Pre-trained Language Models

NLP Course 2025

September 30, 2025

A pedagogical journey from wasteful repetition to efficient transfer learning

45 slides • 4 sections • 2 checkpoints

What You Will Master

Conceptual Understanding

- Why pre-training revolutionized NLP
- Transfer learning principles
- BERT vs GPT paradigms
- Fine-tuning strategies
- Model selection criteria

Practical Skills

- Use pre-trained models
- Fine-tune for your task
- Choose the right model
- Implement with HuggingFace
- Avoid common pitfalls

By the end: You'll understand how to leverage billions of parameters trained on terabytes of text

Pre-training changed everything - from months to minutes, from millions to thousands

Section 1

The Challenge

Reinventing the Wheel

The Reading Revolution: An Analogy

Imagine two medical students:

Student A: Starting from Zero

- Never learned to read
- Must learn alphabet first
- Then words, grammar, sentences
- Finally can read medical texts
- Takes 10 years to become doctor

Warning: 90% of time learning to read, only 10% on medicine

Student B: Transfer Learning

- Already knows how to read
- Starts directly with medical texts
- Focuses only on medical knowledge
- Becomes doctor in 4 years
- Better outcomes, less time

100% of time on medicine, reading skill transfers

Intuition: Pre-training is like teaching AI to read before teaching it specific tasks!

This analogy captures why pre-training changed everything in NLP

The $10 Million Problem No One Talked About

The wasteful reality of 2017-2018:

The Old Way:

- Company A: Sentiment analysis
  - Cost: $500,000
  - Time: 2 weeks on 64 GPUs
  - Learns: Grammar from scratch
- Company B: Translation
  - Cost: $500,000
  - Time: 2 weeks on 64 GPUs
  - Learns: Grammar from scratch
- Company C: Q&A
  - Cost: $500,000
  - Time: 2 weeks on 64 GPUs
  - Learns: Grammar from scratch

Total waste: $1.5 million!

The Smart Way (Today):

- Pre-train ONCE: $1 million
- Fine-tune for each task:
  - Sentiment: $10,000
  - Translation: $10,000
  - Q&A: $10,000

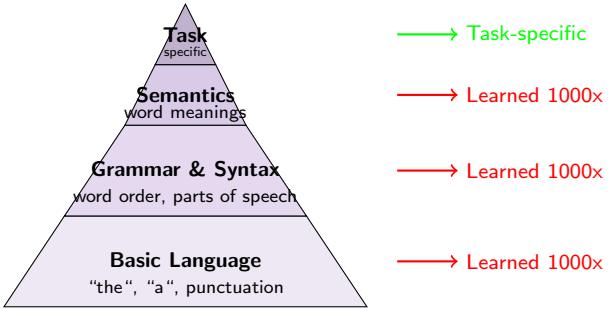
Total cost: $1.03 million
Savings: $470,000!

Plus: Better performance!

Every team was re-learning what "the" means from scratch!

What Every Model Was Learning from Scratch

The redundant learning pyramid:

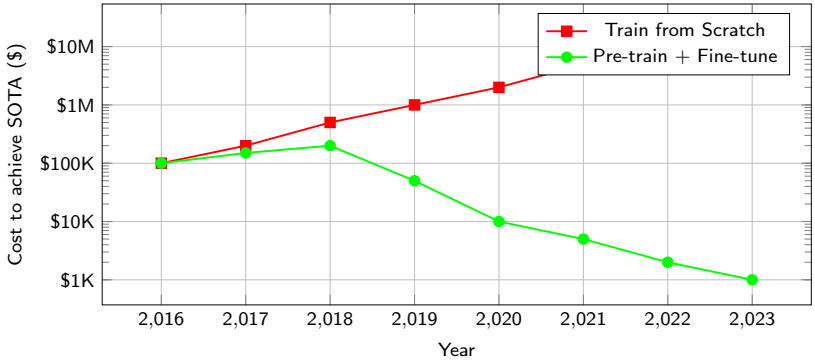


Common Misconception: "Each task needs its own model from scratch" - False! 95% of language understanding is general

The bottom 95% of the pyramid is identical for all NLP tasks

The Computational Reality

Training costs over time:



Pre-training inverted the cost curve - better models got cheaper!

