FilinGPT: A Byte-Level Financial Language Model

Kevin Mota da Costa

February 20, 2026

Abstract

This document presents *FilinGPT*, a byte-level autoregressive language model trained on financial 10-K filings. The system was implemented from scratch using NumPy, including training, inference, dataset pipeline, and evaluation tools.

Two training configurations are compared: a short baseline (200 steps) and a full financial checkpoint (100,000 steps). Quantitative and qualitative evaluations demonstrate substantial convergence and language structure acquisition.

1 Introduction

FilinGPT is a minimal autoregressive language model designed to:

- Operate at byte-level (vocabulary size = 258),
- Learn financial language patterns from 10-K filings,
- Provide reproducible training and evaluation pipelines,
- Demonstrate measurable convergence using perplexity.

The project emphasizes transparency and explicit implementation rather than reliance on high-level deep learning frameworks.

2 Model Architecture

The model is a simple multi-layer perceptron (MLP) trained for next-token prediction.

Given a context window of length $L = 16$, the model predicts the probability distribution over the next byte.

2.1 Input Representation

Each input token is represented as a one-hot vector:

$$x_t \in \mathbb{R}^V$$

where $V = 258$ is the vocabulary size.

The context window is flattened:

$$X \in \mathbb{R}^{L \cdot V}$$

2.2 Forward Pass

The model computes:

$$h = \sigma(W_1 X + b_1)$$

$$z = W_2 h + b_2$$

$$\hat{y} = \mathrm{softmax}(z)$$

where σ is a non-linear activation function.

3 Training Objective

The model is trained using cross-entropy loss:

$$\mathcal{L} = -\sum_{i=1}^{V} y_i \log(\hat{y}_i) \tag{1}$$

Perplexity is defined as:

$$\mathrm{PPL} = \exp(\mathcal{L}) \tag{2}$$

Perplexity measures how well the model predicts the next token.

4 Dataset Pipeline

The training data is built from SEC 10-K filings:

- Bronze: raw filings
- Silver: cleaned full documents
- Gold: extracted Management Discussion and Analysis (MDA)

Pipeline steps:

1. Extract MDA sections
2. Chunk documents
3. Byte-level tokenization
4. Batch construction

Dataset statistics:

- Vocabulary size: 258
- Context length: 16
- Sequence length: 256
- Total sequences: 609

5 Training Configurations

5.1 Baseline (200 Steps)

- Steps: 200
- Learning rate: 0.01
- Final loss: 5.3182
- Final perplexity: 204.02

5.2 Financial Checkpoint (100k Steps)

- Steps: 100,000
- Learning rate: 0.02
- Final loss: 0.8071
- Final perplexity: 2.24

Perplexity reduction:

