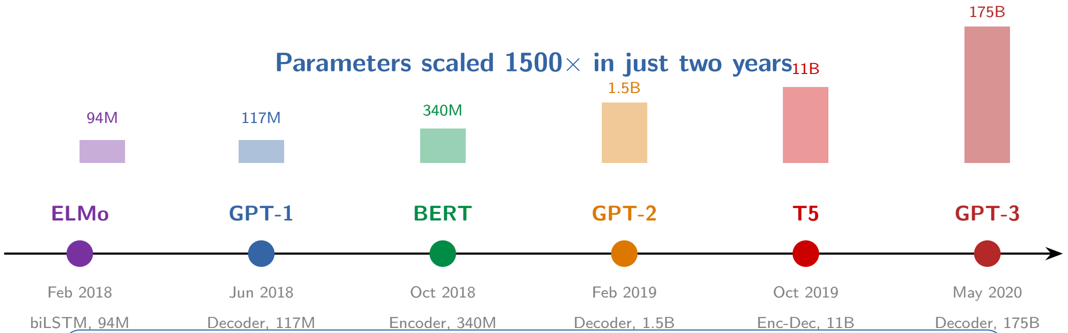
Early Notable Models

ELMo · GPT · BERT · GPT-2 · T5 · GPT-3

The timeline: 2018–2020



Each model introduced a key idea that shaped the field:
contextual embeddings → pre-train + fine-tune → bidirectional → zero-shot → text-to-text → few-shot

ELMo — contextual embeddings (2018)

Before ELMo: Word2Vec / GloVe give each word a **single fixed vector**
"bank" has the same embedding in "river bank" and "bank account"

Architecture:

Character CNN $\rightarrow$
2-layer biLSTM (4096 units)

Forward: predict next token
Backward: predict prev token

Final: learned weighted
sum of all layers

Innovation:

Same word $\rightarrow$ **different vectors**
depending on context!

Lower layers: syntax
Higher layers: semantics

Trained on 1B words

"I went to the **bank** to deposit money" $\rightarrow$ finance vector
"I sat on the river **bank**" $\rightarrow$ geography vector

The *same* word gets *different* representations — this was revolutionary

Impact: proved that pre-trained contextual representations transfer to downstream tasks.
Directly inspired GPT and BERT. But ELMo used LSTMs — transformers would do it better.

GPT-1 — the pre-train + fine-tune paradigm (2018)

“Improving Language Understanding by Generative Pre-Training” (OpenAI)

Stage 1: Pre-train

Causal LM on BooksCorpus (800M words, 7000 books)

Predict next token (unsupervised)

$\longrightarrow$

Stage 2: Fine-tune

Add task-specific head on top of pre-trained model

Supervised on labeled data (minimal changes)

Type	**Layers**	**Heads**	d_{model}	**Params**	**Context**	**Tokenizer**
Decoder-only	12	12	768	117M	512	BPE (40k)

Results:

SOTA on 9/12 tasks

+8.9% commonsense reasoning

+5.8% NLI

Beat specialized models

Key insight:

Generative pre-training on raw text transfers to diverse downstream tasks

Birth of “pre-train + fine-tune”

GELU activation · Learned position embeddings · Training: 100 epochs on BooksCorpus

BERT — bidirectional is better (2018)

GPT's limitation: decoder-only = unidirectional (left-to-right only).
"The _____ sat on the mat" — GPT can't use right context to fill the blank!

BERT's solution: Masked Language Modeling (bidirectional)

GPT (unidirectional):
The [?] sat on the mat
← only sees "The"

$\longrightarrow$

BERT (bidirectional):
The [MASK] sat on the mat
← sees "The" AND
"sat on the mat"

	Layers	**Hidden**	**Heads**	**Params**	**Vocab**
BERT-Base	12	768	12	110M	30,522 (WordPiece)
BERT-Large	24	1024	16	340M	30,522 (WordPiece)