The divergence started with BERT in 2018

The Carbon Footprint Reality

Environmental cost of training:

Training from Scratch (per model):

- Energy: 27,000 kWh
- CO2: 12 metric tons
- Equivalent to:
  - 5 cars for 1 year
  - 125 flights NYC-SF
  - Average home for 2.5 years

If 1000 companies do this:

- 12,000 metric tons CO2
- Small city's annual emissions

Pre-training + Fine-tuning:

- Pre-train once: 12 metric tons
- Fine-tune: 0.01 metric tons
- 1000 companies: 22 metric tons total

545x reduction in carbon footprint!

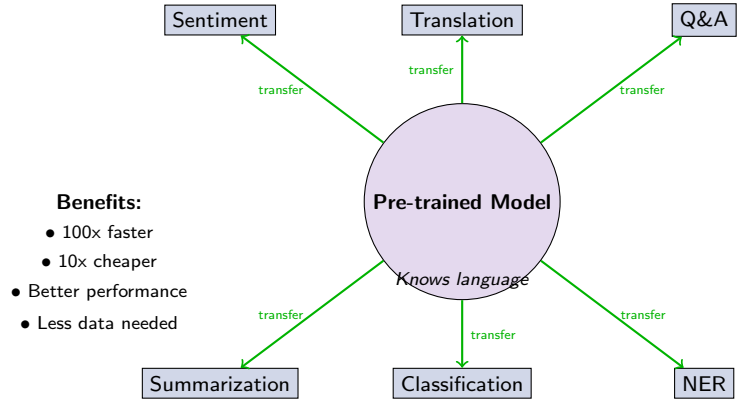
[Carbon footprint visualization]

Real World: Google's BERT pre-training saved an estimated 50,000 metric tons of CO2

Environmental responsibility drove adoption as much as cost savings

The Transfer Learning Opportunity

What if we could share knowledge?



Benefits:

- 100x faster
- 10x cheaper
- Better performance
- Less data needed

One model's knowledge can bootstrap infinite downstream tasks

Section 2

The Foundation

Learning from Everything

What is Pre-training?

Teaching models to understand language itself:

Traditional Supervised Learning:

- Need: Labeled data
- Example: “Great movie¡‘ → Positive
- Scale: Thousands of examples
- Cost: $100K for annotation
- Coverage: One specific task

Warning: Limited by human annotation speed

Pre-training (Self-Supervised):

- Need: Raw text only
- Example: “The [?] sat on the mat“
- Scale: Billions of examples
- Cost: Just compute time
- Coverage: All of language

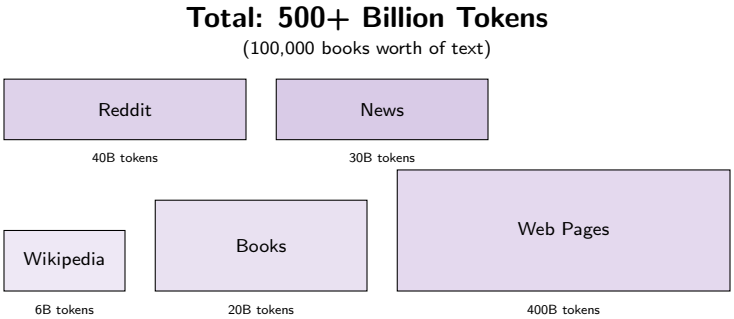
Unlimited data from the internet

Intuition: Pre-training turns the internet into a free teacher - every sentence becomes a learning example

Self-supervision was the key unlock - no humans needed

The Scale of Pre-training Data

Where does the knowledge come from?



Models read more in training than a human could in 10,000 lifetimes

GPT-3 trained on 45TB of compressed text

Self-Supervised Learning: The Secret Sauce

How to create infinite training examples:

Method 1: Masked Language (BERT)

1. Take: "The cat sat on the mat"
2. Mask: "The [MASK] sat on the mat"
3. Predict: "cat"
4. Learn: Context → Word

Advantages:

- Bidirectional context
- Natural for understanding
- Good for classification

Method 2: Next Word (GPT)

1. Take: "The cat sat on the"
2. Predict: "mat"
3. Learn: Sequence → Next
4. Continue: Generate text

Advantages:

- Natural generation
- Autoregressive
- Good for creation

Both methods turn unlabeled text into labeled training data automatically

This self-supervision eliminates the annotation bottleneck

Transfer Learning: The Theory

Why does knowledge transfer work?