$$99.13\% \text{ reduction}$$

6 Quantitative Results

6.1 Perplexity (Full Training)

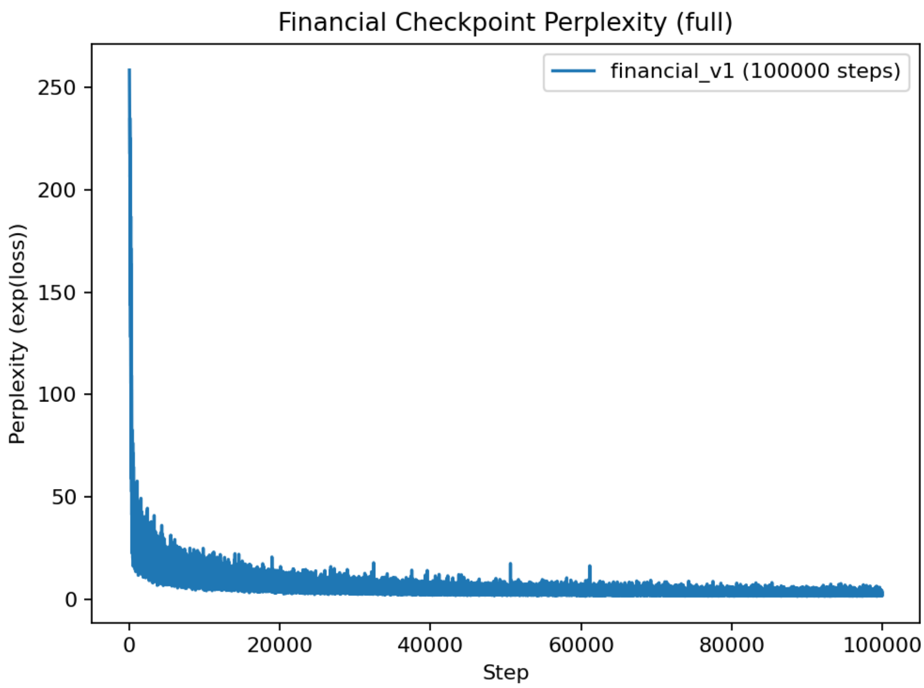


Figure 1: Perplexity evolution during full 100k-step training.

6.2 Loss Comparison (Log Scale)

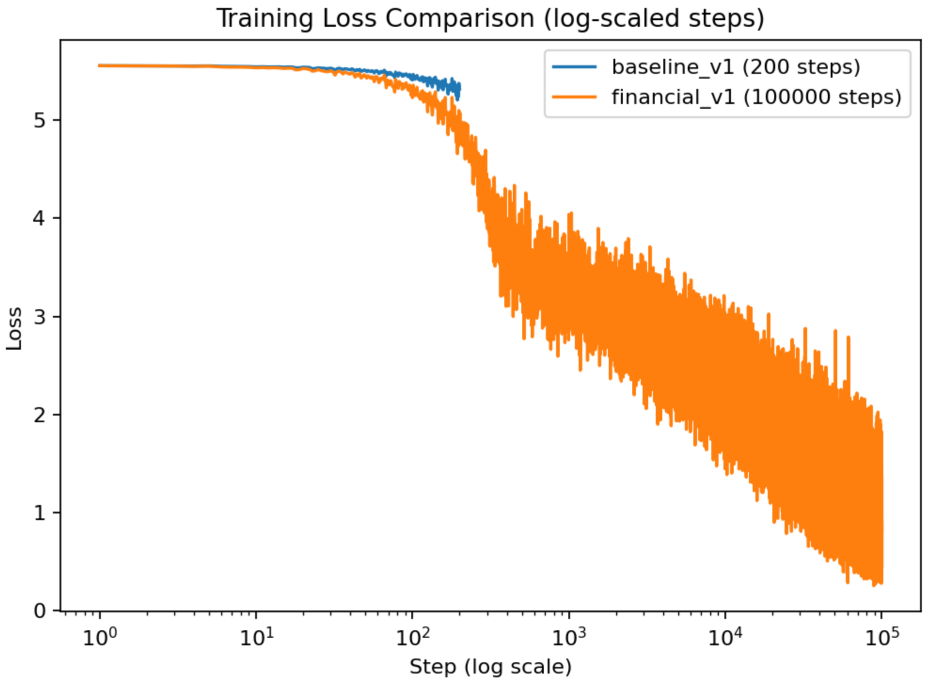


Figure 2: Loss comparison with log-scaled steps.

