

DOTE 6635: Artificial Intelligence for Business Research

# Pretraining

Renyu (Philip) Zhang

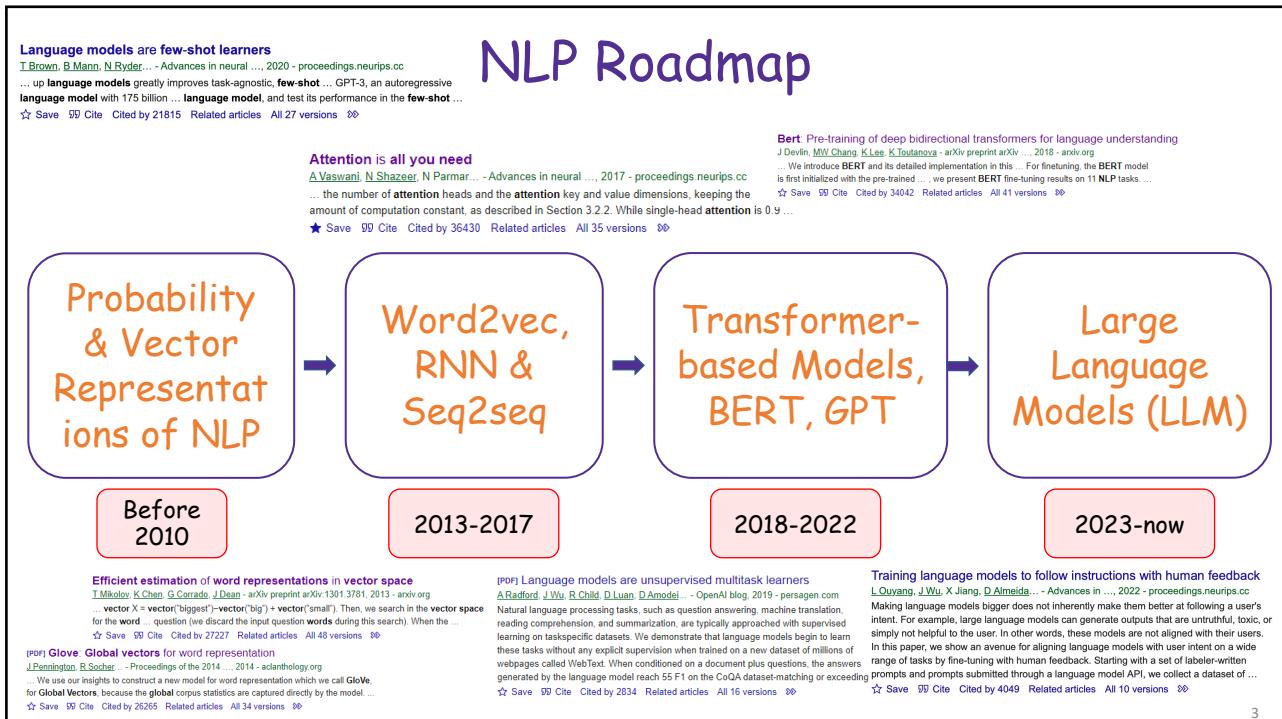
1

## Agenda

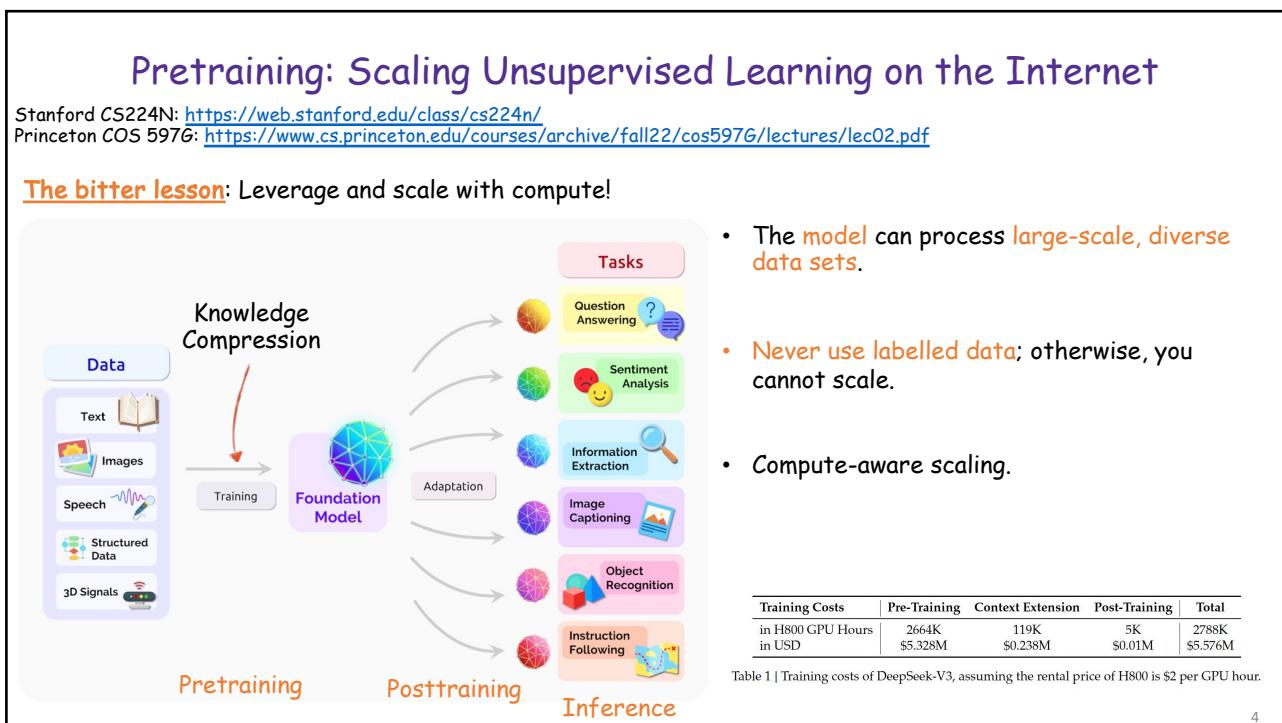
- BERT: Bidirectional Encoder Representations from Transformers
- GPT: Generative Pretrained Transformers