[Transfer learning landscape visualization]

Key insights:

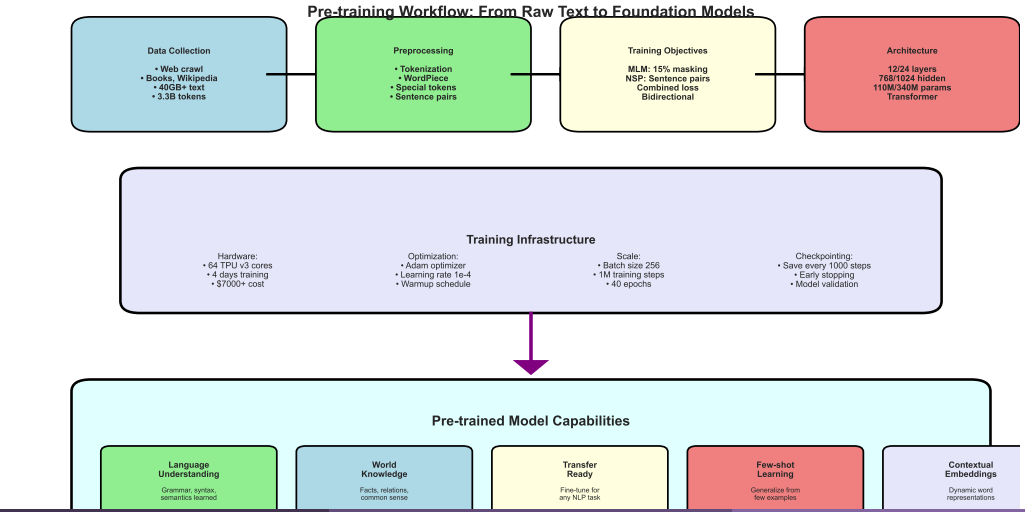
- **Lower layers**: Learn general features (word meanings, grammar)
- **Middle layers**: Learn combinations (phrases, concepts)
- **Upper layers**: Learn task-specific patterns
- **Transfer**: Reuse lower/middle, adapt upper

Intuition: Like learning piano helps with typing - the finger coordination transfers

Neural networks naturally learn hierarchical representations

The Pre-training Process Workflow

From raw text to capable model:



Layers of Understanding: What Models Learn

Knowledge emerges at different depths:

Early Layers (1-4):

- Character patterns
- Word boundaries
- Basic syntax
- Parts of speech

Example: "run" vs "runs"

Middle Layers (5-8):

- Semantic meaning
- Word relationships
- Phrase structure
- Dependencies

Example: "bank" (river/money)

Deep Layers (9-12):

- Abstract concepts
- Long-range patterns
- Task-specific features
- Reasoning steps

Example: Inference, causality

Each layer builds on the previous - from letters to logic

Probing studies reveal this consistent hierarchy across models

Test Your Understanding

Quick Quiz:

Q1: What's the main advantage of pre-training?

A) Faster inference
B) Transfer learning to any task
C) Smaller models
D) Less memory usage

Q2: How does self-supervision work?

A) Humans label the data
B) Models label each other
C) Create labels from the text itself
D) Random labeling

Q3: Why is pre-training expensive but fine-tuning cheap?

Answers:

A1: B - Transfer learning to any task

- Pre-trained knowledge applies everywhere
- One model bootstraps many applications
- Eliminates starting from scratch

A2: C - Create labels from the text itself

- Mask words $\rightarrow$ predict them
- Or predict next word
- No human annotation needed

A3: D - Both B and C

- General knowledge learned once
- Task-specific needs little data

Section 3

The Architecture

Two Paradigms: BERT and GPT

BERT: Bidirectional Encoder Representations

The fill-in-the-blank master:

Core Idea:

- Look at context from both sides
- Mask random words (15%)
- Predict what's hidden
- Learn deep bidirectional representations

[BERT masking visualization]

