



Training a Small Language Model

Tokenizers, Training, and Real-World Pitfalls

David vonThenen
Senior AI/ML Engineer



[@davidvonthenen](https://twitter.com/davidvonthenen)



David vonThenen

- Are you Human or an AI?
- I want 5 Kubernetes
- Virtual Machines are Real
- Cloudy, cloudy, cloudy...
- There is storage for that!

     [@davidvonthenen](https://twitter.com/davidvonthenen)



Agenda

- **(Small) Language Model 101**
- **Building A Language Model From Scratch**
 - **Live Demo!**
- **Recommended Way: Fine-Tune (& Quantize)**
 - **Live Demo!**
- **Bonus: Many Types Of Language Models**
 - **Live Demo!**
- **Q&A**

(Small) Language Models 101

The Basics and Process...

What Is A Small Language Model?

- **It's Like A Large Language Model...**
 - **Processing, Understanding, and Generating Natural Language Content**
 - **Generate Answer To User's Question**
 - **Predicts "What Comes Next" In Text**
 - **SLM = Subset Of Models That Require Few Resources To Run**
 - **Modest or Low-End GPUs**
 - **Even On CPU (Accelerator Instructions)**
 - **GOAL:**
 - **Good Quality (Latency, Mem)**
 - **Predictable/Lower Cost (Watts/Power)**



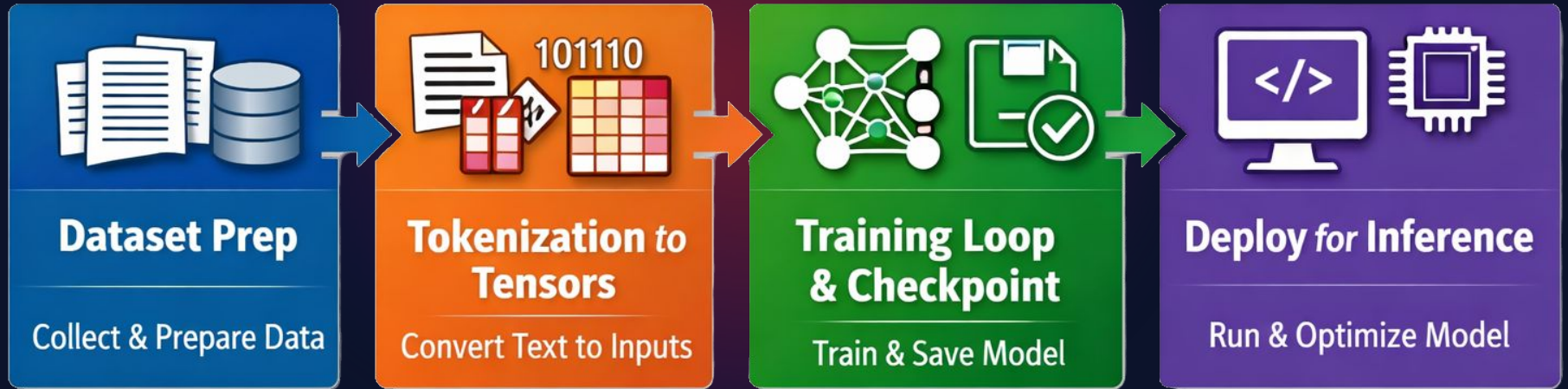
Image Attribution

What Are Small Language Models? Real World Example and Training Data

<https://www.shaip.com/blog/small-language-models-real-word-example-and-training-data/>

Our Project's Mental Map

Building A nanoGPT-Style Causal LM From Scratch



Typical "GPT" Model Architecture

Blueprint for our nanoGPT-style Causal LM:

- Input → IDs: Text Converted To Token IDs
- Each Token ID Becomes a Vector
- **Attention**: Decides What Prev. Tokens Matter
- **Transformer**: Context in // → Relationships
- **Prediction Head**: State → Ranked Next Tokens
- **Autoregressive Loop**: Appends Token, Repeat

Why "Small" in Small Language Model?

