

CSE-256 PA2 Writeup

Tianyi Chen

Feburary 15 2026

Contents

| | |
|---|----------|
| 1 Part 1: Encoder Trained With Classifier (40 Points) | 2 |
| 1.1 Part 1.1: Encoder Implementation | 2 |
| 1.2 Part 1.2: Feedforward Classifier Implementation | 2 |
| 1.3 Part 1.3: Joint Encoder & Classifier Training | 2 |
| 1.4 Part 1.4: Sanity Checks | 2 |
| 1.5 Part 1.5: Evaluation | 2 |
| 2 Part 2: Pretraining Decoder Language Model (30 Points) | 4 |
| 2.1 Part 2.1: Decoder Implementation | 4 |
| 2.2 Part 2.2: Decoder Pretraining | 4 |
| 2.3 Part 2.3: Sanity Checks | 4 |
| 2.4 Part 2.4: Evaluation | 4 |
| 3 Part 3: Architectural Exploration (30 Points) | 6 |
| 3.1 Rotary Positional Embedding (RoPE) | 6 |
| 3.2 Disentangled Attention | 6 |

1 Part 1: Encoder Trained With Classifier (40 Points)

All the encoder experiments are logged on WandB: [Link-to-WandB](#)

1.1 Part 1.1: Encoder Implementation

Encoder part is a standard transformer encoder with bi-directional attention mechanism. The encoder is a stack of $L = 4$ identical transformer blocks. The input is first mapped from token IDs to vectors via an embedding layer. Each block consists of a pre-norm multi-head self-attention layer (with $h = 2$ heads and head dimension $d_k = 32$) followed by a pre-norm feed-forward network with hidden size $4d_{\text{model}} = 256$. Residual connections are applied after both sublayers. The encoder outputs a sequence of representations of shape (B, S, d_{model}) . The detailed config is shown in 1.

Table 1: Encoder configuration and dimension parameters (default values).

| Parameter | Symbol | Value |
|---------------------------|--|----------------|
| Vocabulary size | $ V $ | 5755 |
| Embedding dimension | d_{model} | 64 |
| Number of layers | L | 4 |
| Number of attention heads | h | 2 |
| Head dimension | $d_k = d_{\text{model}}/h$ | 32 |
| Maximum sequence length | S_{max} | 32 |
| Multi-head attention | $d_{\text{model}} \times d_{\text{model}}$ (Q,K,V,O) | 64×64 |
| FFN intermediate size | $4 \cdot d_{\text{model}}$ | 256 |

1.2 Part 1.2: Feedforward Classifier Implementation

The classification head is a single layer feed-forward network that maps the encoder representation to class log-probabilities. It is equipped with a single hidden layer with size of 100.

1.3 Part 1.3: Joint Encoder & Classifier Training

In the baseline model, the original encoder output is averaged along the dimension of sequence, obtaining a global representation for the whole sequence. The classification head then mapped the vector into log-probabilities. The predictions are compared against the true labels to compute a Negative Likelihood (NLL) loss.

1.4 Part 1.4: Sanity Checks

In sanity checks, all the attention maps are visualized in Fig. 1. Specifically, in Layer 2 Head 0, the attention is highly concentrated on a single key, showing a strong vertical stripe, which indicates that nearly all query tokens attend to the same token and suggests a global aggregation behavior.

Generally, across layers, the attention maps evolve from relatively diffuse and noisy patterns in lower layers to more structured and concentrated patterns in higher layers. Several heads exhibiting column-wise dominance that reflects increasing specialization and global information aggregation.

1.5 Part 1.5: Evaluation

The training curve and test accuracy after per epoch are shown in Fig. 2. **The final as well as the optimal accuracy is 85.3%.** The whole number of parameters are 564K including embedding. The breakdown is detailed in Tab. 2.