Example:
Input: "The [MASK] sat on the [MASK]"
Output: "cat", "mat" (high probability)

Architecture:

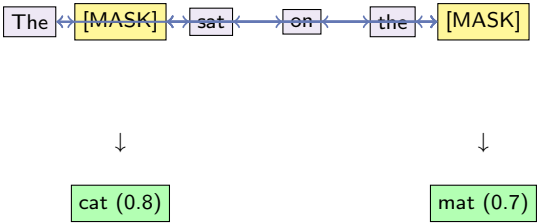
- Transformer encoder only
- 12/24 layers (Base/Large)
- 110M/340M parameters
- Attention in all directions

Intuition: BERT is like a detective examining all clues simultaneously to solve the mystery

BERT = Bidirectional Encoder Representations from Transformers

Masked Language Modeling (MLM) Explained

How BERT learns from context:



Training details:

- 80% of masks: Replace with [MASK]
- 10% of masks: Replace with random word
- 10% of masks: Keep original (but still predict)

This prevents the model from only looking for [MASK] tokens

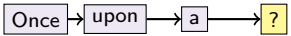
The 80-10-10 rule prevents overfitting to the [MASK] symbol

GPT: Generative Pre-trained Transformer

The story continuation expert:

Core Idea:

- Predict the next word
- Use only left context
- Autoregressive generation
- Learn to continue any text



time (0.85)
day (0.10)
night (0.05)

Architecture:

- Transformer decoder only
- 12/24/48 layers (GPT/GPT-2/GPT-3)
- 117M/1.5B/175B parameters
- Causal (left-to-right) attention

Example:
Input: “Once upon a“
Output: “time“ (highest probability)

Intuition: GPT is like an author who can only see what’s been written so far

GPT = Generative Pre-trained Transformer

Causal Language Modeling Explained

How GPT learns to generate:

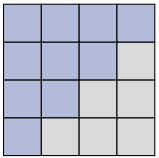
Training Process:

1. Start with text sequence
2. Predict each word from previous
3. Compare with actual next word
4. Update to improve predictions

Generation Process:

1. Start with prompt
2. Generate next token
3. Add to sequence
4. Repeat until done

Attention Mask:



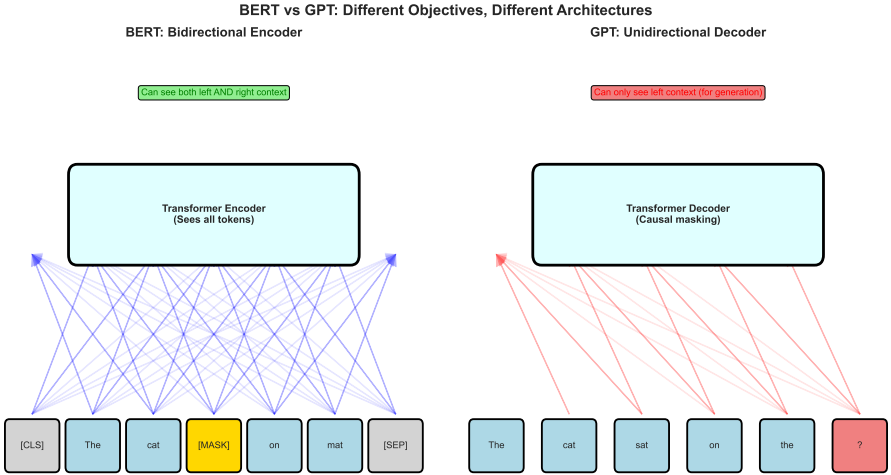
Can attend
Cannot attend (future)

Example Generation:

- Prompt: "The weather"
- Step 1: "The weather **is**"
- Step 2: "The weather is **nice**"
- Step 3: "The weather is nice **today**"