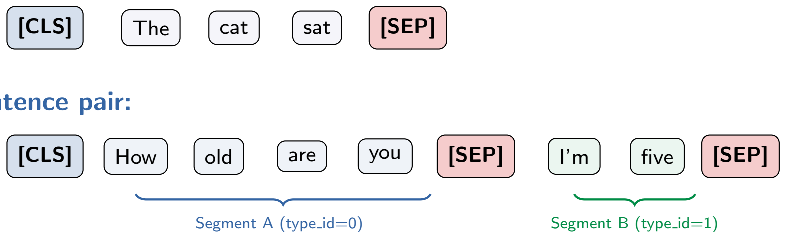
Trained on BooksCorpus (800M words) + English Wikipedia (2.5B words) = 3.3B words total

Encoder-only: no autoregressive generation. Designed for **understanding** tasks.

BERT — input format and special tokens

Single sentence:

Sentence pair:



Input = Token Embedding + Segment Embedding + Position Embedding

[CLS]

Classification token.
Its final hidden state is the sequence representation
for classification tasks.

[SEP]

Separator token.
Marks boundaries between sentences. Used with
segment embeddings.

[MASK]

Masking token.
Replaces 15% of tokens during MLM pre-training.
Never seen at
fine-tune time.

Tokenizer: WordPiece (30,522 vocab) · Subword: “playing” → “play” + “##ing” · Max length: 512

BERT — pre-training objectives

Masked LM (MLM)

Randomly mask
15% of tokens

Model predicts original token
using bidirectional context

Of the 15% selected:
80% → [MASK]
10% → random token
10% → unchanged

"The cat [MASK] on the mat"
→ predict "sat"

Next Sentence Prediction

Given sentence A and B,
predict: is B the actual next
sentence after A?

50% real pairs (IsNext)
50% random pairs (NotNext)

Binary classification on [CLS]

[CLS] A [SEP] B [SEP]
→ IsNext / NotNext

Later found: NSP was too easy (topic detection, not coherence).
RoBERTa (2019) dropped NSP entirely and got **better** results.
ALBERT replaced it with Sentence Order Prediction (SOP).

Training: 4 Cloud TPUs (16 chips), 4 days · Estimated cost: ~$500 (2018 prices!)

BERT — why everyone lost their minds

BERT set SOTA on 11 NLP tasks simultaneously (October 2018)

GLUE benchmark:
Previous SOTA: 72.8
BERT-Large: **80.5**
+7.7 points absolute improvement

SQuAD v1.1 (QA):
Previous SOTA: 91.7 F1
BERT-Large: **93.2 F1**
First to surpass human-level (91.2)

MultiNLI:
Previous SOTA: 82.1
BERT: **86.7**

CoNLL NER:
Previous SOTA: 92.2 F1
BERT: **92.8 F1**

The BERT effect: a single pre-trained model, fine-tuned with just a linear head, beat *years* of task-specific engineering on virtually every NLU benchmark.

BERT became the default starting point for NLP from late 2018 through 2020

Spawned: RoBERTa, ALBERT, DistilBERT, DeBERTa, SpanBERT, SciBERT, BioBERT, . . .

Three architectures, three strengths

Encoder-only

Bidirectional attention

Best for: understanding

NLI, NER, QA, classification

BERT, RoBERTa

Decoder-only

Causal (L$\rightarrow$R) attention

Best for: generation

Text, code, chat, reasoning

GPT, LLaMA, Mistral

Encoder-Decoder

Bidirectional enc + causal dec

Best for: seq-to-seq

Translation, summarization

T5, BART, mBART

2024 landscape: decoder-only dominates (GPT-4, Claude, LLaMA, Mistral, Gemini).
With enough scale + prompting, decoder-only handles understanding tasks too.

BERT-style encoders still used for: embeddings, retrieval, NER, classification at small scale

GPT-2 — scaling up, zero-shot emerges (2019)

“Language Models are Unsupervised Multitask Learners” (OpenAI)

Variant	Params	Layers	Hidden
Small	124M	12	768
Medium	355M	24	1024
Large	762M	36	1280
	1.5B	**48**	**1600**

vs. GPT-1:

13× more parameters
Pre-LayerNorm (more stable)
Context: 1024 tokens (vs. 512)
WebText: 40 GB

Zero-shot breakthrough:

SOTA on 7/8 LM benchmarks
with **no fine-tuning at all**
Coherent multi-paragraph text
[illegible]

“Too dangerous to release”: OpenAI withheld the full model for 9 months.
Staged release: Small (Feb) → Medium
(May) → Large (Aug) → XL (Nov 2019)

First major debate about responsible AI release. Mostly a PR event in hindsight.