- Lower Parameter Count
- Limited Context Window
- (Smaller) Vector Dimensions (aka Reasoning Capacity)



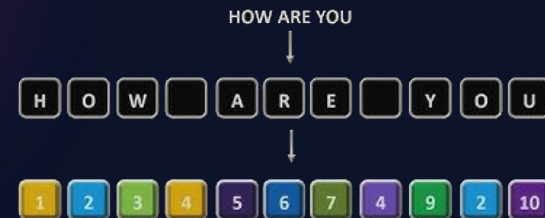
Dataset Prep

- Dataset For (Small) Language Model
 - TON of Text Documents
- For Our nanoGPT-style Causal LM
 - Small/Clean Dataset (Ex: [TinyStories](#) 1.6GB)
 - Exp: [FineWeb-Edu](#) (10.4TB)
 - [Paper](#) Discuss Time/Effort Curate DS
- Train/Validation/Test Datasets
 - Train – Update Weights (Practice)
 - Validation – Tune Hyperparameters (Coach)
 - Test – How Well We Did (The "Final Exam")



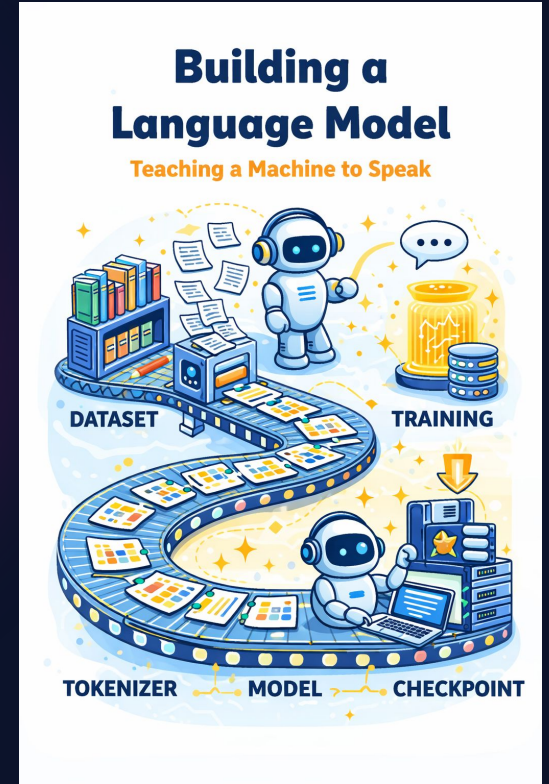
Tokenization

- Computers Don't Learn Using Text
 - Tokenizers Break Text Into Smaller Pieces
 - Typically Sub-Word, Word or Multi-Word
 - Ex: Byte-Pair Enc., WordPiece, SentencePiece
- Assigns a "Number" To The Token
- Those Numbers Are The "Alphabet"
- Change The Tokenize, Change The "Language"
- Think Of This As An Encoding
 - Lower Level: UTF8 versus Unicode
 - Higher Level: English versus Spanish



The Training Process

- What Is Model Training?
 - Predict → Compare To Truth → Adjust
- Score The Model Over Iterations
 - Lower Score Is Better
 - Save Our PyTorch Checkpoint
- The Knobs That Matter:
 - Learning Rate
 - Loss Func (Quantify Diff Truth → Answer)
 - Reinforce Answers with Better Score
 - Perplexity: How Wrong On Average
 - Memory/Time Limits



Let's Build The Model...

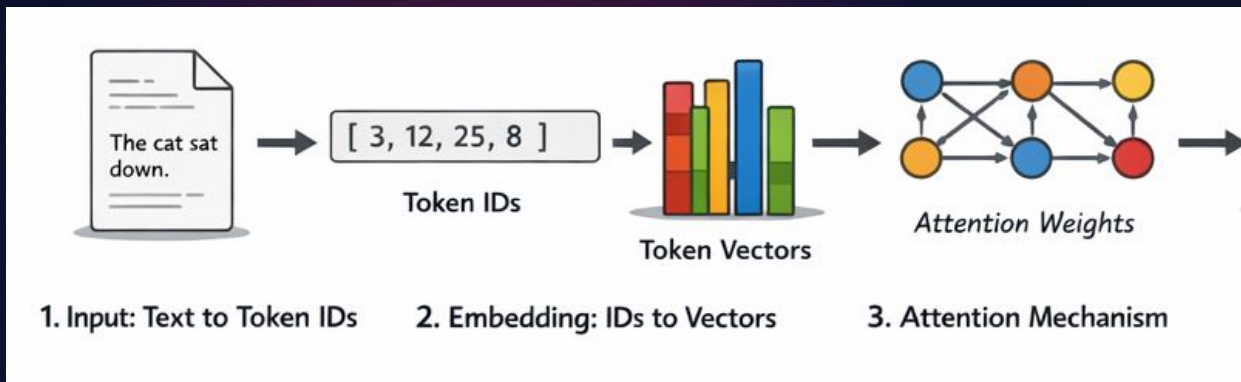


Image Attribution
ChatGPT Generated



Naively Attempting This Like ML...