2

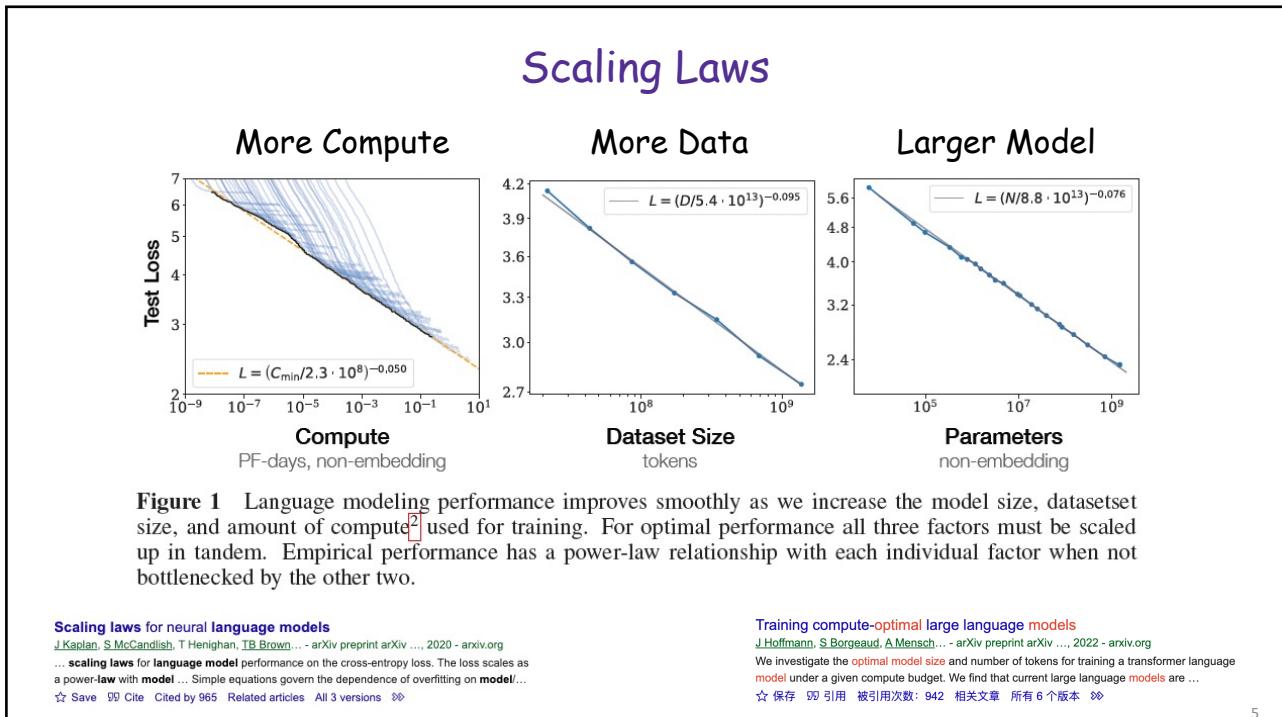
2



3



4



5

## Jason Wei's Typical Day

 **Jason Wei** ✅  
@\_jasonwei

My typical day as a Member of Technical Staff at OpenAI:

[9:00am] Wake up

[9:30am] Commute to Mission SF via Waymo. Grab avocado toast from Tartine

[9:45 am] Recite OpenAI charter. Pray to optimization Gods. Learn the Bitter Lesson

[10:00am] Meetings (Google Meet). Discuss how to train larger models on more data

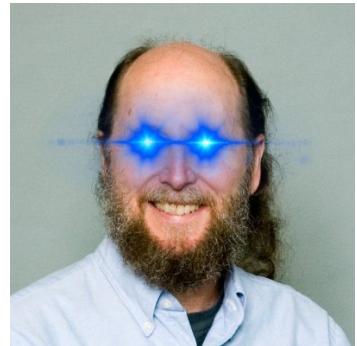
[11:00am] Write code to train larger models on more data. pair= @hwchung27

6

6

## The Bitter Lesson

- References: <http://www.incompleteideas.net/IncIdeas/BitterLesson.html>  
<https://www.youtube.com/watch?v=vbVfAqPI8ng>
- The biggest lesson that can be read from 70 years of AI research is that **general methods that leverage computation** are ultimately the most effective, and by a large margin.
- Leveraging domain knowledge (short-term & specific) vs. Leveraging computation (long-term & general).
- Bitter lesson: Leveraging domain knowledge is **self-satisfying** and **intellectually inspiring**, but plateaus in the long-run or even inhibits further progress.



Prof. Richard Sutton

7

7

## Jason Wei's Typical Day (Cont'd)

[12:00pm] Lunch at the canteen (vegan, gluten-free)  
[1:00pm] Actually train large models models on more data  
[2:00pm] Debug infra issues (why the fck did I pull from master?)  
[3:00pm] Babysit model training. Play with Sora  
[4:00pm] Prompt engineer aforementioned large models trained on more data  
[4:30pm] Short break, sit on avocado chair. Wonder how good Gemini Ultra actually is  
[5:00pm] Brainstorm potential algorithmic improvements for models  
[5:05pm] Conclude that algorithmic changes are too risky. Safer to just scale compute and data  
[6:00pm] Dinner. Clam chowder with Roon  
[7:00pm] Commute back home  
[8:00pm] Have a wine and get back to coding. Ballmer's peak is coming  
[9:00pm] Analyze experimental runs. I have a love/hate relationship with wandb  
[10:00pm] Launch experiments to run overnight and get results by tomorrow morning  
[1:00am] Experiments actually get launched  
[1:15am] Bedtime. Satya and Jensen watch from above. Compression is all you need. Good night

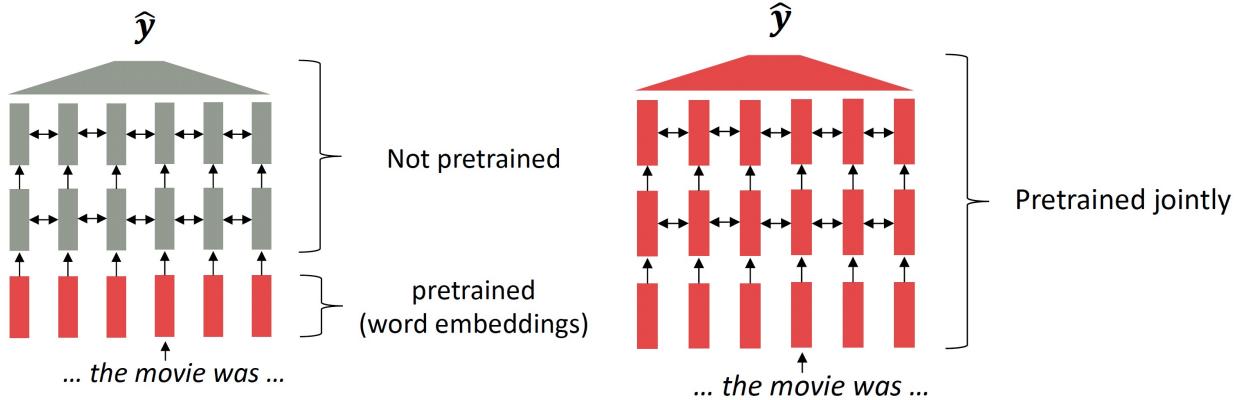
8

8

## From Pretrained Word Embeddings to Pretrained Models

Stanford CS224N: <https://web.stanford.edu/class/cs224n/>

Princeton COS 597G: <https://www.cs.princeton.edu/courses/archive/fall22/cos597G/lectures/lec02.pdf>



