A simple code I wrote to see how Transformers work and how the softmax gives us the probability distribution for the next word is as follows - (Full Disclaimer - I have used AI to generate the following code but all inferences derived from it are from personal experience)

```
import torch
from transformers import GPT2LMHeadModel, GPT2Tokenizer, top_k_top_p_filtering
import torch.nn.functional as F

# Load the model and tokenizer from the local directory
model_name = "./gpt2_local"  # Path to the locally stored GPT-2 model
tokenizer = GPT2Tokenizer.from_pretrained(model_name)
model = GPT2LMHeadModel.from_pretrained(model_name)

# Test the model with some input
input_text = "The quick brown fox"
input_ids = tokenizer.encode(input_text, return_tensors="pt")

# Get the model's output logits
with torch.no_grad():
    outputs = model(input_ids)
    logits = outputs.logits

# Get the logits for the last token in the input
last_token_logits = logits[:, -1, :]
top_k = 5
top_p = 0.95
#####
temperature = 1
last_token_logits /= temperature
filtered_logits = top_k_top_p_filtering(last_token_logits, top_k=top_k, top_p=top_p)
# Calculate probabilities using softmax
probs = F.softmax(filtered_logits, dim=-1)

# Get the top 10 predictions
top_k = 10
top_k_probs, top_k_indices = torch.topk(probs, top_k)

# Print the top-k predicted words with their probabilities
print(f"Input: '{input_text}'")
print(f"Top {top_k} predictions for the next word:")

for i, (index, prob) in enumerate(zip(top_k_indices[0], top_k_probs[0])):
    predicted_word = tokenizer.decode([index])
    print(f"{i+1}: {predicted_word} (Probability: {prob.item():.4f})")
```

Here we can tweak the top_p, top_k and temperature parameters and see how it affects the probability distribution of the next 10 possible words the model might output.

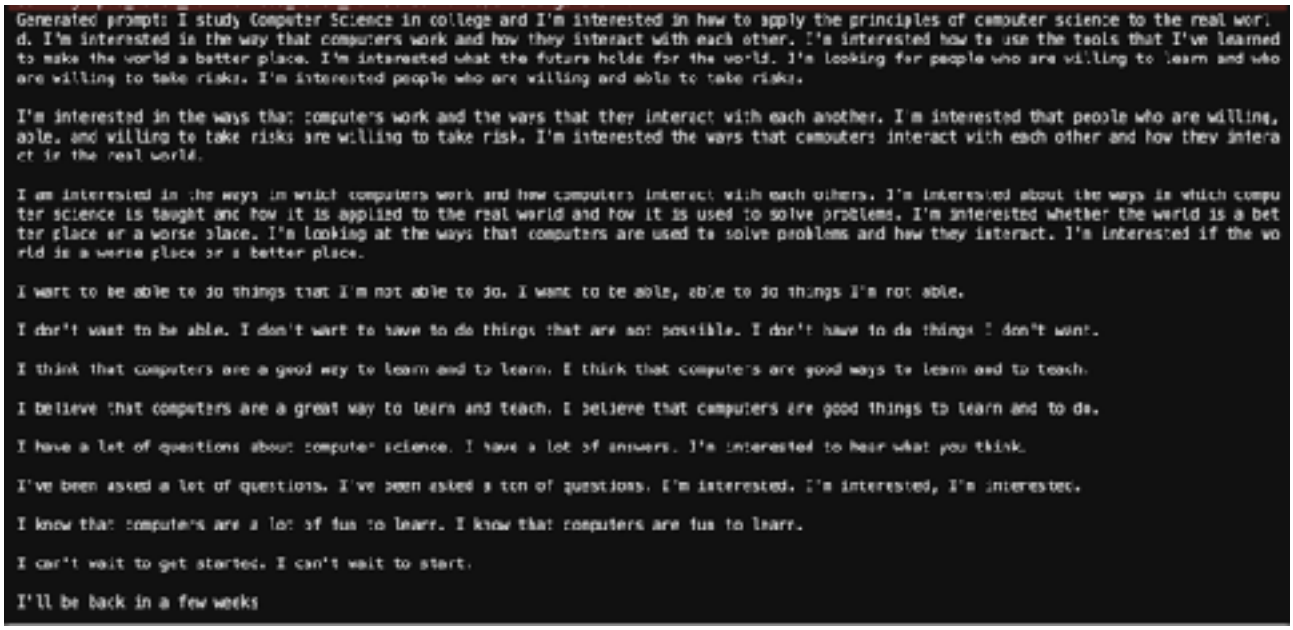
```
Input: 'The quick brown fox'
Temperature: '1'
Top_p: '0.95'
Top_k: '10'
Top-10 predictions for the next word:
1: es (Probability: 0.4725)
2:  was (Probability: 0.1666)
3:  is (Probability: 0.1255)
4: 's (Probability: 0.1206)
5: , (Probability: 0.1141)
6: ' (Probability: 0.0000)
7: # (Probability: 0.0000)
8: & (Probability: 0.0000)
9: % (Probability: 0.0000)
10: * (Probability: 0.0000)
```

```
Input: 'The quick brown fox'
Temperature: '1'
Top_p: '0.95'
Top_k: '10'
Top-10 predictions for the next word:
1: es (Probability: 0.3777)
2:  was (Probability: 0.1331)
3:  is (Probability: 0.1002)
4: 's (Probability: 0.0965)
5: , (Probability: 0.0911)
6:  that (Probability: 0.0647)
7: - (Probability: 0.0510)
8:  had (Probability: 0.0459)
9:  has (Probability: 0.0307)
10: ' (Probability: 0.0000)
```

