Assignment 2 Report

1. Datasets Details

This experiment used two open-source English text corpora: Wikipedia (Wikimedia, 2023-11-01) and FineWeb-Edu (HuggingFaceFW).

A total of 256 samples were taken from each source (512 documents). All texts were normalized and cleaned by lowercasing, removing HTML tags, collapsing spaces, and compressing repeated characters.

After filtering short (<50 words) and duplicate entries, about 480–500 unique documents remained. Texts were tokenized with the GPT-2 tokenizer, using the EOS token for padding, and split into 128-token blocks for uniform input length.

This produced roughly 20k–25k tokenized chunks, divided into 90% training and 10% validation sets for next-token prediction.

2. Model architecture and parameters

The implemented model, MiniGPT, is a lightweight transformer-based language model built using PyTorch.

It follows the standard GPT architecture with token embeddings, positional embeddings, stacked Transformer encoder layers, layer normalization, and a final linear projection head for next-token prediction.

Key components:

Embedding layer: Maps input token IDs and positional indices into 128-dimensional vectors.

Transformer encoder layers (×2): Each uses 4 attention heads, hidden dimension = 4 × 128 = 512, and GELU activation.

Layer normalization: Applied before the output head to stabilize training.

Output head: A linear layer projecting the hidden states back to the vocabulary size for logits prediction.

Parameters:

Embedding dimension (d_model)	128
Number of layers (n_layers)	2

Number of attention heads (n_heads)	4
Feed-forward dimension	512 (= 4 × d_model)
Sequence length (block_size)	128
Vocabulary size	GPT-2 tokenizer vocabulary (~50k)

3. Training setup and hyperparameter experiments

The MiniGPT model was trained for 5 epochs on a GPU (CUDA) using the AdamW optimizer with a learning rate of 5×10^{-4}. Training used a batch size of 32 and a sequence length of 128 tokens, with cross-entropy loss applied for next-token prediction.

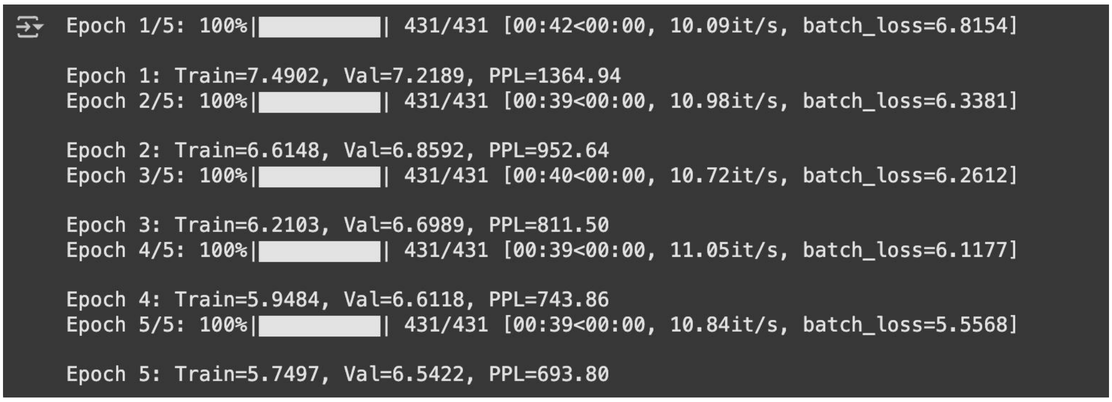
A custom dataset and collate function handled token padding and batching.

After each epoch, the model was evaluated on a validation set, and both training loss and perplexity (PPL) were recorded. Model checkpoints were saved after every epoch for reproducibility.