6.3 Perplexity Comparison (First 200 Steps)

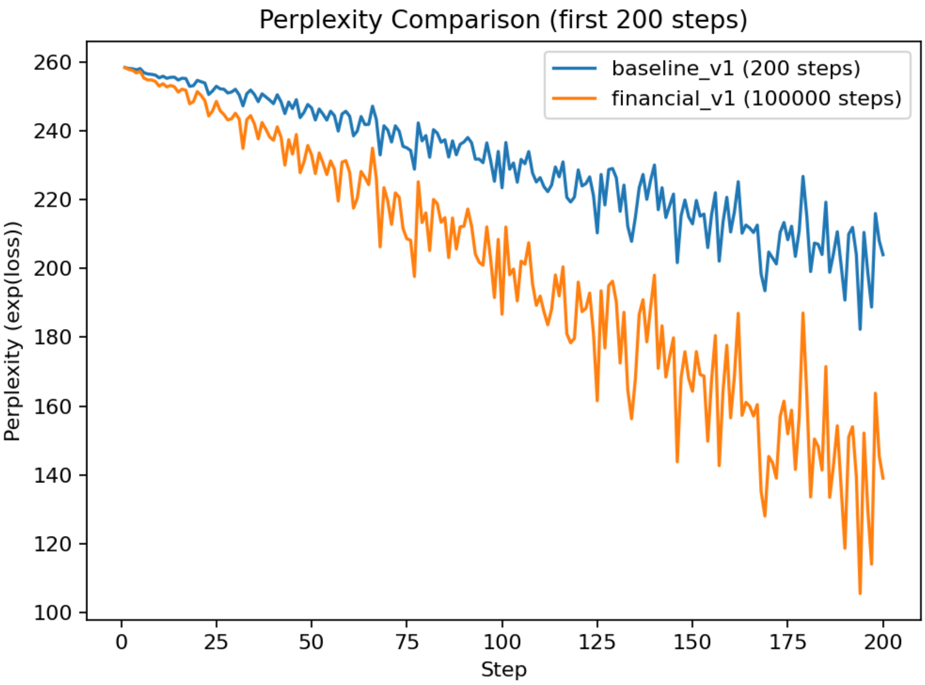


Figure 3: Perplexity comparison during first 200 steps.

