What is an LLM? How does it relate/differ from n-gram models? [2 points]

A large language model is a DNN language model which is trained on humongous amounts of data, has a high number of trainable parameters (into billions and trillions) and can perform complex tasks like efficient text generation etc. In case of NLP use case, they work on the idea of parallelization and contextual embeddings (via self-attention). They differ from traditional n-gram models in terms of the ability to capture context.

While n-gram models are frequency based probabilistic models, that is, they do a frequency-based probability calculation (conditional probability of token sequences), LLMs use self-attention mechanism which calculates attention scores (scaled dot products between Q,K vectors and find contextual vectors by multiplying attention scores with the V vectors).

Ans 2

Experiment 1: Baseline (No Regex Cleaning, 2M characters)

Parameters: Batch Size = 4, Block Size = 8, Learning Rate = 1e-2
Text Size: ~2 million characters (uncleaned)
Loss (Final @ Iter 9000): **2.4186**

Observation: Starting with unfiltered text resulted in slower convergence and higher initial noise due to the presence of special characters and irregular formatting.

```
Interation 0: loss 5.2288
Interation 1000: loss 4.5936
Interation 2000: loss 4.1267
Interation 3000: loss 3.6019
Interation 4000: loss 2.8823
Interation 5000: loss 2.7306
Interation 6000: loss 2.8916
Interation 7000: loss 3.0858
Interation 8000: loss 2.7470
Interation 9000: loss 2.4186
```

Experiment 2: Regex Cleaning Applied, 2M characters

Parameters: Batch Size = 4, Block Size = 8, Learning Rate = 1e-2
Text Size: ~2 million characters (cleaned using regex)
Loss (Final @ Iter 9000): **2.2667**

Observation: Removing special characters and normalizing whitespace helped the model train more efficiently, showing improved convergence and lower final loss compared to the uncleaned version.

```
Interation 0: loss 4.1492
Interation 1000: loss 3.3847
Interation 2000: loss 2.7102
Interation 3000: loss 2.8838
Interation 4000: loss 2.7011
Interation 5000: loss 2.2078
Interation 6000: loss 2.6523
Interation 7000: loss 2.4475
Interation 8000: loss 2.1486
Interation 9000: loss 2.2667
```

Experiment 3: Larger Block Size (12), Regex, 2M characters

Parameters: Batch Size = 4, Block Size = 12, Learning Rate = 1e-2
Text Size: ~2 million characters (cleaned using regex)
Loss (Final @ Iter 9000): **2.2651**

Observation: Increasing the block size to 12 did not drastically impact final loss, but may improve the model's ability to capture longer patterns, which could enhance the quality of generated text.

```
Interation 0: loss 3.8315
Interation 1000: loss 3.3981
Interation 2000: loss 2.9921
Interation 3000: loss 2.6285
Interation 4000: loss 2.4304
Interation 5000: loss 2.6598
Interation 6000: loss 2.4799
Interation 7000: loss 2.2251
Interation 8000: loss 2.1137
Interation 9000: loss 2.2651
```

Experiment 4: Larger Corpus (9M characters), Block 12, Regex

Parameters: Batch Size = 4, Block Size = 12, Learning Rate = 1e-2
Text Size: ~9 million characters (cleaned using regex)
Loss (Final @ Iter 19000): **1.9582**

Observation:
Scaling up the training data size significantly improved model performance. The lower loss suggests stronger learning of language patterns and better generalization, though training took more time.

```
Interation 0: loss 4.0232
Interation 1000: loss 3.7116
Interation 2000: loss 2.8937
Interation 3000: loss 2.8213
Interation 4000: loss 2.4352
Interation 5000: loss 2.4499
Interation 6000: loss 2.4872
Interation 7000: loss 2.3182
Interation 8000: loss 2.5858
Interation 9000: loss 2.2085
Interation 10000: loss 2.1476
Interation 11000: loss 2.2687
Interation 12000: loss 2.2353
Interation 13000: loss 2.3546
Interation 14000: loss 2.2765
Interation 15000: loss 2.1992
Interation 16000: loss 2.1957
Interation 17000: loss 2.1981
Interation 18000: loss 2.3027
Interation 19000: loss 1.9582
```