes the resulting output more random/diverse, but it also makes less grammatical sense.

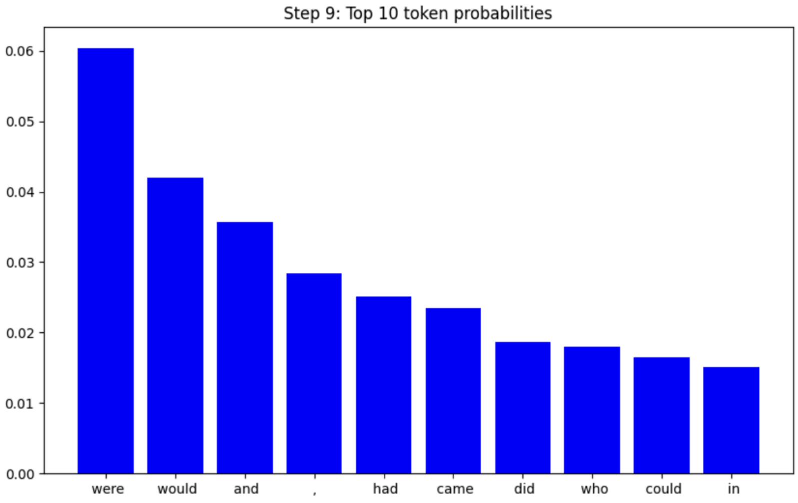




Step 2: Top 10 token probabilities



Step 4: Top 10 token probabilities



Step 9: Top 10 token probabilities