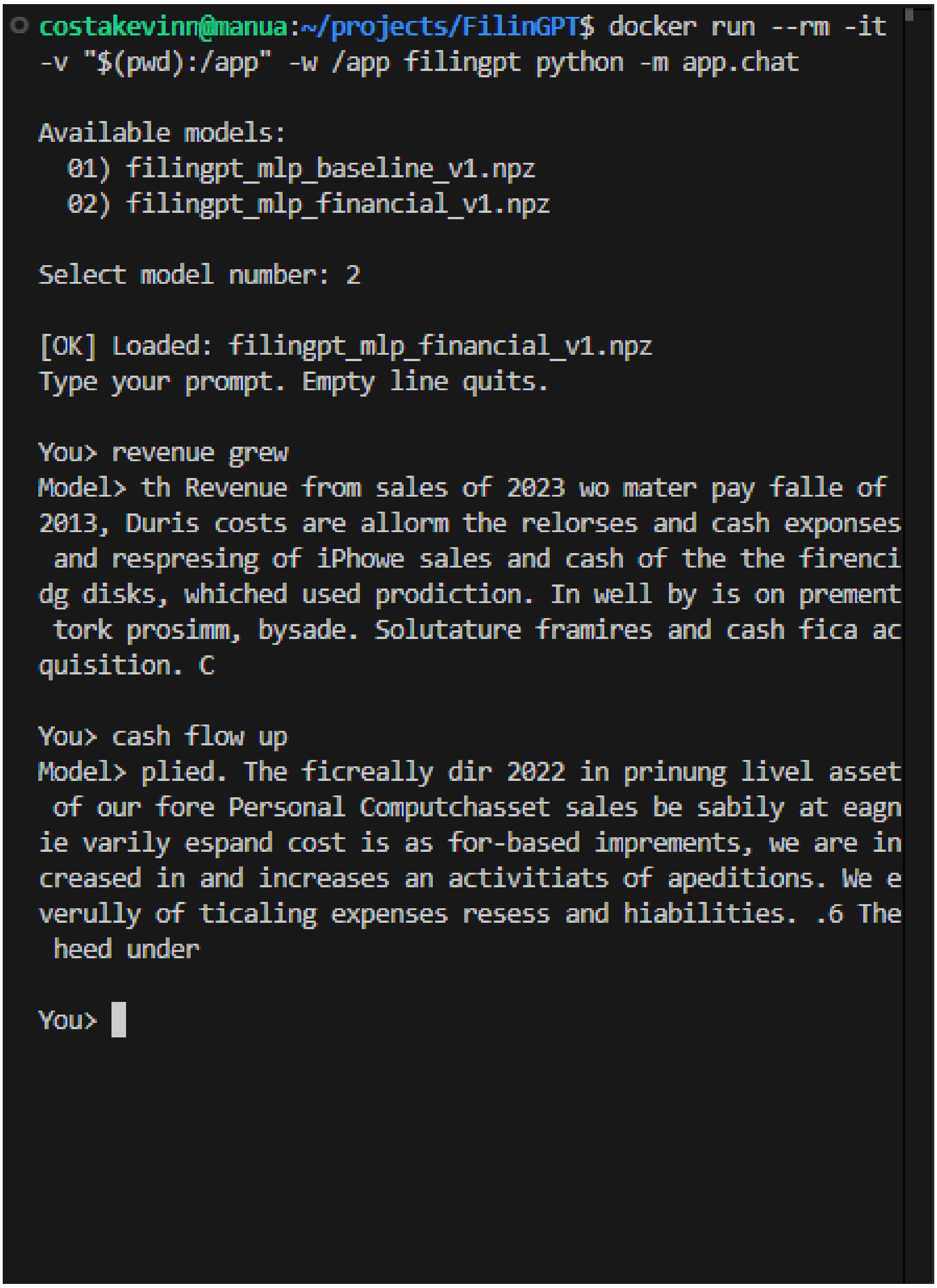
7 Qualitative Evaluation

7.1 Baseline Output

Figure 4: Baseline model output after 200 steps.

The baseline model produces mostly incoherent byte-level noise.

7.2 Financial Model Output



```
costakevinn@manua:~/projects/FilinGPT$ docker run --rm -it
-v "$(pwd):/app" -w /app filingpt python -m app.chat

Available models:
  01) filingpt_mlp_baseline_v1.npz
  02) filingpt_mlp_financial_v1.npz

Select model number: 2

[OK] Loaded: filingpt_mlp_financial_v1.npz
Type your prompt. Empty line quits.

You> revenue grew
Model> th Revenue from sales of 2023 wo mater pay falle of
2013, Duris costs are allorm the relorses and cash exponses
 and respresing of iPhowe sales and cash of the the firenci
dg disks, whiched used prodiction. In well by is on prement
 tork prosimm, bysade. Solutature framires and cash fica ac
quisition. C

You> cash flow up
Model> plied. The ficreally dir 2022 in prinung livel asset
 of our fore Personal Computchasset sales be sabily at eagn
ie varily espand cost is as for-based imprements, we are in
creased in and increases an activitiats of apeditions. We e
verully of ticaling expenses resess and hiabilities. .6 The
 heed under

You>
```

Figure 5: Financial checkpoint output after 100k steps.

The financial model produces structured English text with recognizable financial terminology, showing learned statistical patterns.

8 Discussion

The comparison between baseline and full training highlights:

- Stable convergence of cross-entropy loss,
- Massive perplexity reduction (258 $\rightarrow$ 2.24),
- Emergence of structured financial language,
- Clear qualitative improvement in generation.

Despite its simplicity, the model demonstrates that even small MLP architectures can capture structured domain-specific language when trained sufficiently.

9 Conclusion

FilinGPT demonstrates:

- End-to-end language model training from scratch,
- Byte-level autoregressive modeling,
- Reproducible data and training pipelines,
- Quantitative evaluation using perplexity,
- Observable emergence of domain language structure.

This project serves as a foundational implementation of language modeling principles applied to financial documents.