EPIC FAIL

For some things, there's just no excuse

Image Attribution
https://wikioflawness.fandom.com/wiki/Epic_Fails

Out Of Memory

```
[rank7]: torch.OutOfMemoryError: CUDA out of memory. Tried to allocate 3.07 GiB. GPU 7 has a total capacity of 79.19 GiB of which 1.30 GiB is free. Including non-PyTorch memory, this process has 77.88 GiB memory in use. Of the allocated memory 74.88 GiB is allocated by PyTorch, and 1.39 GiB is reserved by PyTorch but unallocated.
```

If reserved but unallocated memory is large try setting

```
PYTORCH_CUDA_ALLOC_CONF=expandable_segments:True to avoid fragmentation. See documentation for Memory Management (https://pytorch.org/docs/stable/notes/cuda.html#environment-variables)
```

terminate called without an active exception

```
...
```

```
W0115 02:57:42.062000 14986 torch/distributed/elastic/multiprocessing/api.py:899] Sending process 15053 closing signal SIGTERM
```

```
...
```

```
E0115 02:57:43.329000 14986 torch/distributed/elastic/multiprocessing/api.py:873] failed (exitcode: -6) local_rank: 6 (pid: 15059) of binary: /usr/bin/python3
```

Building A GPT-Style LM

Easier Said Than Done...

Single H100 (or H200) Isn't Enough!

- This Is Super Comical
 - Almost Immediately Crashed Out
 - Not Enough Memory
 - Dataset Too Large
- Switched To Single Node 8x H100
 - EXPENSIVE For An Experiment
 - Cost: \$24 to 28 Per Hour
 - Will Require 1000s Of Hours
 - We will quit early...



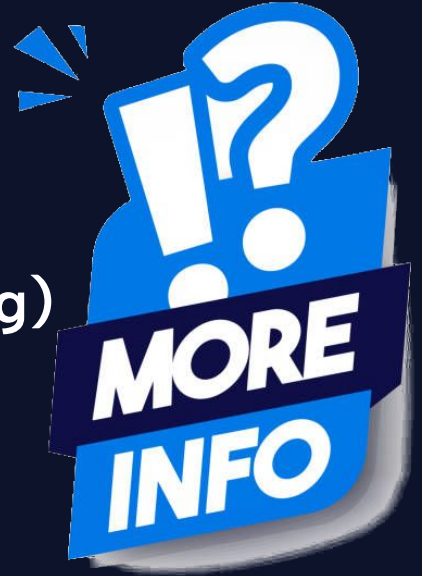
Streaming And Chunking

- The FineWeb-Edu (10.4TB) Is Too Large
 - Can't Load Everything Into Memory
 - Static Load Will Never Work
- Switched To Streaming
 - StreamingTokenDataset Interface
 - Storage Solutions/Integrations
 - Deterministic Split
 - Streaming Shuffle (Approximate)
 - No Epochs, Step Based
 - Epoch = 1 Pass Thru Data, Step = Batch Processing



More Info...

- Using [tiktoken](#) Tokenizer (GPT2 byte-pair encoding)
- Matched GPT2 Vocab: 50,304 Vocab
 - Pad To Multiple Of 64
- OpenAI's [GPTConfig](#)
 - block_size: 2048 ("Look Back" Window)
 - n_layer: 24 (Transformer Block = Reasoning)
 - n_head: 32 (Attention Heads = Lanes)
 - n_embd: 2048 (Vector Dimensionality)
- [FineWeb-Edu](#) (10.4TB) -> 1.8B Params



Demo: SLM From Scratch

<https://youtu.be/VUuVro-Dv7c>

More Info...

**LET'S BUILD GPT.
FROM SCRATCH.
IN CODE.
SPELLED OUT.**



The image features a black and white photograph of Andrej Karpathy on the right side, smiling and pointing towards a diagram of a neural network architecture. The diagram is a flowchart showing the flow of data through various layers, including 'Input Embedding', 'Attention', 'Feed Forward', and 'Output'. There are also some yellow starburst graphics above the diagram. Below the text, there are three fire emojis.