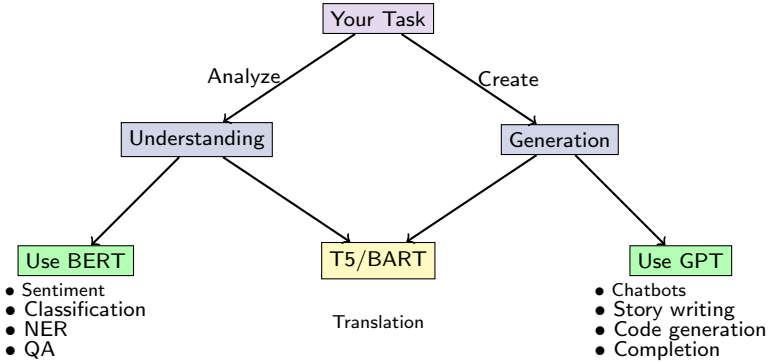
BERT vs GPT: Head-to-Head Comparison

Two paradigms, different strengths:



Choosing the Right Approach

Decision tree for model selection:



Real World: Google uses BERT for search understanding, OpenAI uses GPT for ChatGPT

Checkpoint 2: BERT vs GPT Understanding

Test Your Architecture Knowledge

Architecture Quiz:

Q1: Why can't BERT generate text well?

A) Too small
B) Bidirectional training
C) Wrong architecture
D) Not enough data

Q2: What makes GPT good at generation?

A) Larger size
B) Autoregressive training
C) Better data
D) Faster inference

Q3: For sentiment analysis, which is better?

Answers:

A1: B - Bidirectional training

- BERT sees future context during training
- Can't generate sequentially
- Designed for understanding, not creation

A2: B - Autoregressive training

- Learns to predict next token
- Natural for sequential generation
- Each output becomes next input

A3: B - BERT understands full context

- Sees entire sentence at once
- Better for classification tasks

Section 4

The Revolution

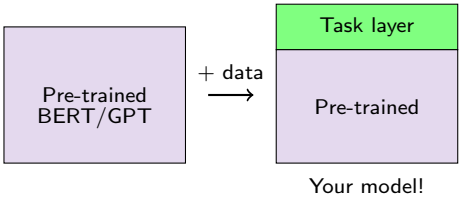
From Pre-training to Your Task

Fine-tuning: Teaching New Tricks

Adapting general knowledge to specific tasks:

The Process:

1. Start with pre-trained model
2. Add task-specific head
3. Train on your labeled data
4. Much less data needed (100s-1000s)
5. 10-100x faster than from scratch



What changes:

- Top layers adapt most
- Lower layers change little
- Task head learns mapping

Example - Sentiment Analysis:

- Pre-trained BERT: Knows language
- Your data: 1000 movie reviews
- Fine-tune: 1 hour on 1 GPU
- Result: 95% accuracy

The Model Zoo: Choose Your Fighter!

Pre-trained models available today:

Model	Size	Type	Best For	Company
BERT-base	110M	Encoder	Classification	Google
BERT-large	340M	Encoder	Understanding	Google
RoBERTa	355M	Encoder	Robust tasks	Facebook
GPT-2	1.5B	Decoder	Generation	OpenAI
GPT-3	175B	Decoder	Few-shot	OpenAI
GPT-4	1T	Decoder	Everything	OpenAI
T5	11B	Enc-Dec	Text-to-text	Google
BART	400M	Enc-Dec	Summarization	Facebook
mT5	13B	Enc-Dec	Multilingual	Google

Real World: HuggingFace hosts 100,000+ pre-trained models ready to use

The model zoo grows daily - new models appear weekly

Using Pre-trained Models in Practice

5 lines of code to state-of-the-art:

```
from transformers import pipeline

# Load pre-trained model
classifier = pipeline(``sentiment-analysis``)

# Use immediately
result = classifier(``I love this movie!``)
# Output: [{'label': 'POSITIVE', 'score': 0.999}]

# Or fine-tune on your data
from transformers import AutoModelForSequenceClassification, Trainer

model = AutoModelForSequenceClassification.from_pretrained(``bert-base``)
trainer = Trainer(model=model, train_dataset=your_data)
trainer.train()  # Fine-tunes in hours, not weeks
```

Modern libraries make pre-trained models as easy as importing a package

HuggingFace Transformers is the standard library for pre-trained models

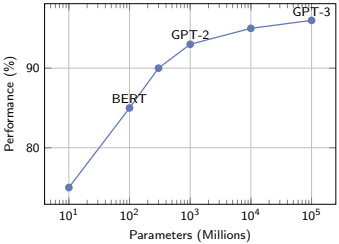
Which Model Should You Use?