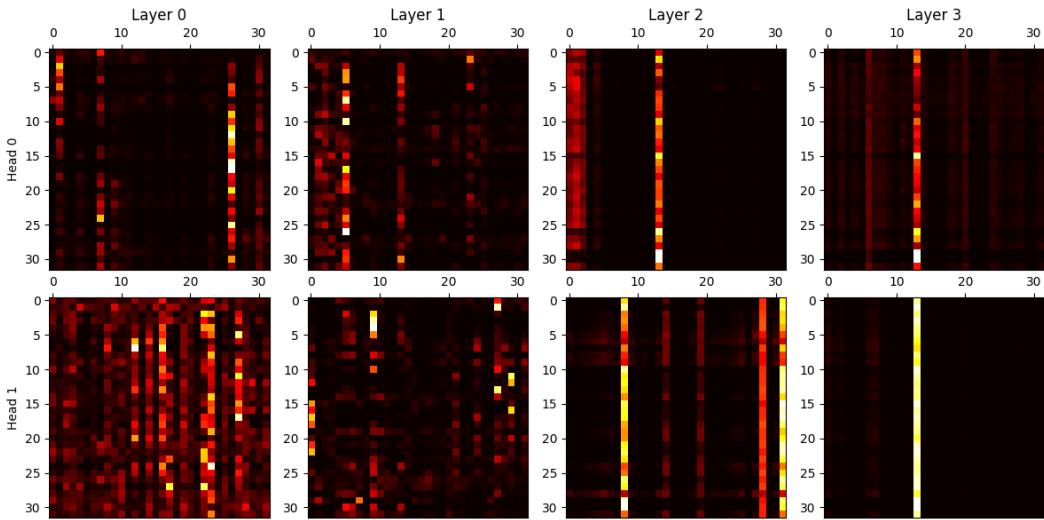


Figure 1: Attention maps in encoder.

Table 2: Parameter count breakdown of the encoder (default configuration).

| Component | Formula | Num |
|----------------------------------|---|-----------------------------------|
| Token embedding | $ V \cdot d_{\text{model}}$ | $5755 \times 64 = 368,320$ |
| Multi-head attention (per layer) | $4 \cdot d_{\text{model}}^2$ | $4 \times 64^2 = 16,384$ |
| FFN (per layer) | $2 \cdot d_{\text{model}} \cdot 4d_{\text{model}}$ | $2 \times 64 \times 256 = 32,768$ |
| Encoder layer total (per layer) | $4d_{\text{model}}^2 + 2d_{\text{model}} \cdot 4d_{\text{model}}$ | 49,152 |
| All encoder layers | $L \cdot 49,152$ | $4 \times 49,152 = 196,608$ |
| Total | — | 564,928 |

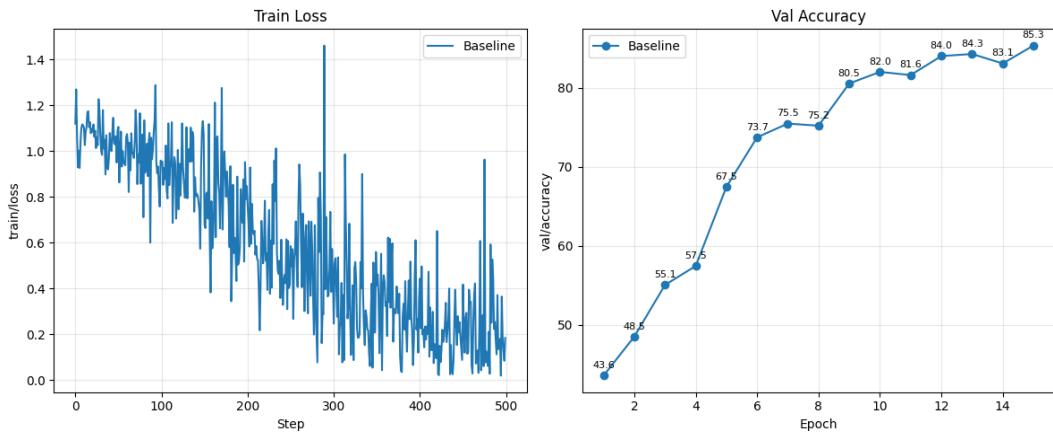


Figure 2: Baseline evaluation in train loss and test accuracy.

2 Part 2: Pretraining Decoder Language Model (30 Points)

All the decoder experiments are logged on WandB: [Link-to-WandB](#)

2.1 Part 2.1: Decoder Implementation

Decoder is a standard transformer decoder with causal attention mechanism. The whole structure is similar to Part 1.1. An additional causal mask is applied during attention computing.

2.2 Part 2.2: Decoder Pretraining

The model is trained using Cross-Entropy Loss. As shown in Fig. 3, **the final train perplexity is 155.1**.

