Advanced Transformers

NLP Course 2025

September 30, 2025

From BERT to GPT-3: The Scaling Revolution

When Size Started to Matter

The Discovery

- Scaling **Warning: changes everything**
- Emergent abilities appear
- **Quality** from quantity
- Power laws rule

The Models

- T5: **11B parameters**
- GPT-3: **175B parameters**
- Switch: **1.6T parameters**
- Compute as currency

The Impact

- Few-shot learning works
- In-context learning emerges
- Task-agnostic models
- AI becomes mainstream

The moment language models became foundation models

Bigger is Different

The Kaplan Scaling Laws (2020)

$$L = aN^{-\alpha} + bD^{-\beta} + L_{\infty}$$

Where:

- N = number of parameters
- D = dataset size (tokens)
- $\alpha \approx 0.076$, $\beta \approx 0.095$
- Loss decreases predictably with scale

The Chinchilla Laws (2022)

Compute-optimal training: $N \propto D^{0.5}$

Key insight:

- Most models are **Warning: undertrained**
- Need 20 tokens per parameter
- Smaller models + more data = better
- Changes entire industry approach

From "make it bigger" to "train it longer"

Checkpoint: Understanding Scaling Laws

Quick Check: Test Your Understanding

Question 1:
What's the key insight from Chinchilla scaling laws?

- A) Bigger models always better
- B) 20 tokens per parameter optimal
- C) More layers = more performance
- D) Dataset size doesn't matter

Question 2:
Why do scaling laws matter?

- A) They predict loss predictably
- B) They reduce training cost
- C) They make models smaller
- D) They eliminate overfitting

Answers:

Answer 1: B - 20 tokens per parameter

- Most models were undertrained
- Chinchilla: compute-optimal training
- Changed industry from "make it bigger" to "train it longer"

Answer 2: A - Predict loss predictably

- $L = aN^{-\alpha} + bD^{-\beta}$
- Loss decreases with scale
- Enables planning investments
- Power laws guide research

Key takeaway: Scaling follows predictable laws, but optimal training requires balance between model size and data
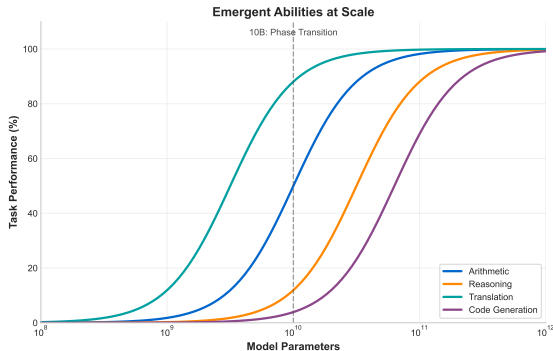
What Are Emergent Abilities?

Capabilities that:

- Appear **suddenly** at scale
- Were **Warning: not** explicitly trained
- Show sharp phase transitions
- Cannot be predicted from smaller models

Examples at Different Scales

Parameters	Emergent Ability
1B	Basic syntax
10B	Multi-step reasoning
50B	Chain-of-thought
100B+	In-context learning

Emergent Abilities at Scale

The Mystery

Nobody knows why:

- Sharp transitions occur
- Specific scales matter
- Some tasks need 100B+
- Others emerge at 1B

The Unified Framework
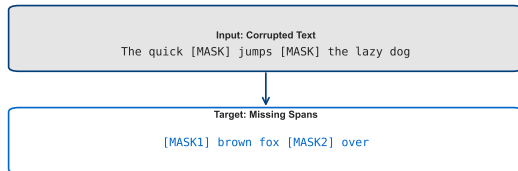
Every task as text-to-text:

- Translation: **"translate English to French: hello"**
- Summarization: **"summarize: [article]"**
- Question: **"question: what is NLP?"**
- Classification: **"sentiment: great movie"**

Architecture Choices

Component	Decision
Model	Encoder-decoder
Size	60M to 11B
Objective	Span corruption
Dataset	C4 (750GB text)

Key Innovation: Span Corruption

Input: Corrupted Text
The quick [MASK] jumps [MASK] the lazy dog

Target: Missing Spans
[MASK1] brown fox [MASK2] over

Performance Impact

- SOTA on 20+ benchmarks
- Single model, many tasks
- Better than task-specific models
- Scales predictably