9

## From Pretrained Word Embeddings to Pretrained Models

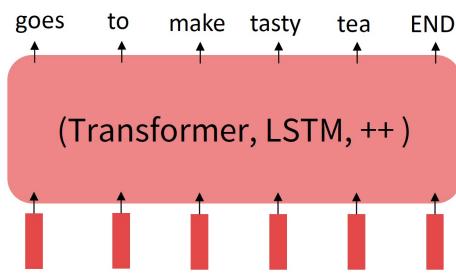
Stanford CS224N: <https://web.stanford.edu/class/cs224n/>

Princeton COS 597G: <https://www.cs.princeton.edu/courses/archive/fall22/cos597G/lectures/lec02.pdf>

- Pretraining can improve downstream NLP applications by serving as **parameter initialization**.

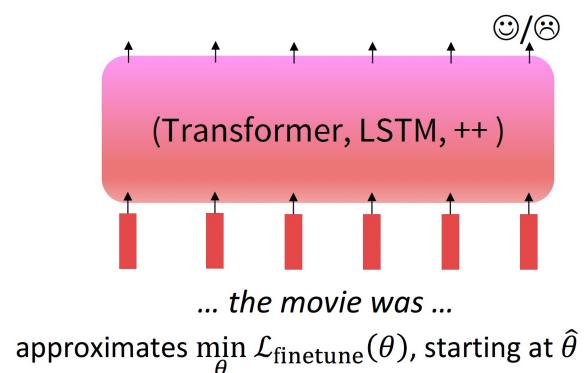
### Step 1: Pretrain (on language modeling)

Lots of text; learn general things!



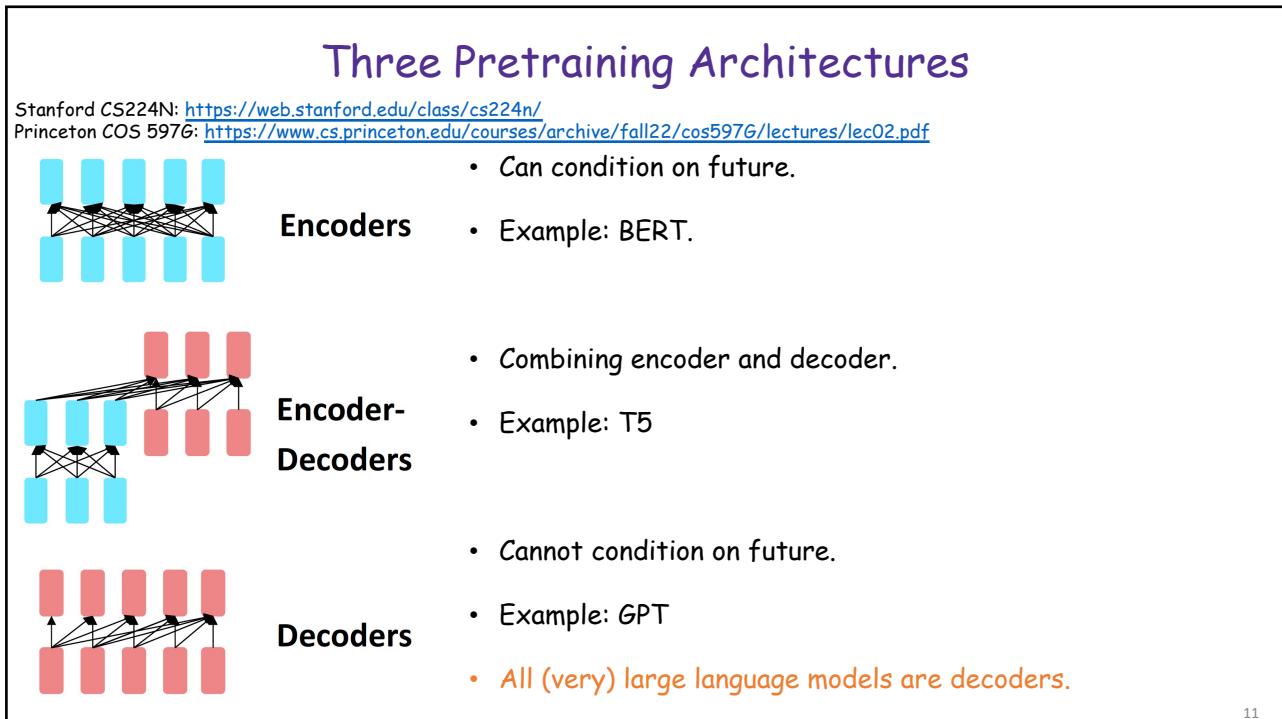
### Step 2: Finetune (on your task)

Not many labels; adapt to the task!