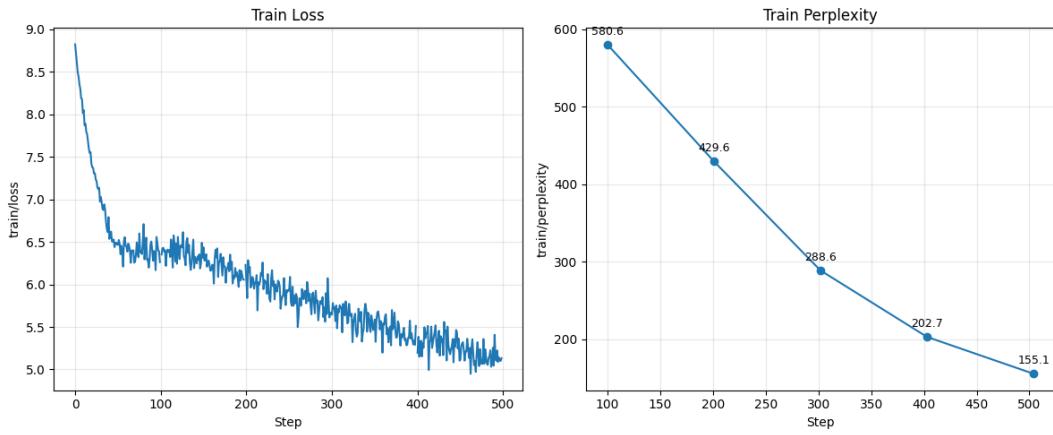


Figure 3: Training loss and perplexity in language modeling.

2.3 Part 2.3: Sanity Checks

All the attention maps are visualized in Fig. 4. Specifically, in Layer 0 Head 0, the attention is strictly confined to the diagonal, indicating that each token primarily attends to itself and nearby previous tokens under the causal mask.

Generally, across layers and heads, the attention maps consistently exhibit a clear lower-triangular structure due to autoregressive masking, with most heads showing a dominant diagonal while some develop sharper vertical stripes that suggest focus on particular earlier positions.

2.4 Part 2.4: Evaluation

As shown in Fig. 5, the lowest perplexity on W. Bush., Obama and H. Bush. is 429.8, 345.9 and 410.3 individually. Differences in perplexity mainly reflect distributional mismatch between the training data and each politician's speech style. From the quantitative results, the training corpus is more similar to Obama, that test set will naturally yield lower perplexity.

The number of parameters are most the same as Tab. 2, except for the FFN hidden size is 100. The number is 485,056 (485K) including the embedding.

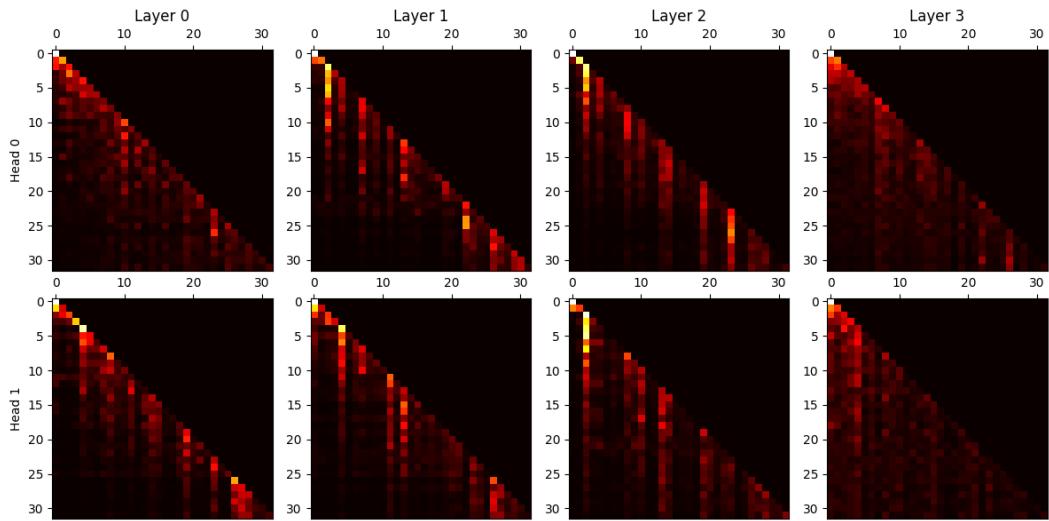


Figure 4: Attention maps in decoder.

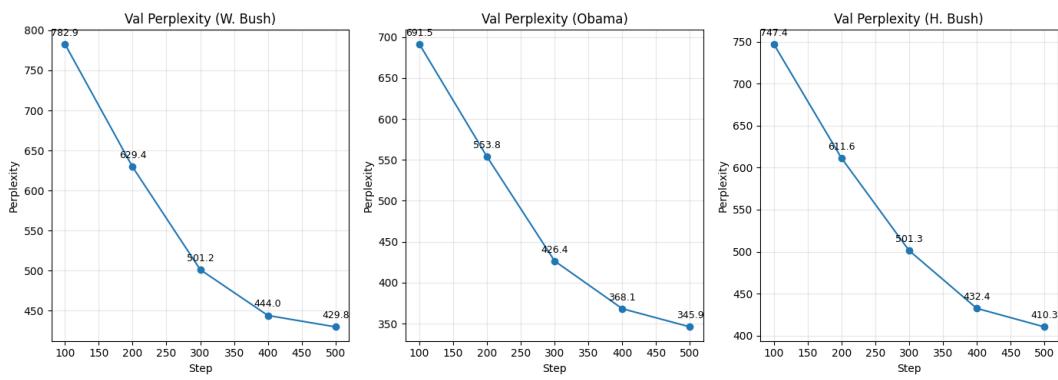


Figure 5: Test perplexity on W. Bush., Obama and H. Bush.

