Results and observations on english and hindi versions of TinyStories dataset

Overview:

The research paper "**TinyStories: How Small Can Language Models Be and Still Speak Coherent English?"** provided us with a means that can help us design a dataset that:

- Can preserve the core element of natural language, such as grammar, language tasks and reasoning
- Is smaller and more refined in breadth and diversity.

The TinyStories dataset contains 2.12 million rows in English language, the task to translate into Indian regional languages like Hindi, Marathi etc required large computation. I translated more than 100,000 rows of the English TinyStories dataset into Hindi and checked for coherency during inference. By optimising the training parameters of the model during training, loss function was reduced and output during inference gave better coherent result.

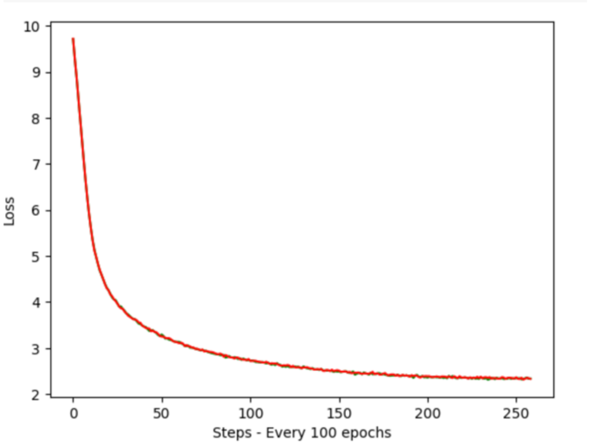
Similarly in case of the Hindi translated version of the dataset, the changes made included using 'sentencepiece' tokenizer instead of the 'tiktoken' tokeniser due to its ability to handle non-latin and non-english languages better and have better control over the vocabulary using unigram encoding instead of the byte-pair encoding of tiktoken. The output in case of the Hindi dataset was less coherent due to the smaller dataset translated and less number of iterations conducted to minimize the train and val loss error. With larger dataset translated and more number of iterations implemented , the coherency of the Hindi version too would improve providing better results.

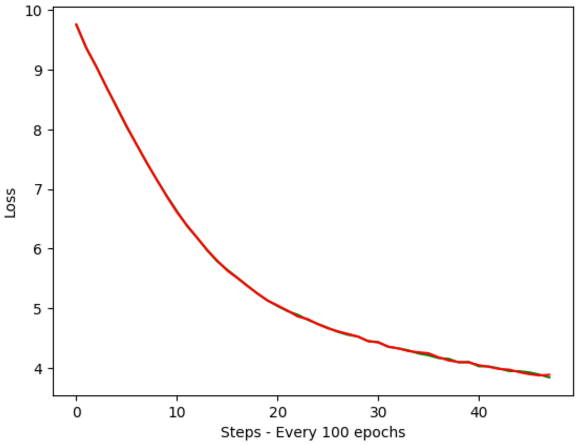
Working:

The optimizations involved in decreasing the loss function and providing stability and avoiding erratic uneven movement patterns were:

- Increasing n_layers to14 from 12

- Setting dropout = 0.15 from 0.2
- Increasing iterations to 26000 from 25000. (more iterations required to further improve performance but due to limitations could increase only by 1000.)
- Setting minimum lr = 1e-5 from 5e-4
- Increasing block size to 128 from 64 to gain larger context of the data for the model to see.
- Decrease gradient accumulation steps to 20 from 50. Larger gradient value causes instability and too many gradients
- Increase warmup steps to 2000 from 100, providing better generalization for the model and better training.

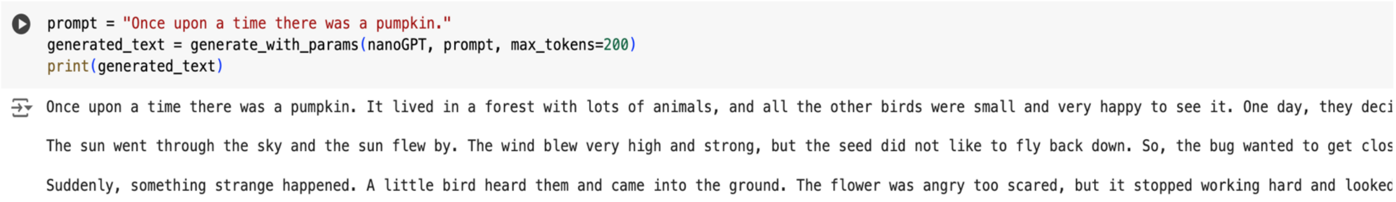




The left graph depicts the loss function of the English dataset model, and the right graph is of the Hindi dataset model. Both the graphs depict improved stability, avoiding uneven movement patterns and the English dataset model depicts better minima convergence now. In case of the Hindi dataset model, more iterations required to further improve performance and yield better minima convergence.