Video Attribution

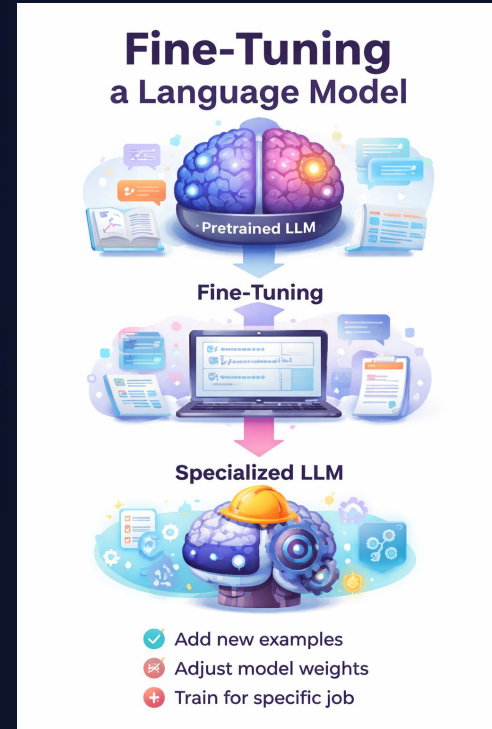
Andrej Karpathy - Formerly OpenAI & Director of AI and Autopilot Vision at Tesla

<https://www.youtube.com/watch?v=kCc8FmEb1nY>

**Recommended Way:
Fine-Tune (And Quantize)
The (Much) Easier Way...**

What is Fine-Tuning?

- **Start With A Pre-Trained Model**
 - Generalist That Knows Language (Qwen 2.5)
- **Specialize The Model To One Specific Job**
 - Create A Subject Matter Expert For The Function
 - Adding And Modify Weights
- **Same Training Loops (Epoch v Steps)**
 - Predict → Compare → Adjust
- **Overall: Change The Purpose Of Model**
 - Analogy: Fork Proj → Feature Branch → 1 Purpose



What Is Quantization?

- Storing Model Weights Using Smaller Numbers
 - Example: FP32 → INT8, INT4, FP16, BF16
- What Are The Goals?
 - Smaller Memory Footprint
 - Faster Inference
 - Less Expensive GPUs Or CPU Only
- What Doesn't Change?
 - Context Window, Attention Heads, Layers, etc
- How Is This Achieved?
 - Changing How Precise/Type The Numbers Are



3 Methods for Quantization

1. Quantization Aware Training

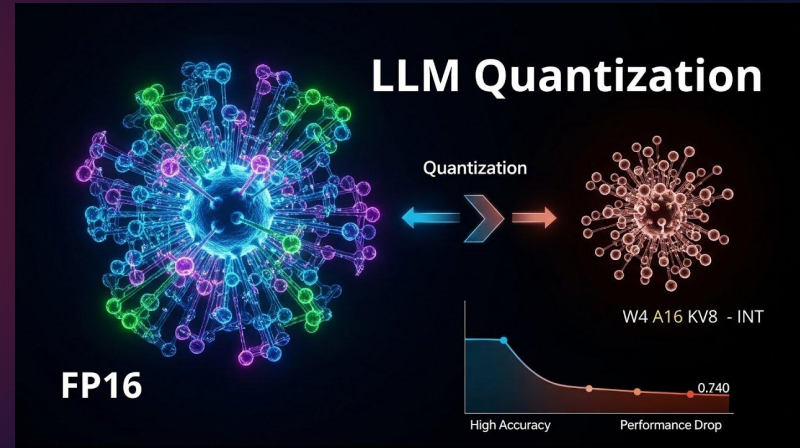
- *Native* Training Weights/Activations
- Unique Training Process, Low Effort

2. Static/Post-Train Quantization

- Requires Calibration Dataset
- More Work, Good Accuracy

3. Dynamic Quantization

- No Data Or Retrain Needed
- Runtime Overhead
- Not Optimal For Very Small Models



TLDR: Quantization



Image Attribution

What is LLM Quantization - Condensing Models to Manageable Sizes -

<https://www.exactcorp.com/blog/deep-learning/what-is-quantization-and-llms>

The Process...

LLM Fine-Tuning Workflow



MOBISOFT
360-DIGITAL REALITY

Training A SQL SLM Model

Our Fine-Tuning & Quantization Process

- Base Model: Qwen2.5-7B-Instruct
- Dataset: Schema + Question → SQL
 - Download: synthetic text to sql
- Fine-tune Strategy: LoRA adapters
 - Freeze Weights → Attention Projections (Updates)
- Quant Strategy: INT8
- Prompt Format: HF apply chat template

SLM That Generates SQL Statements!