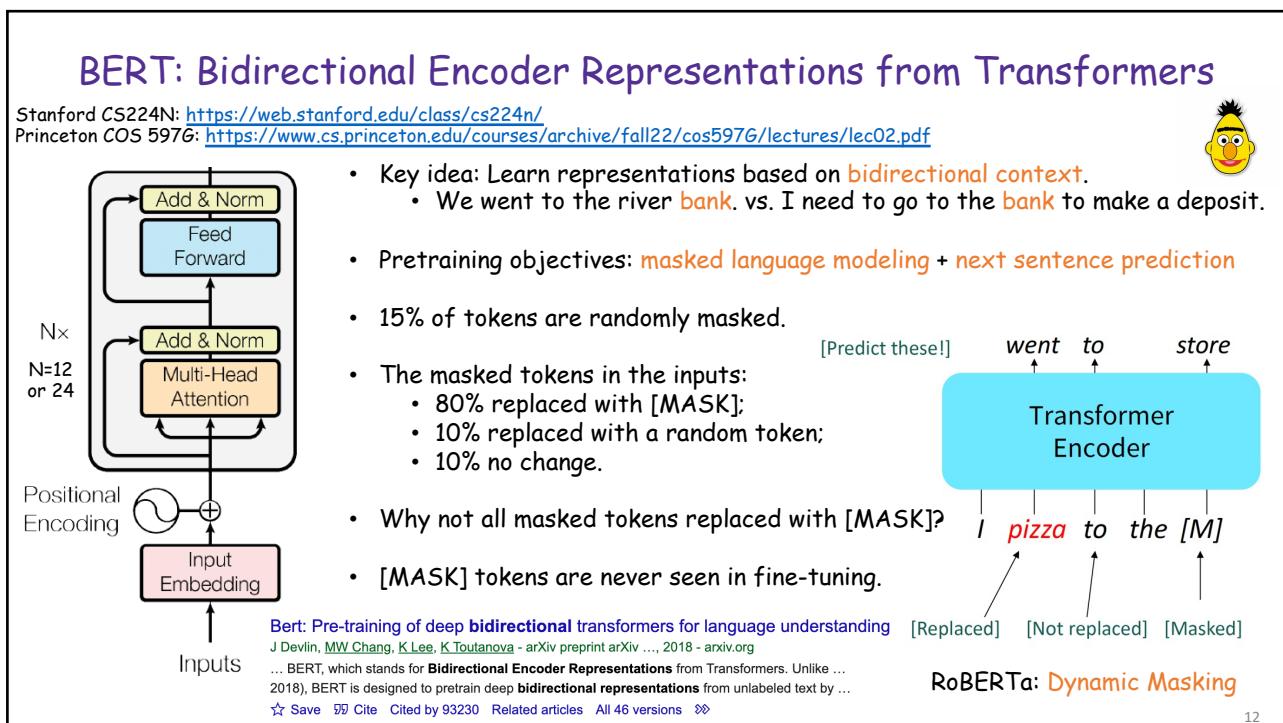
10

10



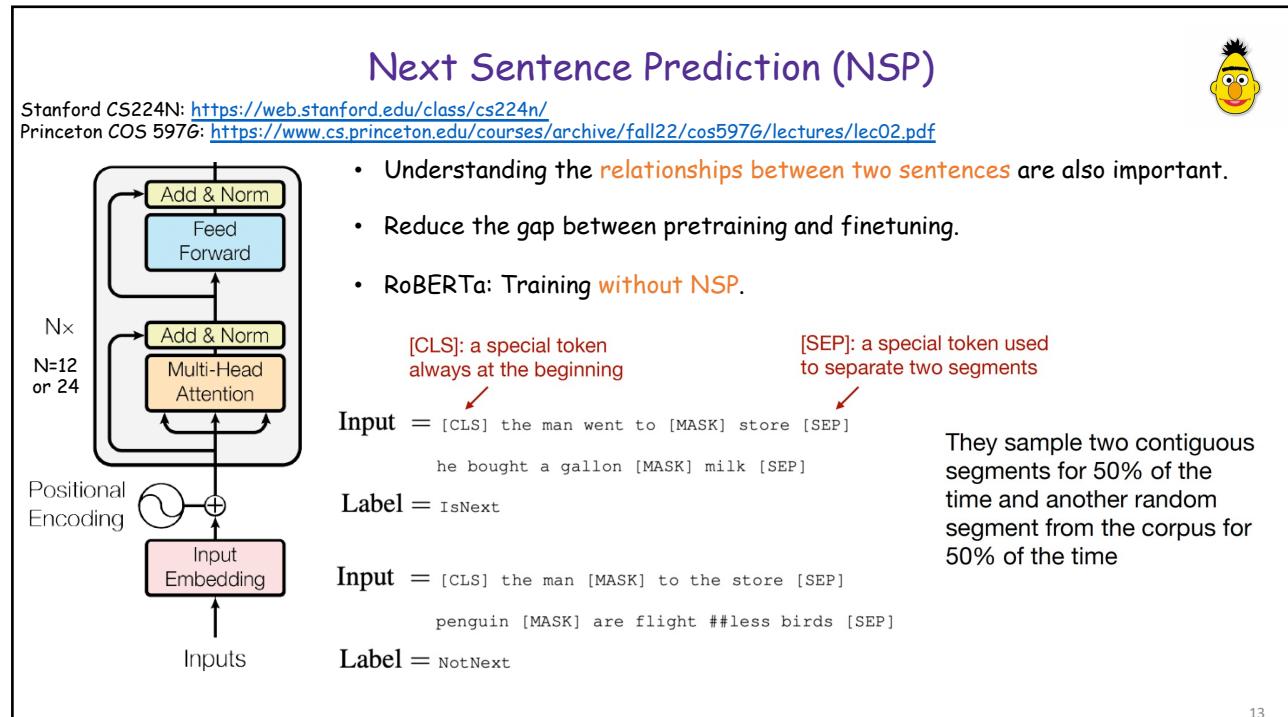
11

11



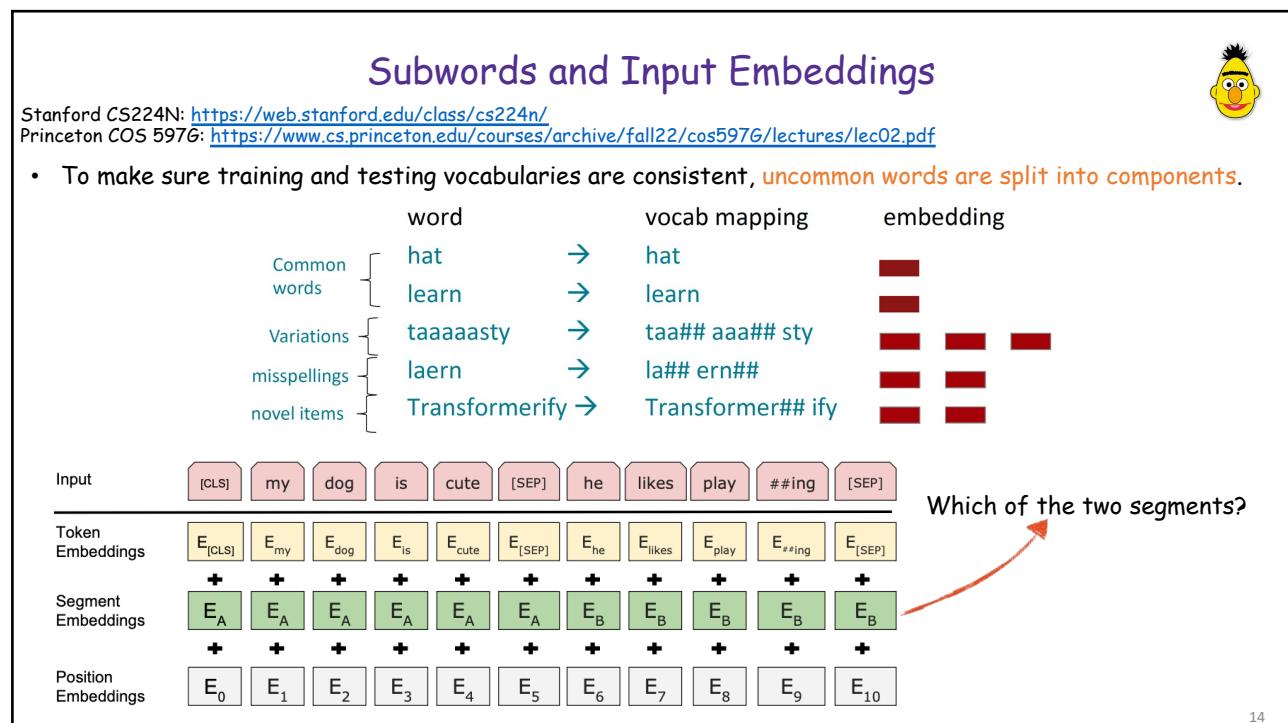
12

12



13

13

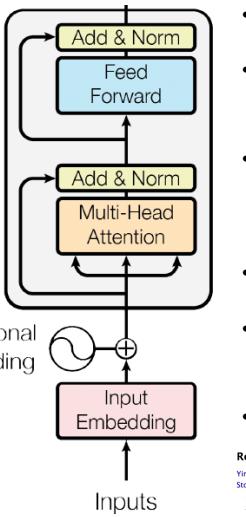


14

14

## BERT Pretraining: Putting Together

Stanford CS224N: <https://web.stanford.edu/class/cs224n/>  
Princeton COS 597G: <https://www.cs.princeton.edu/courses/archive/fall22/cos597G/lectures/lec02.pdf>



- BERT-base: 12 layers, 768-dim hidden size, 12 attention heads, 110M parameters
- BERT-large: 24 layers, 1024-dim hidden size, 16 attention heads, 340M parameters
- Trained on: Wikipedia (2.5B Tokens) + BookCorpus (0.8B Tokens)
  - RoBERTa: BookCorpus (16GB), CC-News (76GB), OpenWebText (38GB), Stories (31GB)
- Max sequence size: 512-word pieces (roughly 256 + 256 non-contiguous sequences)
- Trained for 1M steps, batch size = 256
  - Batch size increased to 8K for RoBERTa
- Pretrained with 64 TPUs for 4 days

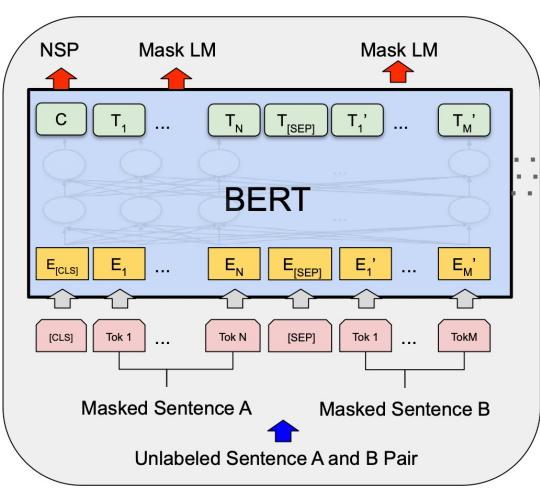
RoBERTa: A Robustly Optimized BERT Pretraining Approach  
Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov  
Language model pretraining has led to significant performance gains but careful comparison between different approaches is challenging. Training is computationally expensive, often done on private datasets of different sizes, and, as we will show, hyperparameter choices have significant impact on the final results. We present a reimplementation of BERT pretraining Ott et al., 2018, and carefully measure the impact of many hyperparameters. In particular, we find that the BERT word piece vocabulary can make a large difference in the performance of every model published after it. Our best model achieves state-of-the-art results on GLUE, RACE and SQuAD. These results highlight the importance of previously overlooked design choices, and raise questions about the source of recently reported improvements. We release our models and code.

15

15

## BERT Pretraining: Putting Together

Stanford CS224N: <https://web.stanford.edu/class/cs224n/>  