3 Part 3: Architectural Exploration (30 Points)

In this section, two architectural variations, RoPE and Disentangled Attention, are implemented to compare the performance with the original baseline.

3.1 Rotary Positional Embedding (RoPE)

Rotary Positional Embedding (RoPE) encodes positional information by applying a rotation to the query and key vectors in self-attention, enabling relative position modeling directly within the attention mechanism. As shown in Fig. 6, the encoder equipped with RoPE demonstrates faster training convergence and a more pronounced increase in validation accuracy. **The optimal accuracy reaches 86.5%, compared with 85.3% for the baseline.**

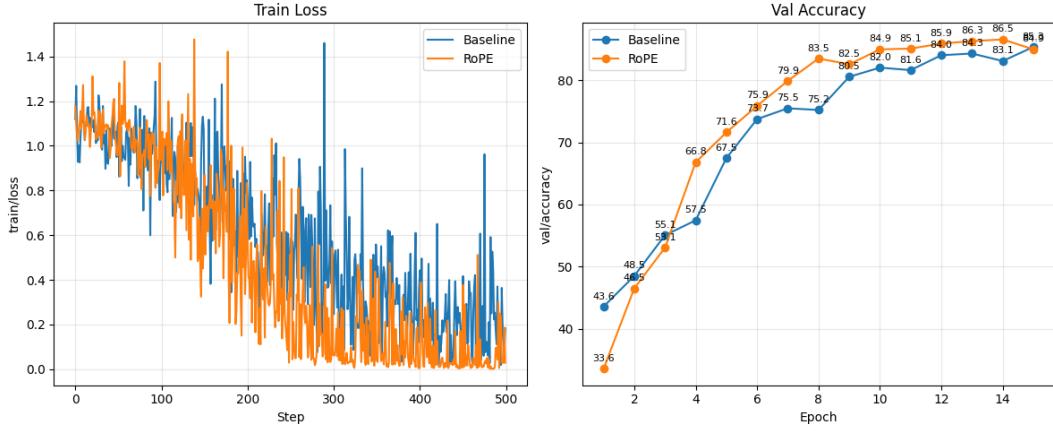


Figure 6: Comparison results between Baseline and RoPE.

3.2 Disentangled Attention

Disentangled Attention separates content and positional information into independent attention components, allowing the model to compute content-to-content, content-to-position, and position-to-content interactions explicitly rather than mixing them within a single dot product. As shown in Fig. 7, this variant achieves lower training loss and faster growth in validation accuracy compared with the baseline. **The optimal accuracy reaches 86.3%, compared with 85.3% for the baseline.**

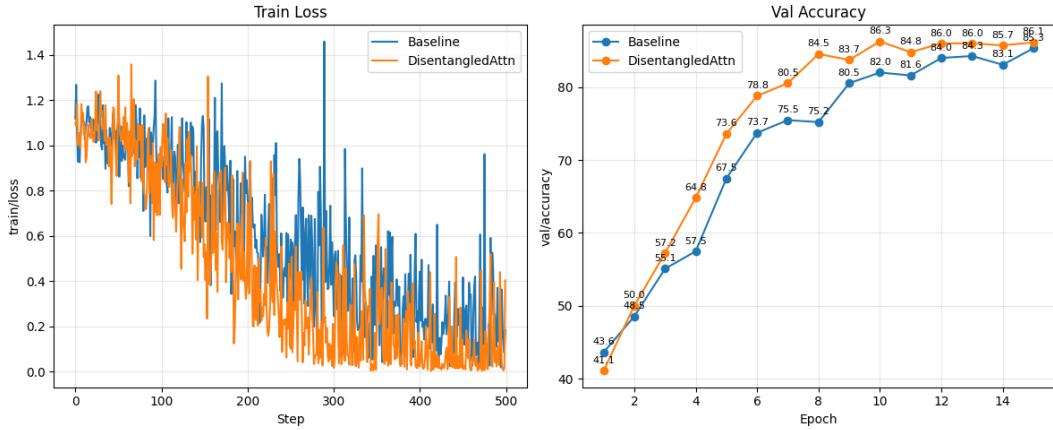


Figure 7: Comparison results between Baseline and Disentangled Attention.