Key lesson: scale alone produces qualitatively
new capabilities. Model “still underfits WebText.”

T5 — everything is text-to-text (2019)

Unified framework: cast *every* NLP task as "text in, text out"
Same model · same loss · same hyperparameters · different task prefixes

Translation:
"translate English to German: Hello"
→ "Hallo"

Summarization:
"summarize: [long article]"
→ "[short summary]"

Classification:
"sst2 sentence: The movie was great"
→ "positive"

Question answering:
"question: What is the capital? context: . . . "
→ "Paris"

Variant	Params	Enc/Dec layers	Trained on
Small	60M	6 / 6	C4 (750 GB)
Base	220M	12 / 12	C4
Large	770M	24 / 24	C4

Encoder-decoder with span corruption objective · Introduced **C4 dataset** (750 GB cleaned web text)
Massive systematic study: compared architectures, objectives, data sizes, transfer approaches

T5 — the definitive transfer learning study

T5 paper systematically compared every design choice:

Architectures
Encoder-decoder
Decoder-only
Prefix LM
Winner: enc-dec

Objectives
Autoregressive (CLM)
BERT-style (MLM)
Span corruption
Prefix LM
Winner: span corruption

Data quality
Unfiltered web
Filtered web (C4)
Different domains
Winner: clean + large

Transfer
Fine-tune all params
Adapters
Multi-task
Gradual unfreezing
Winner: fine-tune all

Key findings:

1. Encoder-decoder > decoder-only for most tasks (at equal compute)
2. Span corruption (denoising) > standard MLM > autoregressive
3. Scaling model size + data size + training steps all help, roughly equally
4. Data quality matters more than quantity — C4's cleaning was crucial

GPT-3 — the leap to 175B (2020)

"Language Models are Few-Shot Learners" (OpenAI, NeurIPS 2020)

Layers	**Heads**	d_{model}	**Head dim**	**Context**	**Params**
96	96	12,288	128	2048	**175 billion**

100× larger than GPT-2, 1500× larger than GPT-1

Training data:

300B tokens total
Common Crawl (60%)
WebText2 (22%)
Books (16%) + Wikipedia (3%)

Training cost:

355 GPU-years
∼**$4.6 million**
10,000 V100 GPUs for weeks

8 model sizes spanning 3 orders of magnitude:

Ada (125M) → Babbage (1.3B) → Curie (6.7B) → **Davinci (175B)**

June 2020: OpenAI launched the GPT-3 API — first major "LLM as a service"

Text in, text out. No training needed. This began the era of API-based AI.

GPT-3 — in-context learning

No gradient updates. No fine-tuning. Just text in the prompt.

Zero-shot

Task description only.
No examples.

"Translate English to French: cheese →"

Relies on pre-training

One-shot

1 demonstration example.

"sea otter → loutre de mer
cheese →"

Learn format from 1 example

Few-shot

10–100 examples in the prompt.

"sea otter → loutre
peppermint → menthe
cheese →"