Princeton COS 597G: <https://www.cs.princeton.edu/courses/archive/fall22/cos597G/lectures/lec02.pdf>



- MLM and NSP are trained together.
- [CLS] is pretrained for NSP.
- The other token representations are pretrained for MLM

16

16

## Pretrain Once, Finetune Many Times



Stanford CS224N: <https://web.stanford.edu/class/cs224n/>  
Princeton COS 597G: <https://www.cs.princeton.edu/courses/archive/fall22/cos597G/lectures/lec02.pdf>

**Sentence-Level Task**

- Sentence pair classification tasks:

**MNLI**      Premise: A soccer game with multiple males playing.  
Hypothesis: Some men are playing a sport.      {entailment, contradiction, neutral}

**Multi-genre Natural Language Inference**: Predict the relationship between two sentences.

**QQP**      Q1: Where can I learn to invest in stocks?  
Q2: How can I learn more about stocks?      {duplicate, not duplicate}

**Quora Question Pairs**: Detect paraphrase questions.

- Single sentence classification tasks:

**SST2**      rich veins of funny stuff in this movie      {positive, negative}  
**Sentiment Analysis**

17

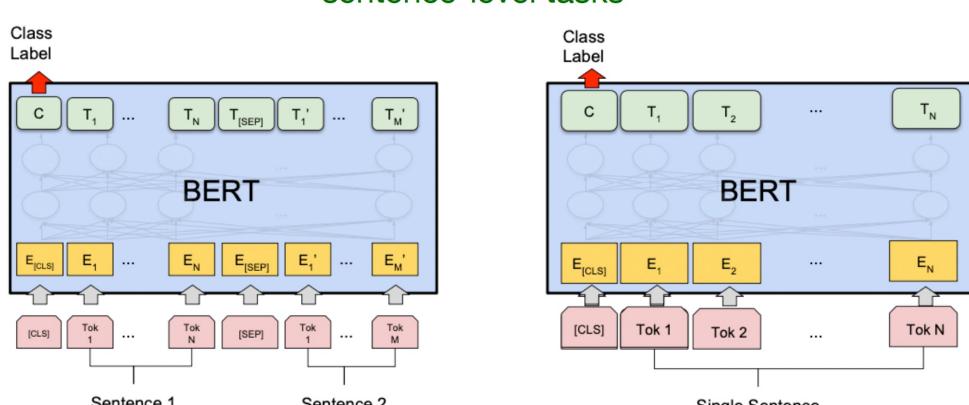
17

## Pretrain Once, Finetune Many Times



Stanford CS224N: <https://web.stanford.edu/class/cs224n/>  
Princeton COS 597G: <https://www.cs.princeton.edu/courses/archive/fall22/cos597G/lectures/lec02.pdf>

**sentence-level tasks**



The diagram illustrates BERT's architecture for different sentence-level tasks. It shows two main configurations:

(a) **Sentence Pair Classification Tasks:** MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG

This configuration processes two sentences, Sentence 1 and Sentence 2. Each sentence is tokenized into tokens (Tok 1, Tok N, Tok 1, ..., Tok M) and a special [CLS] token. These tokens are mapped to embeddings ( $E_{[CLS]}$ ,  $E_1$ , ...,  $E_N$ ,  $E_{[SEP]}$ ,  $E_1'$ , ...,  $E_M'$ ). The embeddings are fed into the BERT encoder, which consists of multiple layers of bidirectional attention. The final output is a sequence of hidden states ( $C$ ,  $T_1$ , ...,  $T_N$ ,  $T_{[SEP]}$ ,  $T_1'$ , ...,  $T_M'$ ) from which a class label is predicted.