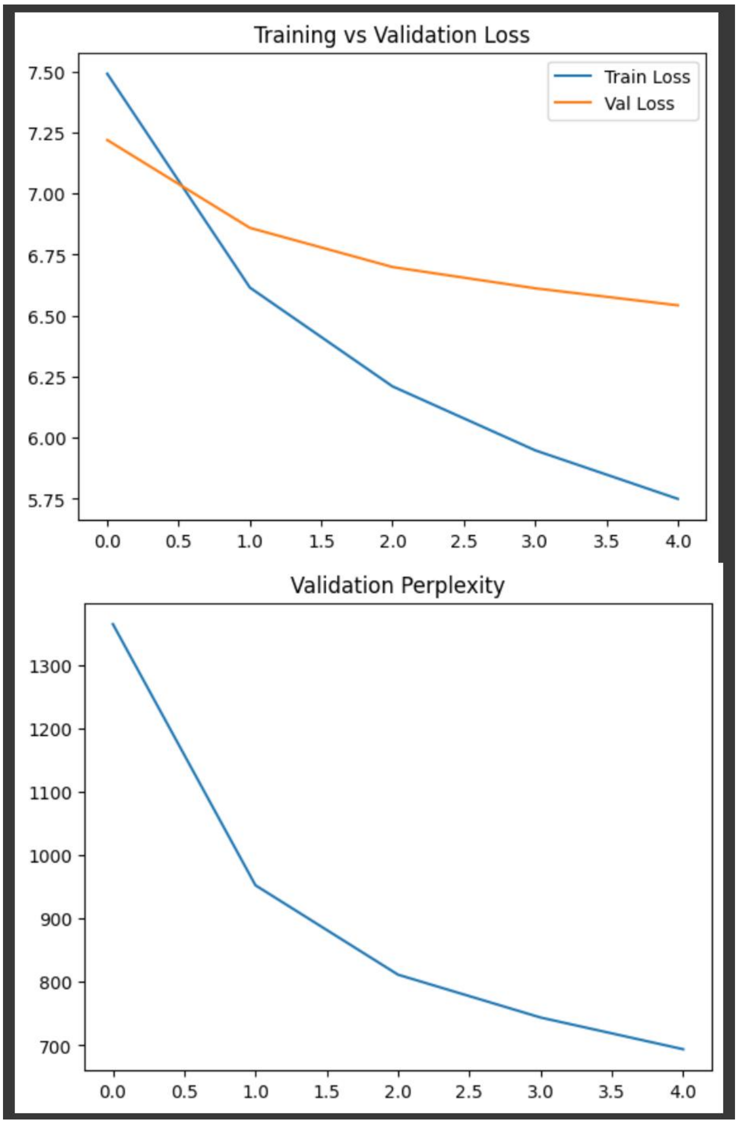
hyperparameter experiments:

Number	Learning Rate	Batch Size	Embedding Dim	Layers
Exp-1 (Baseline)	5e-4	32	128	2
Exp-2 (Smaller model)	5e-4	32	64	1
Exp-3 (Larger model)	5e-4	32	256	2
Exp-4 (Learning rate test)	1e-3	32	128	2
Exp-5 (Batch size variation)	5e-4	64	128	2

Exp-1 (Baseline):



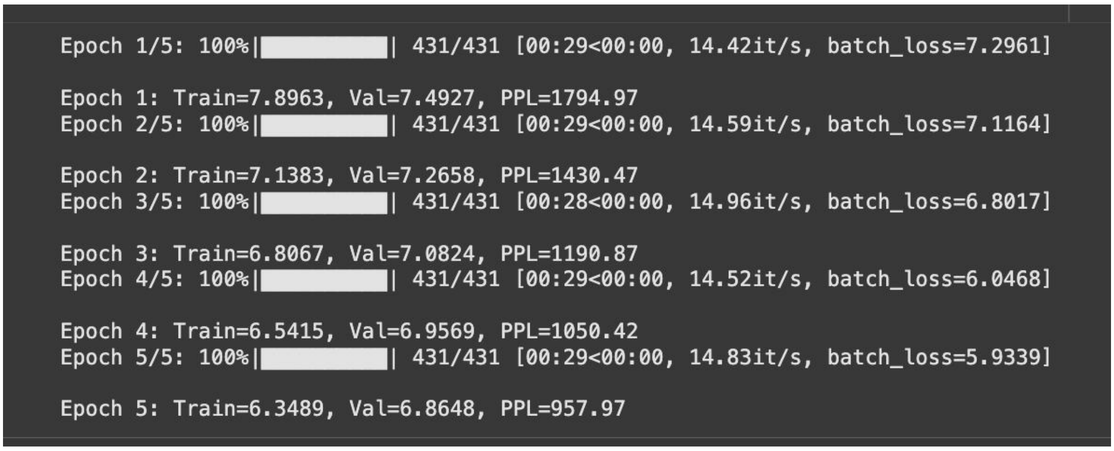
```
Epoch 1/5: 100%|██████████| 431/431 [00:42<00:00, 10.09it/s, batch_loss=6.8154]

Epoch 1: Train=7.4902, Val=7.2189, PPL=1364.94
Epoch 2/5: 100%|██████████| 431/431 [00:39<00:00, 10.98it/s, batch_loss=6.3381]

Epoch 2: Train=6.6148, Val=6.8592, PPL=952.64
Epoch 3/5: 100%|██████████| 431/431 [00:40<00:00, 10.72it/s, batch_loss=6.2612]

Epoch 3: Train=6.2103, Val=6.6989, PPL=811.50
Epoch 4/5: 100%|██████████| 431/431 [00:39<00:00, 11.05it/s, batch_loss=6.1177]

Epoch 4: Train=5.9484, Val=6.6118, PPL=743.86
Epoch 5/5: 100%|██████████| 431/431 [00:39<00:00, 10.84it/s, batch_loss=5.5568]

Epoch 5: Train=5.7497, Val=6.5422, PPL=693.80
```