Practical decision guide:

Start with these questions:

1. What's your task?
   - Classification → BERT/RoBERTa
   - Generation → GPT-2/GPT-3
   - Translation → T5/mT5
2. How much data do you have?
   - <100 examples → GPT-3 few-shot
   - 100-1000 → Fine-tune small model
   - >1000 → Fine-tune large model
3. What's your budget?
   - Free → BERT-base
   - $100s → GPT-2
   - $1000s → GPT-3 API

Model size vs performance:

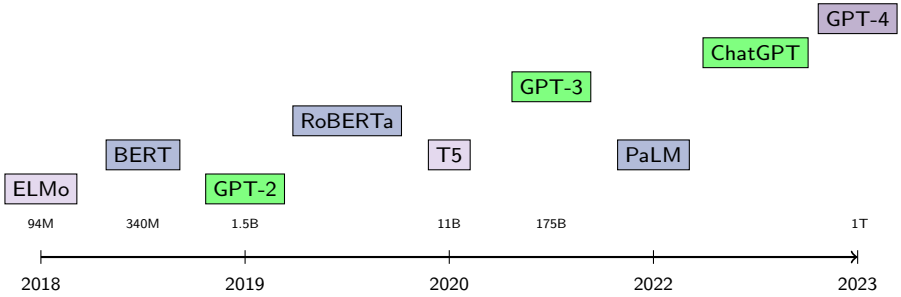


Diminishing returns:
10x size → 2-3% improvement

Common Misconception: "Bigger is always better" - False! BERT-base beats GPT-3 on many specific tasks

The Evolution: From ELMo to GPT-4

How we got here:



Key milestones:

- **2018**: Pre-training idea proven (ELMo, BERT)
- **2019**: Scale begins (GPT-2 “too dangerous to release“)
- **2020**: Few-shot learning (GPT-3)
- **2022**: Instruction following (ChatGPT)

Real-world Applications and Impact

Pre-trained models in production:

Search & QA:
- Google Search (BERT)
- Bing (GPT-4)
- Stack Overflow
- Customer support

10B+ queries daily

Content Creation:
- GitHub Copilot
- Copy.ai
- Jasper
- ChatGPT

$1B+ market

Understanding:
- Gmail Smart Reply
- Grammarly
- Translation
- Medical diagnosis

1B+ users

Impact: Every text you read online has likely been processed by a pre-trained model

Pre-trained models are the foundation of modern AI applications

The Future: What's Next?

Where pre-training is heading:

Current Trends:

- **Multimodal**: Text + Images + Audio
- **Longer context**: 100K+ tokens
- **Efficiency**: Same performance, 10x smaller
- **Specialization**: Domain-specific models
- **Personalization**: Models that know you

Challenges:

- Computational cost
- Data quality
- Hallucination
- Bias and fairness

Emerging Capabilities:

- Reasoning and planning
- Tool use and APIs
- Self-correction
- Continuous learning
- Uncertainty awareness

Next 5 Years:

- Personal AI assistants
- Scientific discovery
- Creative partners
- Educational tutors
- Medical advisors

The Pre-training Revolution

1. **The Problem:** Everyone was learning language from scratch
2. **The Solution:** Pre-train once, fine-tune many times
3. **The Methods:**
   - BERT: Bidirectional understanding (fill-in-blanks)
   - GPT: Autoregressive generation (predict next)
4. **The Impact:**
   - 100x faster development
   - 10x cost reduction
   - Better performance
5. **The Practice:** Use HuggingFace, start small, scale if needed
6. **The Future:** Multimodal, efficient, personalized

Summary: From Zero to Hero

Your journey today:

The Challenge:

- Wasteful repetition
- $10M problem
- Environmental impact
- Starting from scratch

The Architecture:

- BERT for understanding
- GPT for generation
- Choose based on task
- Both have strengths

The Foundation:

- Self-supervised learning
- Transfer learning theory
- Learning from internet
- Hierarchical knowledge

The Revolution:

- Fine-tuning magic
- Model zoo available
- 5 lines to SOTA
- Changed everything

Thank You!

Questions?

Next: Lab Session - Fine-tuning BERT

We'll fine-tune BERT for sentiment analysis in 30 minutes