(b) **Single Sentence Classification Tasks:** SST-2, CoLA

This configuration processes a single sentence. The sentence is tokenized into tokens ([CLS], Tok 1, Tok 2, ..., Tok N) and a special [CLS] token. These tokens are mapped to embeddings ( $E_{[CLS]}$ ,  $E_1$ ,  $E_2$ , ...,  $E_N$ ). The embeddings are fed into the BERT encoder, which consists of multiple layers of bidirectional attention. The final output is a sequence of hidden states ( $C$ ,  $T_1$ ,  $T_2$ , ...,  $T_N$ ) from which a class label is predicted.

18

18

## Pretrain Once, Finetune Many Times

Stanford CS224N: <https://web.stanford.edu/class/cs224n/>  
Princeton COS 597G: <https://www.cs.princeton.edu/courses/archive/fall22/cos597G/lectures/lec02.pdf>



**Token-Level Task**

- Extractive question answering e.g., SQuAD (Rajpurkar et al., 2016)  
**Standard Question Answer Dataset:** Predict the answer to the question.

SQuAD

**Question:** The New York Giants and the New York Jets play at which stadium in NYC ?

**Context:** The city is represented in the National Football League by the New York Giants and the New York Jets , although both teams play their home games at **MetLife Stadium** in nearby East Rutherford , New Jersey , which hosted Super Bowl XLVIII in 2014 .

(Training example 29,883)

MetLife Stadium

- Named entity recognition (Tjong Kim Sang and De Meulder, 2003)  
**Named Entity Recognition:** Recognize the entity of each word.

CoNLL 2003 NER

John Smith lives in New York

B-PER I-PER O O B-LOC I-LOC

19

19

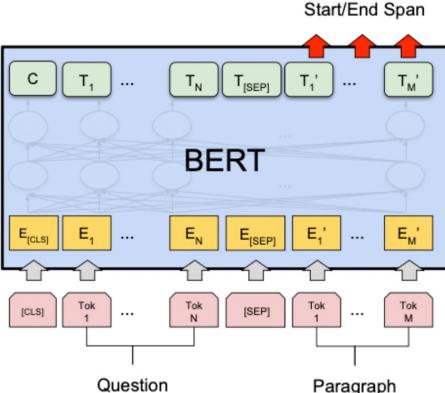
## Pretrain Once, Finetune Many Times

Stanford CS224N: <https://web.stanford.edu/class/cs224n/>  
Princeton COS 597G: <https://www.cs.princeton.edu/courses/archive/fall22/cos597G/lectures/lec02.pdf>



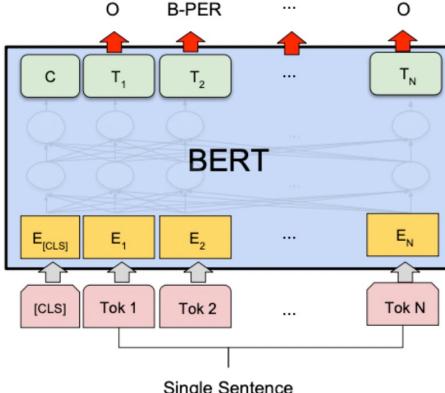
**token-level tasks**

**Start/End Span**



(c) Question Answering Tasks:  
SQuAD v1.1

**O B-PER ... O**



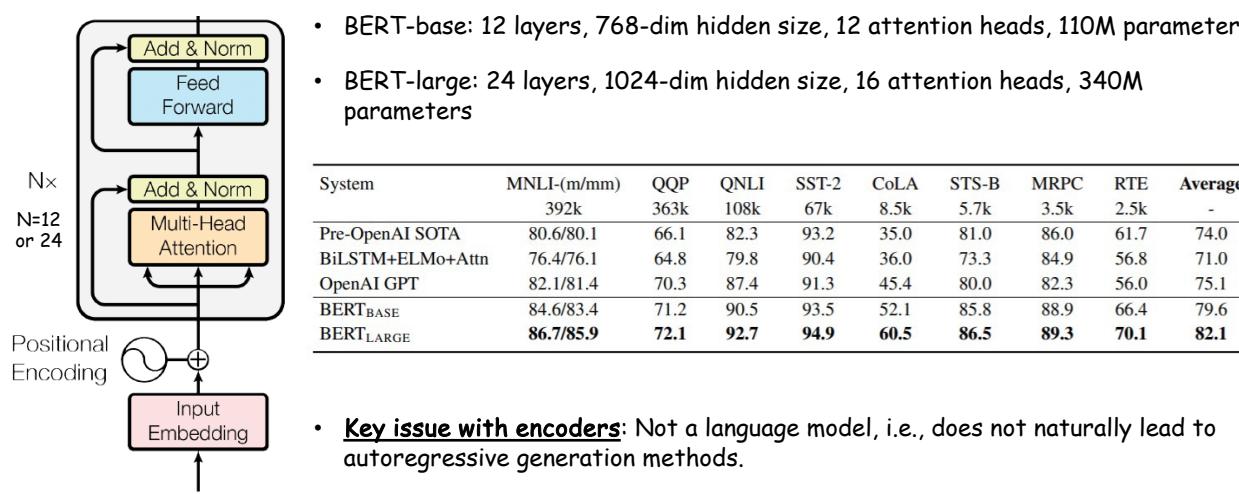
(d) Single Sentence Tagging Tasks:  
CoNLL-2003 NER

20

20

## BERT was the State-of-The-Art

Stanford CS224N: <https://web.stanford.edu/class/cs224n/>  
Princeton COS 597G: <https://www.cs.princeton.edu/courses/archive/fall22/cos597G/lectures/lec02.pdf>



The diagram illustrates the BERT architecture. It starts with 'Inputs' which are converted into 'Input Embedding'. This is followed by 'Positional Encoding' and a stack of  $N \times N=12$  or  $24$  layers. Each layer consists of a 'Multi-Head Attention' block (orange), an 'Add & Norm' block (yellow), a 'Feed Forward' block (light blue), another 'Add & Norm' block (yellow), and another 'Feed Forward' block (light blue). The final output is the sum of the input embedding and the output of the last layer's 'Add & Norm' block.