Training vs Validation Loss

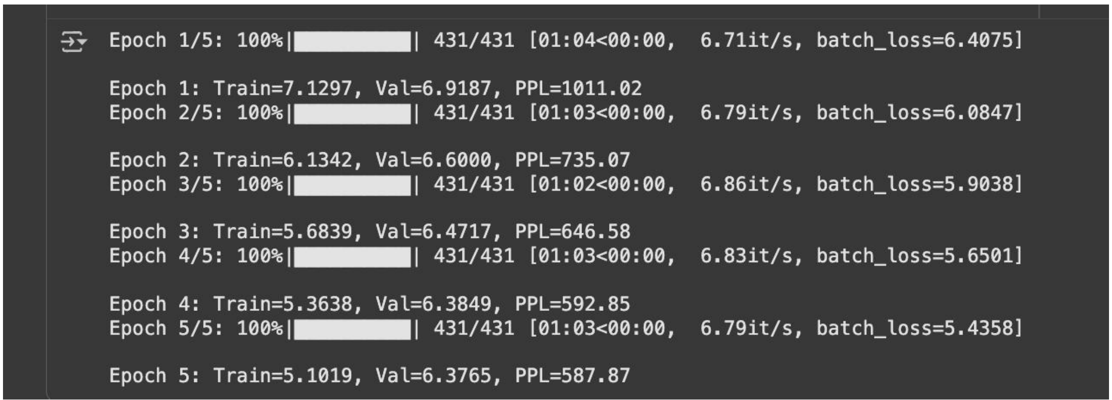
Validation Perplexity

Exp-2 (Smaller model):

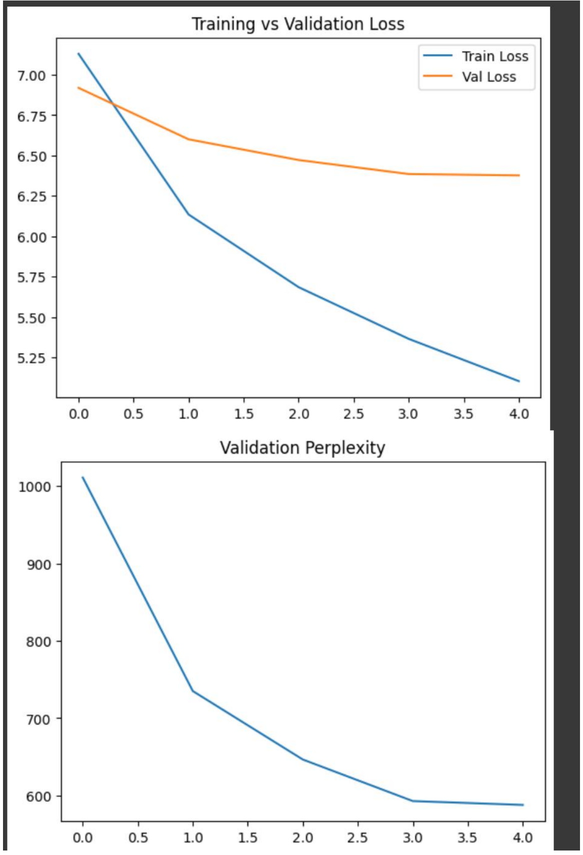


```
Epoch 1/5: 100%|██████████| 431/431 [00:29<00:00, 14.42it/s, batch_loss=7.2961]

Epoch 1: Train=7.8963, Val=7.4927, PPL=1794.97
Epoch 2/5: 100%|██████████| 431/431 [00:29<00:00, 14.59it/s, batch_loss=7.1164]

Epoch 2: Train=7.1383, Val=7.2658, PPL=1430.47
Epoch 3/5: 100%|██████████| 431/431 [00:28<00:00, 14.96it/s, batch_loss=6.8017]

Epoch 3: Train=6.8067, Val=7.0824, PPL=1190.87
Epoch 4/5: 100%|██████████| 431/431 [00:29<00:00, 14.52it/s, batch_loss=6.0468]

Epoch 4: Train=6.5415, Val=6.9569, PPL=1050.42
Epoch 5/5: 100%|██████████| 431/431 [00:29<00:00, 14.83it/s, batch_loss=5.9339]

Epoch 5: Train=6.3489, Val=6.8648, PPL=957.97
```

