# Assignment 1 - Comparing of BLT and Traditional Language Models


Prepared by

*Kushagra Gupta*

(2025909001)


Under the guidance of

*Prof. Vasudeva Varma*

*Prof. Vandan Mujadia*

Submitted in
partial fulfillment of the requirements
for the course of

**CL3410: Language Models and Agents**


INTERNATIONAL INSTITUTE OF
INFORMATION TECHNOLOGY

HYDERABAD


(October 2025)

# 1. INTRODUCTION

This report details the implementation and evaluation of a simplified Byte-Latent Transformer (BLT) model. Its performance on a string reversal task is compared against a standard character-level Transformer baseline. A key objective was to assess the BLT's potential computational efficiency derived from its entropy-based data patching mechanism.

# 2. METHODOLOGY

- **Task:** The models were trained on a string reversal task (e.g., Input: LMA is fun! -> Output: !nuf si AML).
- **Dataset:** Training was performed on train.csv (10,000 samples) and evaluation on test.csv (2,000 samples), containing printable ASCII strings.

### Baseline Model

- **Tokenization:** Employed character-level tokenization using printable ASCII characters plus special PAD, SOS, and EOS tokens.
- **Architecture:** A standard Transformer Encoder-Decoder with 2 encoder layers and 2 decoder layers, d_model=64, and nhead=4.

### BLT Model

- **Tokenization:** Utilized entropy-based patching with a sliding window (window_size=12). Patch boundaries were determined if window entropy exceeded 2.5 or patch length exceeded 20. Patch embeddings (embed_dim=64) were generated by summing hash-based n-gram embeddings (n=1, 2, 3) using 4096 buckets per n-gram size.
- **Architecture:** Implemented a 1-1-2 structure: 1 Transformer Encoder block, 2 Global Transformer Encoder blocks, and 1 Transformer Decoder block. Key parameters included d_model=64, nhead=4, and dropout=0.1.

### Training Details

- **Optimizer:** Adam optimizer was used (lr=1e-3 for Baseline, lr=1e-4 for BLT).
- **Loss Function:** Cross-Entropy Loss, ignoring the PAD index.
- **Epochs:** The Baseline model was trained for approximately 470 epochs, while the BLT model was trained for 372 epochs.

- **Hardware:** Training was conducted on an Apple M2 using the MPS backend

# 3. RESULTS

### Final Performance Metrics

The performance of the BLT and baseline models on the test set is summarized below:

| Metric | Baseline Model | BLT Model |
|---|---|---|
| **Token Accuracy (%)** | **97.77%** | **8.39%** |
| Avg. Input Length (chars) | 70.85 | 70.85 |
| Avg. Predicted Output Length (chars) | 70.85 | 78.03 |
| **Avg. Sequence Length (Test)** | **70.85 chars** | **6.36 patches** |

### Training Curves (BLT Model)

The training progression for the BLT model over 372 epochs is shown below.