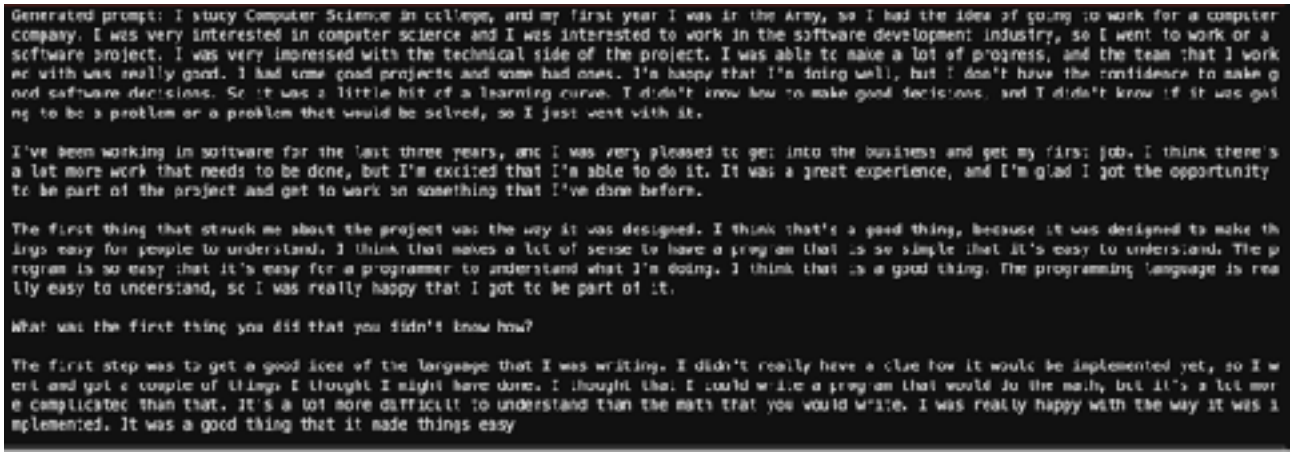
```
Input: 'The quick brown fox'
Temperature: '5'
Top_p: '0.95'
Top_k: '10'
Top-10 predictions for the next word:
1: es (Probability: 0.1354)
2:  was (Probability: 0.1099)
3:  is (Probability: 0.1038)
4: 's (Probability: 0.1036)
5: , (Probability: 0.1019)
6:  that (Probability: 0.0951)
7: - (Probability: 0.0507)
8:  had (Probability: 0.0483)
9:  has (Probability: 0.0362)
10: y (Probability: 0.0052)
```

```
Input: 'The quick brown fox'
Temperature: '5'
Top_p: '0.5'
Top_k: '10'
Top-10 predictions for the next word:
1: es (Probability: 0.2444)
2:  was (Probability: 0.1083)
3:  is (Probability: 0.1074)
4: 's (Probability: 0.1060)
5: , (Probability: 0.1039)
6: ' (Probability: 0.0000)
7: # (Probability: 0.0000)
8: & (Probability: 0.0000)
9: % (Probability: 0.0000)
10: * (Probability: 0.0000)
```

Running samples like this gave me a more intuitive sense of what each of these parameters mean.



This was one of my results with maxlength = 500, topk = 1, topp = 0.95, temperature = 1 and my prompt being “I study Computer Science in college”



All parameters remaining the same, this was the response I got with a topk = 5. We notice that it is not as repititive as the previous case but still does not make all that much sense and is in general giving very deterministic responses.

With a topk of 5 and a temperature of 1.5 we get something slightly more creative but still not all that very sensible.

I did get a questionable response by GPT2 when I only gave a spacebar as a prompt and set top_k = 40, top_p = 0.5, temperature = 1.9. Perhaps, it got a little too creative.

I have not documented everything I have tried but a few interesting attempts I made have been documented in this doc.