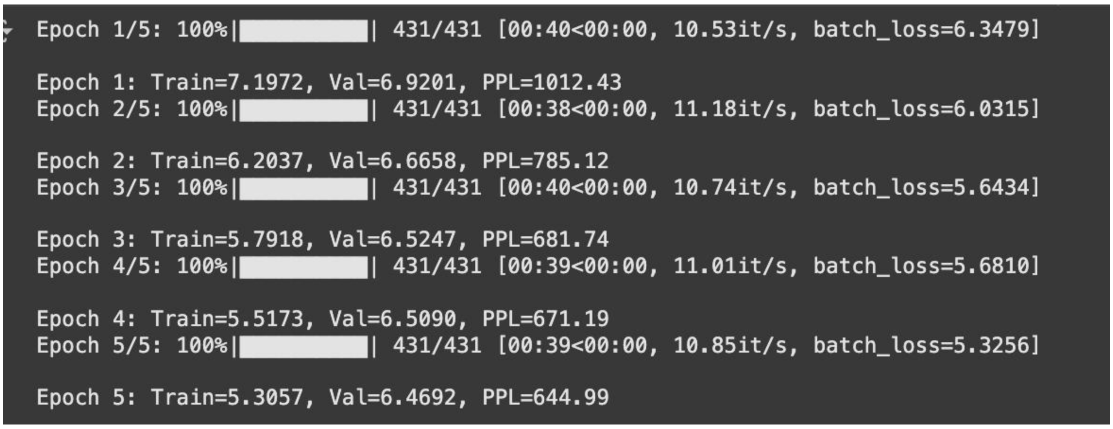
Exp-3 (Larger model):



```
Epoch 1/5: 100%|██████████| 431/431 [01:04<00:00,  6.71it/s, batch_loss=6.4075]

Epoch 1: Train=7.1297, Val=6.9187, PPL=1011.02
Epoch 2/5: 100%|██████████| 431/431 [01:03<00:00,  6.79it/s, batch_loss=6.0847]

Epoch 2: Train=6.1342, Val=6.6000, PPL=735.07
Epoch 3/5: 100%|██████████| 431/431 [01:02<00:00,  6.86it/s, batch_loss=5.9038]

Epoch 3: Train=5.6839, Val=6.4717, PPL=646.58
Epoch 4/5: 100%|██████████| 431/431 [01:03<00:00,  6.83it/s, batch_loss=5.6501]

Epoch 4: Train=5.3638, Val=6.3849, PPL=592.85
Epoch 5/5: 100%|██████████| 431/431 [01:03<00:00,  6.79it/s, batch_loss=5.4358]

Epoch 5: Train=5.1019, Val=6.3765, PPL=587.87
```