OUTPUT:

1. English dataset



```
prompt = "Once upon a time there was a pumpkin."
generated_text = generate_with_params(nanoGPT, prompt, max_tokens=200)
print(generated_text)
```

```
Once upon a time there was a pumpkin. It lived in a forest with lots of animals, and all the other birds were small and very happy to see it. One day, they deci

The sun went through the sky and the sun flew by. The wind blew very high and strong, but the seed did not like to fly back down. So, the bug wanted to get clos

Suddenly, something strange happened. A little bird heard them and came into the ground. The flower was angry too scared, but it stopped working hard and lookec
```

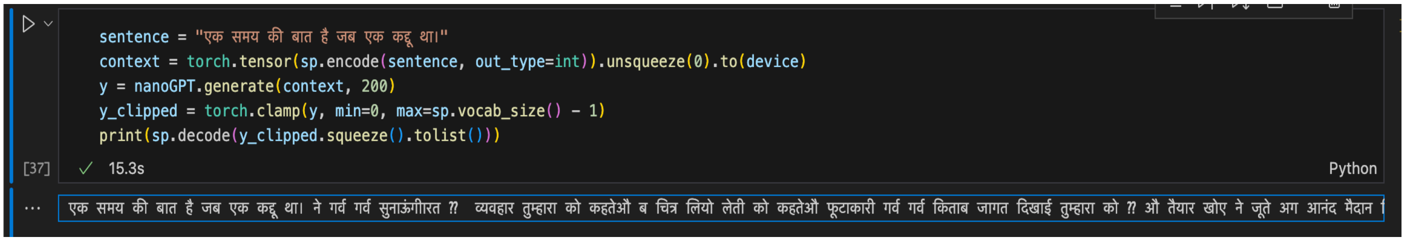
The output generated had decent and better grammar as compared to the grammar previously. Optimizing top p, top k, and temperature parameters was incorporated to improve performance.

Second example: prompt was: “ the sun is bright today, ”

Output: max tokens=200

the sun is bright today, like a lot of things. "Let's play outside and find some water," Ben says. "We can have fun!" They look for the sun and see a big lake with leaves in it. They think it will be fun to have fun. But then they hear a loud noise. It is coming from the shore. The water is dark and wet. The sun is shining and the sky looks hot. "What are you doing?" Anna asks. "I'm looking for something else to eat." Ben says, "I'm hungry. Can we have some juice?" Anna smiles. She says, "Of course, I want to go back inside." They get up. But the water is too fast and slippery. The pond is too big and cold and wet. They are safe. Then they cannot hurt their food. Mom says, but she needs to dry and dad's bathtub. She says, Lily can

2. Hindi dataset:



```
sentence = "एक समय की बात है जब एक कद्दू था।"
context = torch.tensor(sp.encode(sentence, out_type=int)).unsqueeze(0).to(device)
y = nanoGPT.generate(context, 200)
y_clipped = torch.clamp(y, min=0, max=sp.vocab_size() - 1)
print(sp.decode(y_clipped.squeeze().tolist()))
```

```
एक समय की बात है जब एक कद्दू था। ने गर्व गर्व सुनाऊंगीरत ⁇  व्यवहार तुम्हारा को कहतेऔ ब चित्र लियो लेती को कहतेऔ फूटाकारी गर्व गर्व किताब जागत दिखाई तुम्हारा को ⁇ औ तैयार खोए ने जूते अग आनंद मैदान 
```

As can be seen, the output isn’t very coherent and lacks proper grammar and punctuations, despite optimizing the model with top p, top k, temperature parameters like in case of the English dataset model.

Full output:

एक समय की बात है जब एक कद्दू था। ने गर्व गर्व सुनाऊंगीरत ?? व्यवहार तुम्हारा को कहतेऔ ब चित्र लियो लेती को कहतेऔ फूटाकारी गर्व गर्व किताब जागत दिखाई तुम्हारा को ?? औ तैयार खोए ने जूते अग आनंद मैदान दिल छड़ी होता दिखाई पढ़ा ने जीव चले करो मिले देखना एवोकाडोऔ मज़ेदारस्टोर नरमऔ दिखाईऔ लूसी को दिल बनाना ने करवाया फलों याद हँसते को पाकर तैयार याद हिस्साचांदऔऔ दूसरे हँसते मज़ेदार मिलने बनाता सहेली चिप्स खिलाता तस्वीरें लूसी उधार छीलन उम्मीद डरा कूदना बैगपेड़ समुद्र पीने मज़ेदारदूसरेऔ याद खजानाँ बांबी चमकाए ने जगह गर्व गर्व किताबऔऔ तालाब जेक अभी लंबी नाई डरा भागते बदलती उठाया थूका शेर याद अभिभावक लूसीदूसरे घटा सड़ेऔ लूसीदूसरेऔ ने दूसर तैयार्स गिर याद हिस्सा यूनि दिखाई पत्नी मज़ेदार ब्लूई बुझात दिखाई बाघ बनाना तैयारॉंग होताऔमाल सके दुखेंगे ने लियो सीढ़ी पिछवाड़औ ने लियो तैयार तैरऔ स्ट्रॉबेरी इंतज साझाऔमाल याद हिस्सा शिल्प ने गर्व गर्व ब्रोकली उम्मीद शेर याद खजाना दिखाईबापथर्मामीटर मौज याद भरवा इंद्रधनुष नाची तैयार बोतलमाल याद खजाना कूदतीमाल याद नीले ने व्यवहार बकजीब्बीमकनमस्ते