The table below compares various systems on nine NLP tasks:

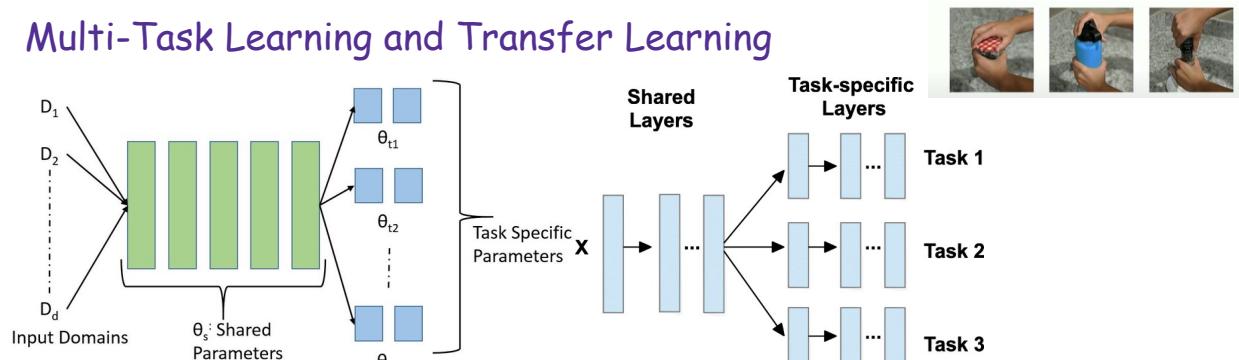
System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	Average
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
BERT <sub>BASE</sub>	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
BERT <sub>LARGE</sub>	<b>86.7/85.9</b>	<b>72.1</b>	<b>92.7</b>	<b>94.9</b>	<b>60.5</b>	<b>86.5</b>	<b>89.3</b>	<b>70.1</b>	<b>82.1</b>

- BERT-base: 12 layers, 768-dim hidden size, 12 attention heads, 110M parameters
- BERT-large: 24 layers, 1024-dim hidden size, 16 attention heads, 340M parameters
- **Key issue with encoders:** Not a language model, i.e., does not naturally lead to autoregressive generation methods.

21

21

## Multi-Task Learning and Transfer Learning



The diagram shows a multi-task learning architecture. On the left, multiple input domains  $D_1, D_2, \dots, D_d$  feed into a shared neural network. The network consists of 'Shared Parameters'  $\theta_s$  and 'Task Specific Parameters'  $\mathbf{X}$ . The shared parameters are used across all domains, while task-specific parameters are added for each task. The output of the shared network is then processed by 'Task-specific Layers' for three different tasks: Task 1, Task 2, and Task 3. The tasks are illustrated with images of hands performing different actions: opening a can, pouring water from a blue bottle, and closing a can.

- Transfer learning: Use the knowledge learned from solving one problem to solve another problem.
- Before BERT, transfer learning in NLP is used at feature representation level: e.g., directly use the prior word embeddings from word2vec as inputs of subsequent tasks.
- After BERT, transfer learning in NLP is mostly used with fine-tuning: e.g., take a pre-trained model and add trainable layers at the final stage to get certain outputs.

22

22

## FinBERT

- Pretrain BERT-base using financial datasets (4.9B tokens in total) with 4 P100 GPUs (100G memory):
  - Corporate annual and quarterly filings from SEC's EDGAR website (1994-2019).
  - Financial analyst reports from Thomson Intestext database (2003-2012).
  - Earnings conference call transcripts from the SeekingAlpha website (2004-2019).
- Finetuning and evaluation:**
  - Sentiment analysis 10,000 sentences
  - 36% positive
  - 46% neutral
  - 18% negative
- Can FinBERT beat GPT-4, Claude-3.5 or DeepSeek-V3 in tasks related to financial texts?
  - How can we make **fair comparisons?**

**Figure 1** Sentiment classification accuracy across sample sizes

Accuracy

% of full training and validation sample

**FinBERT: A large language model for extracting information from financial text**

AH Huang, H Wang, Y Yang - Contemporary Accounting ..., 2023 - Wiley Online Library  
... model that adapts to the **finance** domain. We show that **FinBERT** incorporates **finance** knowledge and can better summarize contextual **information** in **financial texts**. Using a sample of ...  
☆ Save 芚 Cite Cited by 144 Related articles Web of Science: 22 ☰

23

23

## Voice of Monetary Policy

### The Voice of Monetary Policy<sup>†</sup>

By YURIY GORODNICHENKO, THO PHAM, AND OLEKSANDR TALAVERA<sup>‡</sup>

We develop a deep learning model to detect emotions embedded in press conferences after the Federal Open Market Committee meetings and examine the influence of the detected emotions on financial markets. We find that, after controlling for the Federal Reserve's actions and the sentiment in policy texts, a positive tone in the voices of Federal Reserve chairs leads to significant increases in share prices. Other financial variables also respond to vocal cues from the chairs. Hence, how policy messages are communicated can move the financial market. Our results provide implications for improving the effectiveness of central bank communications. (JEL D83, E31, E44, E52, E58, F31, G14)

How can a president not be an actor?  
—Ronald Reagan (1980)

As Chairman, I hope to foster a public conversation about what the Fed is doing to support a strong and resilient economy. And one practical step in doing so is to have a press conference like this after every one of our scheduled FOMC meetings. ... [This] is only about improving communications.  
—Jerome Powell (2018)<sup>§</sup>

Monetary policy is 98 percent talk and 2 percent action, and communication is a big part.  
—Ben Bernanke (2022)<sup>¶</sup>

**The voice of monetary policy**

Y Gorodnichenko, T Pham, O Talavera - American Economic Review, 2023 - aeaweb.org  
... on recent advances in **voice** recognition technology and classify **the voice** tone of **the Fed** chairs into a spectrum of emotions. We, then, study how variations in **voice** tone (emotions) can ...  
☆ Save 芚 Cite Cited by 118 Related articles All 30 versions Web of Science: 9 ☰

24

24

## Agenda

- BERT: Bidirectional Encoder Representations from Transformers
- GPT: Generative Pretrained Transformers

25

25

## Pretraining Decoders

Stanford CS224N: <https://web.stanford.edu/class/cs224n/>

- Key idea: Pretrain decoders as language models  $\Pr(W_n | W_1, W_2, \dots, W_{n-1})$  via autoregression.

$$h_1, \dots, h_T = \text{Decoder}(w_1, \dots, w_T)$$

$$w_t \sim A h_{t-1} + b$$

This is a more challenging task than BERT!