Demo: FineTune & Quant

<https://youtu.be/QqZxWhQoTUw>

Lessons Learned

- MUCH (MUCH MUCH) Cheaper
 - \$100s Versus Multiple \$10,000s
- Leverage An Existing SLM/LLM
 - Select A "Suitable" Base Model
- "Teach" The Specialization
- (Optional) Quantize If Needed
 - Reduce Resource (GPU?), Speed, Mem
- Better Models Come Out All The Time
 - Iterative Approach
 - Cheaper To Update

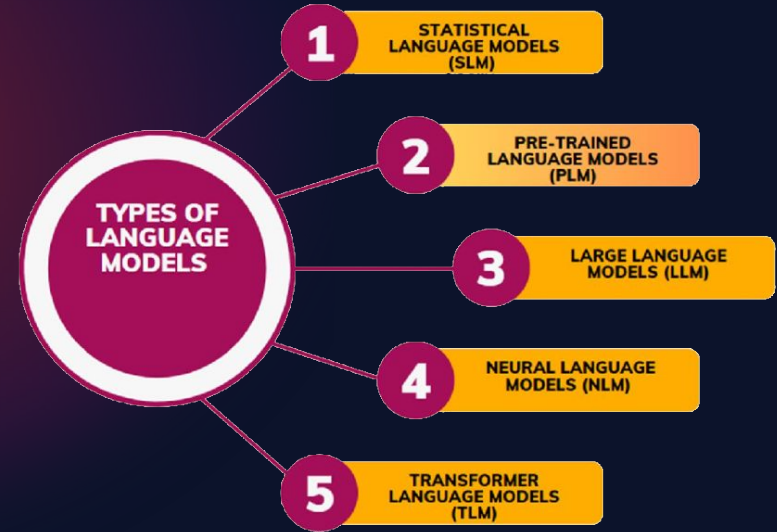


Bonus: Many Types Of LMs

Beyond Chat

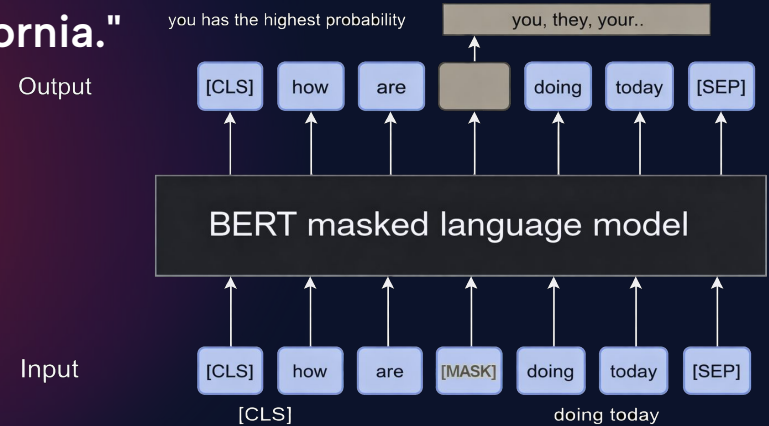
Types Of Language Models

- Most People Are Familiar With Q&A Chatbots
- Many Different Types:
 - Summarization
 - Code Generation
 - MLM – Our Next Demo
 - Classifiers (Example: NER)
 - Text-To-Vectors: Semantic Search
 - Etc, Etc, Etc
- Create Your Own!
 - Input → Computation → Output



Example: Masked Language Model

- What Does An MLM Do?
 - "Fill-In-The-Blank" Model
 - "The LBC stands for Long [MASK], California."
- Predict The "[MASK]" Value
 - Read Whole Sentence → Predict
 - Top-K, Top-P, etc
- Applications Of MLMs:
 - Extraction: Find Names, IDs, etc
 - Classification: Spam Detect, Routing
 - Search/Ranking: Doc Best Matches Query
 - ??????????



Demo: More SLMs

<https://youtu.be/PvncXxCWskM>

Check Out My Other Session...

The Sound of Your Secrets: Teaching Your Model to Spy, So You Can Learn to Defend



David vonThenen / Senior AI/ML Engineer / NetApp

Audience: Intermediate

Topic: Security

The presentation will take place in Room 101 on Saturday, March 7, 2026 - 14:30 to 15:30

Resources

Resources

All Materials (Slides, Code, Instructions, etc):

github.com/davidvonthenen/2026-scale-23x-slm

ODSC East (Apr): [How Model Quantization Fuels the Next Wave of Agentic AI](#)

Blog: [Less Compute, More Impact: How Model Quantization Fuels the Next Wave of Agentic AI](#)

S3 High-Throughput Streaming + Kafka

- [Amazon FSx for NetApp ONTAP \(FSxN\)](#)
 - Access Existing File Volumes as S3 Buckets
 - Feed Datasets Seamlessly Into AWS Services
- [NetApp StorageGRID](#)
 - 20x Throughput, Scalability, Tiering, Branching

Let's Chat on Discord: discord.gg/NetApp





Thank You!



David vonThenen
Senior AI/ML Engineer



[@davidvonthenen](#)