Google's answer to GPT: unify everything

GPT-3: The Model That Changed Everything

The Scale

| Metric	Value
Parameters	175 billion
Layers	96
Hidden size	12,288
Attention heads	96
Training tokens	300 billion
Training cost	$4.6 million

Few-Shot Learning
No gradient updates needed:

- 0-shot: Just describe task
- 1-shot: One example
- Few-shot: 2-10 examples
- **Works surprisingly well!**

In-Context Learning Example

```
1  Translate to French:
2  sea otter -> loutre de mer
3  cheese -> fromage
4  peppermint ->
5  (*@\textcolor{DarkGreen}{menthe poivr\'ee}@*)
```


In-Context Learning Performance by Model Size

Checkpoint: Few-Shot Learning
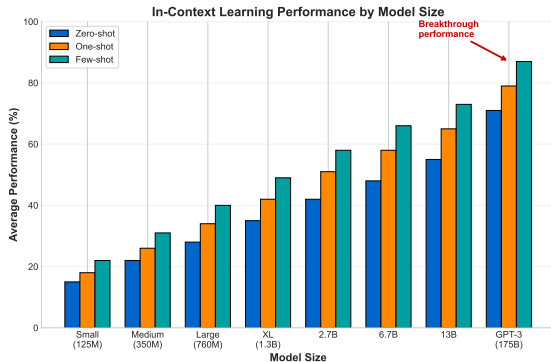
Quick Check: Understanding GPT-3

Question 1:
What makes GPT-3's few-shot learning special?

A) No gradient updates needed

B) Faster training time

C) Smaller model size

D) Better architecture

Answers:

Answer 1: A - No gradient updates

- Just provide examples in prompt
- Model infers pattern from context
- 0-shot, 1-shot, or few-shot
- No fine-tuning required!

Question 2:
What is in-context learning?

A) Learning from examples in prompt

B) Transfer learning technique

C) New training algorithm

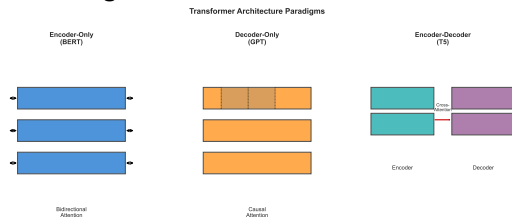D) Data augmentation method

Answer 2: A - Learning from examples

- Model sees pattern in prompt
- Applies to new instances
- Emergent ability at scale
- Revolutionized how we use LLMs

Key takeaway: GPT-3 showed that large enough models can learn tasks from just examples in the prompt, no training needed

Architecture Evolution: From BERT to GPT-3

Three Paradigms

Transformer Architecture Paradigms

Encoder-Only (BERT) — Bidirectional Attention

Decoder-Only (GPT) — Causal Attention

Encoder-Decoder (T5) — Encoder / Decoder / Cross-Attention

1. **Encoder-only** (BERT)
   - Bidirectional context
   - Best for understanding
   - Classification tasks
2. **Decoder-only** (GPT)
   - Autoregressive
   - Best for generation
   - Most scalable
3. **Encoder-Decoder** (T5)
   - Flexible input/output
   - Best for seq2seq
   - More parameters needed

The architecture debate is over: decoder-only won

Why Decoder-Only Won

Advantage	Reason
Simplicity	One stack vs two
Efficiency	Better GPU utilization
Scaling	More predictable
Generation	Natural fit
Training	Simpler objective

The Convergence

All roads lead to autoregressive:

- BERT team moves to decoder (PaLM)
- T5 team adopts decoder (Flan)
- Industry standardizes on GPT-style
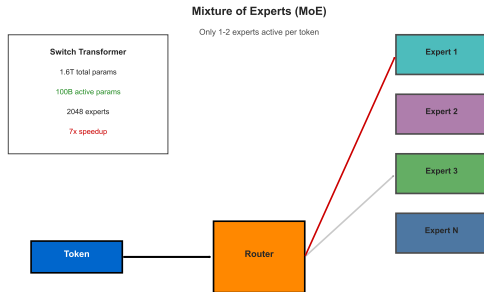- Even vision models follow (ViT-GPT)

The Problem with Dense Models
Every token uses ALL parameters:

- 175B params = 175B operations
- Linear scaling of compute
- Hit hardware limits quickly
- **Warning:** Unsustainable growth

Switch Transformer (2021)

Metric	Value
Total params	1.6 trillion
Active params	100B per token
Experts	2048
Speedup	7x

The MoE Solution

Mixture of Experts (MoE)

Only 1-2 experts active per token

Switch Transformer
1.6T total params
100B active params
2048 experts
7x speedup

Token — Router — Expert 1 / Expert 2 / Expert 3 / Expert N

How It Works

1. Router selects experts
2. Each token $\rightarrow$ 1-2 experts
3. Experts specialize automatically
4. Load balancing critical

1.6T params, 100B compute cost!