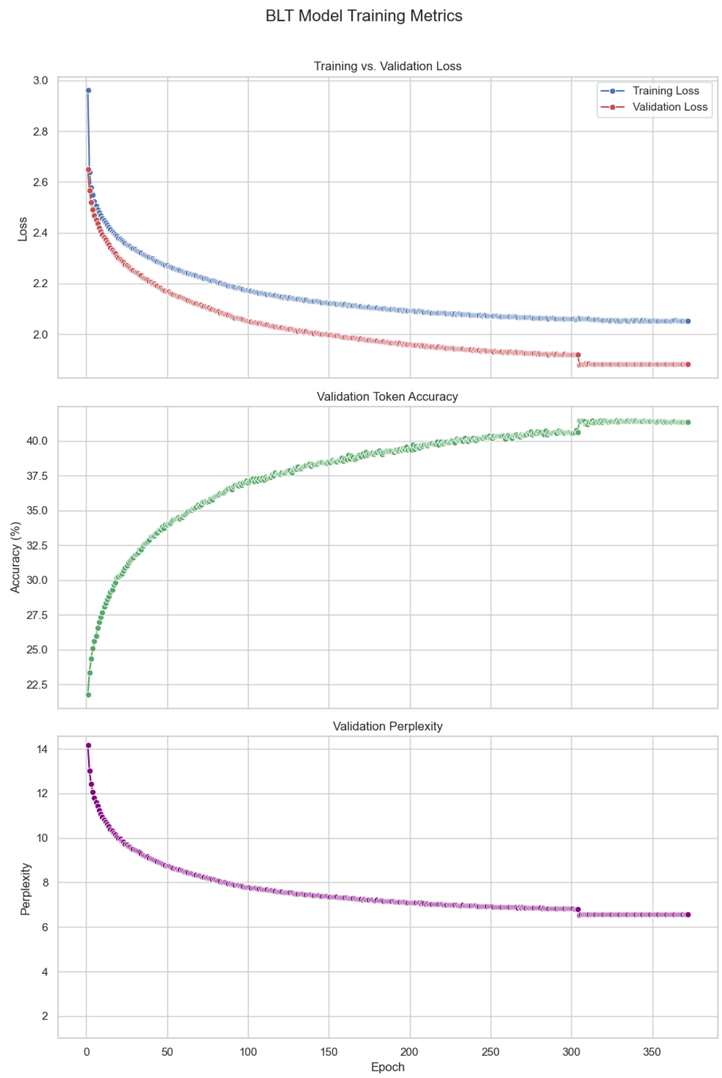


Figure 1: BLT model training loss, validation loss, accuracy, and perplexity over 372 epochs. Validation metrics plateaued after approximately 300 epochs, indicating that further training yielded minimal improvement.

## Hyperparameter Summary

- Baseline Model:
  - **Architecture:** d_model=64, nhead=4, num_encoder_layers=2, num_decoder_layers=2
  - **Training:** lr=1e-3, batch_size=32, epochs_trained=~470, Optimizer=AdamW
- BLT Model:
  - **Architecture:** d_model=64, nhead=4, dropout=0.1, encoder_layers=1,

global_layers=2, decoder_layers=1

- **Tokenization:** window_size=12, entropy_threshold=2.5, max_patch_len=20, num_buckets=4096
- **Training:** lr=1e-4, batch_size=8, epochs_trained=372, Optimizer=Adam

## 4. DISCUSSION

### Performance Comparison

The **Baseline character-level model significantly outperformed the BLT model**, achieving 97.77% token accuracy compared to the BLT's 8.39%. While the BLT model demonstrated initial learning, its performance **plateaued early** at a low accuracy level, as seen in the training curves.

### Computational Efficiency

Theoretically, BLT aims for efficiency by reducing sequence length (average 70.85 chars vs. 6.36 patches), potentially speeding up the Transformer's self-attention. However, in this implementation, the **considerable overhead** from entropy calculation and hash n-gram embedding (multiple lookups and sums per patch) likely **negated any benefits** for these relatively short sequences. The simpler character-level baseline was likely more computationally efficient overall for this task. The BLT's poor convergence also hints at potential inefficiencies in learning from the patched representation.

### Limitations & Observations

Several factors may have contributed to the BLT model's limited performance:

- The model's performance plateaued, suggesting the chosen d_model=64 might offer **insufficient capacity** for the reversal task, or hyperparameters require further tuning.
- The hash n-gram embedding process might **lose crucial positional information** necessary for accurate sequence reversal.
- **Extended training** beyond the plateau yielded no significant improvements.

## 5. CONCLUSION

In summary, the standard character-level Transformer baseline demonstrated **strong performance**, achieving high accuracy (97.77%) on the string reversal task. The implemented Byte-Latent Transformer (BLT) model showed signs of learning but ultimately **performed poorly**, plateauing at a low token accuracy of 8.39%.

While the BLT's patching mechanism successfully reduced the average sequence length (from 70.85 chars to 6.36 patches), offering a theoretical path to improved computational efficiency, this advantage was likely negated. The significant **pre-processing overhead** of entropy calculation and hash n-gram embedding, combined with the model's **convergence difficulties**, rendered the BLT less effective than the simpler baseline for this specific task and implementation.

## 6. APPENDIX

- **Links to Saved Models:** Provide Google Drive links to saved .pt files for both the final BLT model (blt_epoch372.pt) and the baseline model (char_transformer_epoch500.pt).
https://drive.google.com/drive/folders/1BIsIaQ5GCVY-t-xp34LlR9IR8gN2eVGx?usp=share_link
- **Github Repo:** All the data, code file(single ipynb), checkpoints training logs, and predictions.
https://github.com/ltrc/lma-assignment-1-blt-SpyBeast07

**--- END ---**