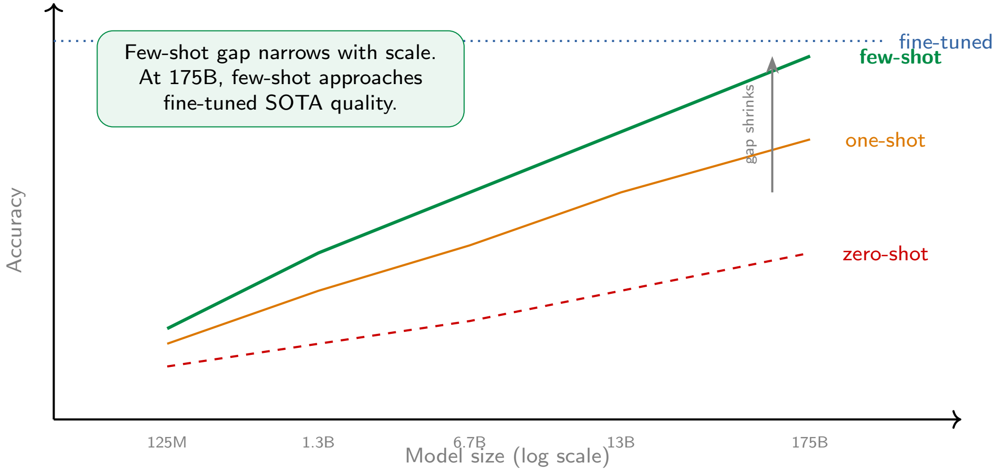
Near fine-tuned quality

Key finding: few-shot performance **scales much faster with model size** than zero-shot. Larger models are dramatically better at learning from context.

CoQA (QA): 85.0 F1 few-shot · Competitive with fine-tuned SOTA on many benchmarks

Translation, arithmetic, code generation, common sense — all from prompting alone

GPT-3 — few-shot scales with model size



The evolution: from fine-tuning to prompting

How we use language models evolved as they scaled

Fine-tune

Pre-train on raw text, then train on task data

GPT-1, BERT

→

Zero-shot

Just describe the task in text.

No examples.

GPT-2

→

Few-shot

Put examples in the prompt.

No training.

GPT-3

→

Chat / Instruct

SFT + RLHF
"Just ask in natural language"

ChatGPT, Claude

Trend: task-specific data requirements **decreased** as model scale **increased**

10k labeled examples → 100 examples → 10 examples → 0 examples (just instructions)

T5's text-to-text anticipated this: if every task is text-in/text-out, the interface naturally becomes a prompt. Scale made it work without training.

Data matters: the data scaling story

Bigger AND cleaner data consistently improved results

Model	**Dataset**	**Size**	**Key characteristic**
ELMo	1B Word Benchmark	1B words	Curated news text
GPT-1	BooksCorpus	800M words	7,000 unpublished books
BERT	Books + Wikipedia	3.3B words	Two curated sources
GPT-2	WebText	40 GB	Reddit-filtered web (3+ karma)
T5	C4	750 GB	Cleaned Common Crawl
GPT-3	Mixed (5 sources)	300B tokens	Weighted sampling strategy
LLaMA	Mixed (8 sources)	1.4T tokens	Publicly available data only
LLaMA 2	Unknown	2T tokens	Undisclosed mix

Quality > quantity:

T5 showed filtered C4 beats unfiltered Common Crawl. GPT-3 upsampled high-quality sources (Wikipedia 3×).

Diversity matters:

GPT-3 mixed: web, books, Wikipedia, code. Broad coverage → broader capabilities.

From 1B words (2018) to 2T tokens (2023) — a 2000× increase in training data

Summary: what each model taught us

Model	Year	Architecture	Params	Key contribution
ELMo	2018	biLSTM	94M	Contextual embeddings (same word → different vectors)
GPT-1	2018	Decoder	117M	Pre-train + fine-tune paradigm
BERT	2018	Encoder	340M	Bidirectional (MLM), SOTA on 11 tasks
GPT-2	2019	Decoder	1.5B	Zero-shot emerges from scale
T5	2019	Enc-Dec	11B	Text-to-text unification, systematic study
GPT-3	2020	Decoder	175B	Few-shot in-context learning, API launch

The themes that emerged:

1. Scale unlocks new capabilities (GPT-1 → 2 → 3)
2. Pre-training on raw text transfers to any task (GPT-1, BERT)
3. Architecture choice matters less than scale (T5 study)
4. Data quality is as important as quantity (T5, GPT-3)
5. Prompting can replace fine-tuning at sufficient scale (GPT-3)

Questions?

Next: Prompting — Zero-shot, Few-shot, Chain-of-Thought