Checkpoint: Sparse Models

Quick Check: Understanding Mixture of Experts

Question 1:
How does Switch Transformer save compute?

A) Fewer total parameters
B) Only activate needed experts
C) Faster GPUs
D) Better optimization algorithm

Answers:

Answer 1: B - Activate only needed experts
- Router selects 1-2 experts per token
- 1.6T total params, 100B active
- 16x parameter efficiency
- Sparsity is the key!

Question 2:
What's the main advantage of MoE?

A) Scale parameters without cost
B) Easier to train
C) Better accuracy
D) Less memory needed

Answer 2: A - Scale without cost
- Add params without compute penalty
- Each expert specializes
- 7x speedup over dense
- Future of scaling

Key takeaway: Sparse models like Switch Transformer show we can have massive parameter counts with manageable compute costs through selective activation

Model Parallelism Types

1. **Data Parallel**
   - Split batch across GPUs
   - Replicate model
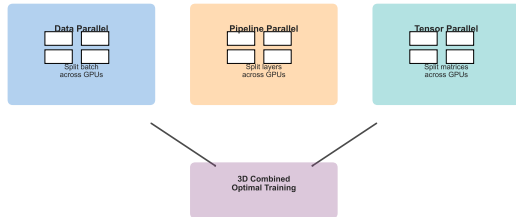   - Synchronize gradients

2. **Pipeline Parallel**
   - Split layers across GPUs
   - Micro-batching
   - Bubble overhead

3. **Tensor Parallel**
   - Split matrices across GPUs
   - High communication
   - Best for large layers

3D Parallelism: Combine all three!

3D Parallelism for Large Model Training

Data Parallel	Pipeline Parallel	Tensor Parallel
Split batch across GPUs	Split layers across GPUs	Split matrices across GPUs

3D Combined Optimal Training

GPT-3 Training Stats

Resource	Amount
GPUs	10,000 V100s
Training time	34 days
FLOPs	3.14×10^{23}
Power usage	1,287 MWh
CO2 emissions	552 tons

From Research to Production

The API Revolution
No more training needed:

- OpenAI API (GPT-3)
- Google Cloud (PaLM)
- Anthropic (Claude)
- Cohere, AI21, etc.

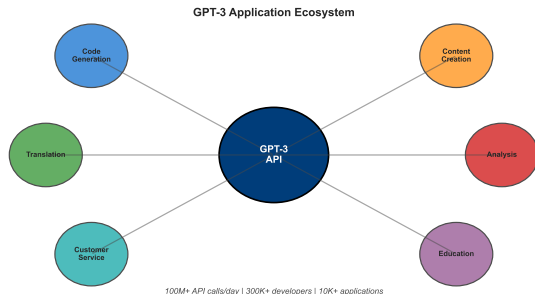**Prompt Engineering**
The new programming:

- Zero-shot prompts
- Few-shot examples
- Chain-of-thought
- Instruction following

Cost Per 1M Tokens

Model	Price
GPT-3 Ada	$0.40
GPT-3 Curie	$2.00
GPT-3 Davinci	$20.00

Real Applications (2021-2023)

GPT-3 Application Ecosystem

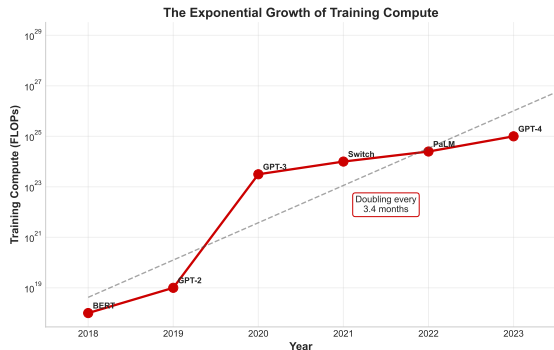100M+ API calls/day | 300K+ developers | 10K+ applications

Success Stories

- GitHub Copilot: 40% of code
- Jasper.ai: $125M revenue
- Copy.ai: 10M users
- ChatGPT: 100M in 2 months

The Compute Race: Power Laws and Politics

Compute Requirements Over Time

The Exponential Growth of Training Compute

Doubling every 3.4 months!