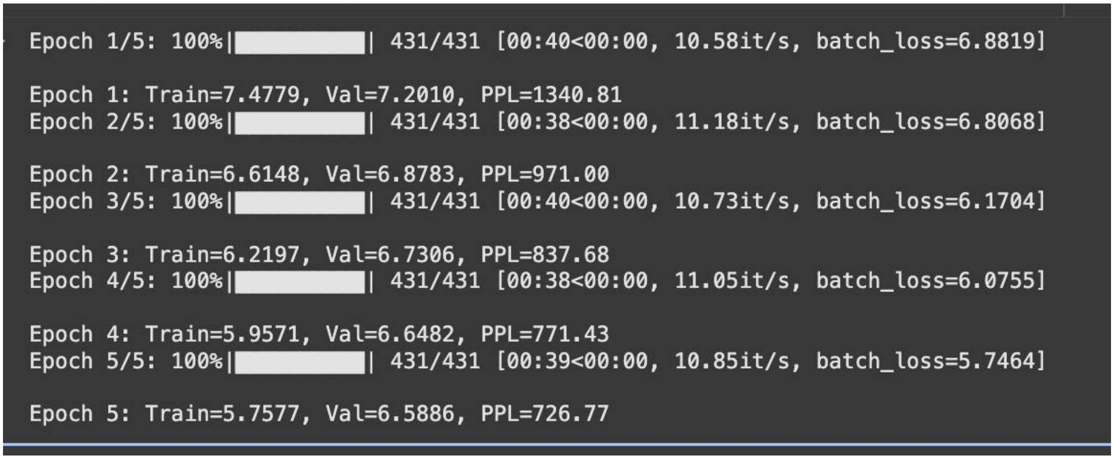


Exp-4 (Learning rate test):



```
Epoch 1/5: 100%|██████████| 431/431 [00:40<00:00, 10.53it/s, batch_loss=6.3479]

Epoch 1: Train=7.1972, Val=6.9201, PPL=1012.43
Epoch 2/5: 100%|██████████| 431/431 [00:38<00:00, 11.18it/s, batch_loss=6.0315]

Epoch 2: Train=6.2037, Val=6.6658, PPL=785.12
Epoch 3/5: 100%|██████████| 431/431 [00:40<00:00, 10.74it/s, batch_loss=5.6434]

Epoch 3: Train=5.7918, Val=6.5247, PPL=681.74
Epoch 4/5: 100%|██████████| 431/431 [00:39<00:00, 11.01it/s, batch_loss=5.6810]

Epoch 4: Train=5.5173, Val=6.5090, PPL=671.19
Epoch 5/5: 100%|██████████| 431/431 [00:39<00:00, 10.85it/s, batch_loss=5.3256]

Epoch 5: Train=5.3057, Val=6.4692, PPL=644.99
```

Exp-5 (Batch size variation)



```
Epoch 1/5: 100%|██████████| 431/431 [00:40<00:00, 10.58it/s, batch_loss=6.8819]

Epoch 1: Train=7.4779, Val=7.2010, PPL=1340.81
Epoch 2/5: 100%|██████████| 431/431 [00:38<00:00, 11.18it/s, batch_loss=6.8068]

Epoch 2: Train=6.6148, Val=6.8783, PPL=971.00
Epoch 3/5: 100%|██████████| 431/431 [00:40<00:00, 10.73it/s, batch_loss=6.1704]

Epoch 3: Train=6.2197, Val=6.7306, PPL=837.68
Epoch 4/5: 100%|██████████| 431/431 [00:38<00:00, 11.05it/s, batch_loss=6.0755]

Epoch 4: Train=5.9571, Val=6.6482, PPL=771.43
Epoch 5/5: 100%|██████████| 431/431 [00:39<00:00, 10.85it/s, batch_loss=5.7464]

Epoch 5: Train=5.7577, Val=6.5886, PPL=726.77
```

Observations and Challenges

Across all five experiments, the model demonstrated consistent convergence behavior.The training and validation losses both decreased steadily over epochs, confirming that the transformer correctly learned next-token dependencies from the dataset. However, the overall perplexity (PPL) values remained relatively high (≈600–1300), indicating that the model still struggles to predict fluent continuations due to limited data scale and model capacity.

Exp-1 (Baseline, 128-dim, 2 layers): Training and validation loss decreased from ~7.5 to ~6.5, with PPL dropping from 1360 to around 690. The model converged stably and served as a solid reference point.
Exp-2 (Smaller model, 64-dim, 1 layer): Training was faster, but both loss and PPL (~950) plateaued early, suggesting underfitting due to insufficient model capacity.
Exp-3 (Larger model, 256-dim, 2 layers): Achieved the lowest final validation loss (~6.38) and PPL (~590), showing improved representational ability. However, training time roughly doubled.

Exp-4 (Higher learning rate 1e-3): Initially converged faster but produced unstable loss curves, occasionally spiking, implying that too large a learning rate harms stability.
Exp-5 (Larger batch size 64): Produced smoother loss curves and slightly faster convergence, but final validation loss (~6.58) was marginally higher than baseline, suggesting no clear advantage under current settings.

Challenges:

Due to limited dataset size (≈500 documents) and 128-token context, the model lacked enough data diversity for strong generalization. GPU constraints limited hyperparameter exploration and sequence length (block size), which affects learning long-range dependencies. Despite using layer normalization and GELU activation, the model's small scale prevented significant PPL reduction.