The Players (2023)

Company	Largest Model
OpenAI	GPT-4 (1T?)

The Hardware Arms Race

- NVIDIA A100: $10,000
- NVIDIA H100: $30,000
- TPU v4: Not for sale
- Custom chips emerging

National AI Strategies

- US: Export controls on chips
- China: $150B investment
- EU: Sovereign cloud initiative
- UK: Safety focus

Compute is the new oil

Known Limitations

1. **Hallucinations**
   - Confident wrong answers
   - Made-up citations
   - No uncertainty estimates

2. **Reasoning Failures**
   - Simple math errors
   - Logic puzzles fail
   - Common sense gaps

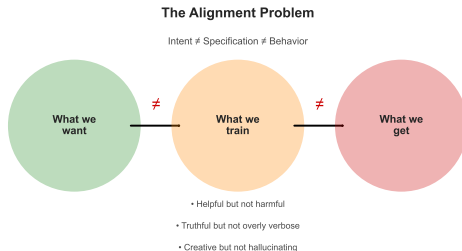3. **Control Problems**
   - Can't guarantee safety
   - Prompt injection attacks
   - Jailbreaking possible

Warning: Bigger models = bigger problems

The Cost Crisis

- Training GPT-4: $100M+
- Running costs: $700K/day
- Environmental impact huge
- Excludes most researchers

The Alignment Problem

The Alignment Problem

Intent ≠ Specification ≠ Behavior

What we want ≠ What we train ≠ What we get

- Helpful but not harmful
- Truthful but not overly verbose
- Creative but not hallucinating

The Road Ahead: What's Next?

Scaling Continues

GPT-5 and beyond:

- 10T parameters coming
- Multimodal by default
- Video understanding
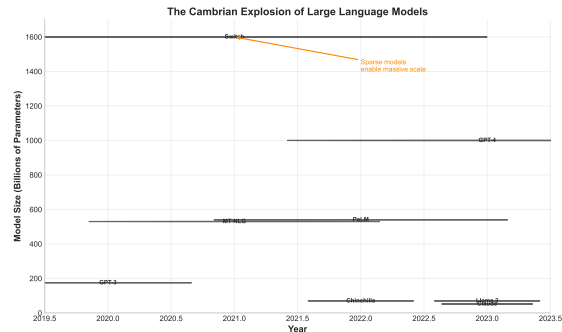- Reasoning breakthroughs?

Efficiency Revolution

Making models smaller:

- Quantization (1-bit models!)
- Knowledge distillation
- Efficient architectures
- On-device inference

New Paradigms

- Retrieval-augmented generation
- Tool use and plugins
- Constitutional AI

The Cambrian Explosion

The Cambrian Explosion of Large Language Models

Open Questions

1. Will scaling laws hold forever?
2. Can we solve hallucinations?
3. Is AGI possible this way?
4. Who controls the models?

What We Learned About Scale

Technical Insights
- Scale brings emergence
- Decoder-only won
- Sparsity enables scale
- In-context learning works
- Compute is everything

Practical Lessons
- APIs democratize AI
- Prompting is programming
- Few-shot often enough
- Fine-tuning less needed
- Costs dropping fast

Future Challenges
- Hallucination problem
- Alignment crucial
- Efficiency needed
- Access inequality
- Safety concerns real

The scaling revolution changed everything. We're still figuring out what that means.

Next week: How these models actually read text (Tokenization)

References

- Kaplan et al. (2020). "Scaling Laws for Neural Language Models"
- Brown et al. (2020). "Language Models are Few-Shot Learners" (GPT-3)
- Raffel et al. (2020). "Exploring the Limits of Transfer Learning with T5"
- Fedus et al. (2021). "Switch Transformers: Scaling to Trillion Parameter Models"
- Hoffmann et al. (2022). "Training Compute-Optimal Large Language Models" (Chinchilla)
- Wei et al. (2022). "Emergent Abilities of Large Language Models"
- Chowdhery et al. (2022). "PaLM: Scaling Language Modeling with Pathways"
- Anil et al. (2023). "PaLM 2 Technical Report"
- OpenAI (2023). "GPT-4 Technical Report"
- Anthropic (2023). "Constitutional AI: Harmlessness from AI Feedback"