[PPL: Improving language understanding by generative pre-training](#)

A Radford, K Narasimhan, T Salimans, I Sutskever

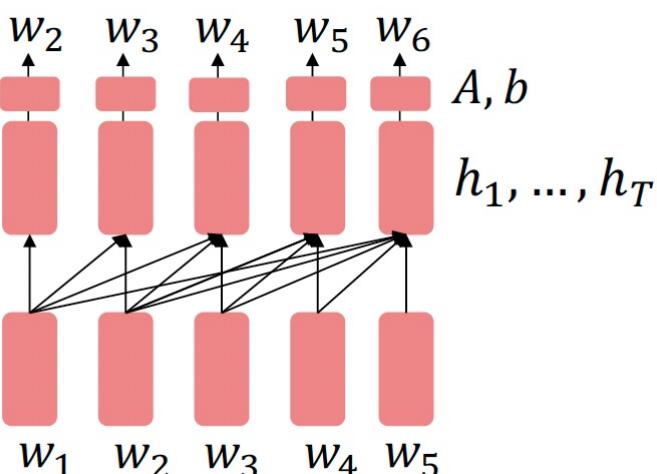
2018 - [mikecaptain.com](#)

### Abstract

Natural language understanding comprises a wide range of diverse tasks such as textual entailment, question answering, semantic similarity assessment, and document classification. Although large unlabeled text corpora are abundant, labeled data for learning these specific tasks is scarce, making it challenging for discriminatively trained models to perform adequately. We demonstrate that large gains on these tasks can be realized by generative pre-training of a language model on a diverse corpus of unlabeled

[SHOW MORE](#) ▾

[☆ Save](#) 99 [Cite](#) Cited by 8363 [Related articles](#) All 15 versions [⊗](#)



26

26

## GPT-1

Stanford CS224N: <https://web.stanford.edu/class/cs224n/>

- Architecture: Only masked self-attention, but deeper and larger.
- 12 layers of transformer decoders, 117M parameters.
- 768-dim hidden states, 3072-dim MLP hidden layers.
- Trained on BooksCorpus of over 7,000 unique books.

[PDF] Improving language understanding by generative pre-training  
A Radford, K Narasimhan, T Salimans, I Sutskever  
2018 - mikedcaption.com

Abstract  
Natural language understanding comprises a wide range of diverse tasks such as textual entailment, question answering, semantic similarity assessment, and document classification. Although large unlabeled text corpora are abundant, labeled data for learning these specific tasks is scarce, making it challenging for discriminatively trained models to perform adequately. We demonstrate that large gains on these tasks can be realized by generative pre-training of a language model on a diverse corpus of unlabeled data. We show that GPT, a generative pre-trained language model, can be finetuned to perform well on a variety of NLP tasks, including reading comprehension, question answering, and text generation. Our results indicate that GPT achieves state-of-the-art performance on several benchmarks, including the SQuAD 1.1 and 2.0 datasets, and outperforms previous models on the Multi-choice Question Answering (MCQA) dataset. We also show that GPT can be used to generate high-quality responses to a wide range of questions, including those from the TriviaQA dataset. Our findings suggest that generative pre-training can be a powerful tool for improving language understanding and generating high-quality text.

SHOW MORE ▾  
☆ Save 59 Cite Cited by 8363 Related articles All 15 versions ⓘ

The diagram illustrates the GPT-1 architecture. It starts with 'Text & Position Embed' at the bottom, which feeds into a stack of 12 layers. Each layer contains 'Masked Multi Self Attention' (red), 'Layer Norm' (purple), and 'Feed Forward' (orange). The output of each layer is combined with the input via residual connections (sum of input and output). The final output of the 12 layers is split into two paths: 'Pretraining' (top path) and 'Finetuning' (bottom path). The 'Pretraining' path leads to 'Text Prediction' and 'Task Classifier'. The 'Finetuning' path leads to various NLP tasks: Classification, Entailment, Similarity, and Multiple Choice. Each task involves a 'Transformer' block followed by a 'Linear' layer. In the 'Similarity' and 'Multiple Choice' sections, the input is split into two parallel paths, each with its own 'Transformer' and 'Linear' layer, before being combined via a residual connection.

27

## GPT-1 Finetuning

Stanford CS224N: <https://web.stanford.edu/class/cs224n/>

The diagram shows the GPT-1 architecture with its 12 layers (labeled '12x') receiving 'Text & Position Embed' at the bottom. The architecture is shared across four finetuning tasks:

- Classification:** Input sequence: Start, Text, Extract. Transformer → Linear.
- Entailment:** Input sequence: Start, Premise, Delim, Hypothesis, Extract. Transformer → Linear.
- Similarity:** Input sequences: Start, Text 1, Delim, Text 2, Extract and Start, Text 2, Delim, Text 1, Extract. Each sequence goes through a Transformer → Linear block, and the outputs are combined via a residual connection.
- Multiple Choice:** Input sequences: Start, Context, Delim, Answer 1, Extract; Start, Context, Delim, Answer 2, Extract; and Start, Context, Delim, Answer N, Extract. Each sequence goes through a Transformer → Linear block, and the outputs are combined via a residual connection.

**Finetuning Loss = Loss of Text Prediction + lambda \* Loss of Classification**

28

**GPT-2**

Stanford CS224N, Lecture 9: <https://web.stanford.edu/class/cs224n/>  
Andrej Karpathy's video reproducing GPT-2: <https://www.youtube.com/watch?v=l8pRSuU81PU>

• Same architecture but order of magnitude larger.

• 48 transformer blocks, 1,600 hidden units per layer, and 25 attention heads, 1.5B parameters.

• Trained on high-quality web-crawled data: 8M documents, 40GB.

• Understanding the contexts:

**Context (human-written):** In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

**GPT-2:** The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science.

Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.

Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

29

29

## Search Engine Optimization (SEO)

• Use **GPT-2** to draft the content for the landing page of a website in search engine optimization (SEO).

• Human-refined AI-generated content is indistinguishable from those created by SEO experts.

• Field experiments suggest AI-generated content outperforms human-generated content.

**Frontiers: Supporting Content Marketing with Natural Language Generation**

Martin Reisenbichler,<sup>a,\*</sup> Thomas Reutterer,<sup>a</sup> David A. Schweiöl,<sup>b</sup> Daniel Dan<sup>c</sup>

<sup>a</sup>Department of Marketing, Vienna University of Economics and Business, Vienna A-1020, Austria; <sup>b</sup>Generali Business School, Marketing Area, Innsbruck University, Innsbruck, Austria; <sup>c</sup>School of Applied Data Science, Modul University, Vienna, Vienna A-1190, Austria

Contract grant sponsor: Austrian Science Fund FWF

Received: June 30, 2021  
Revised: September 5, 2021  
Accepted: October 1, 2021  
Published Online in Article in Advance on April 20, 2022  
DOI: 10.1177/1525800022109297  
<https://journals.sagepub.com/doi/10.1177/1525800022109297>

**Abstract:** Advances in natural language generation (NLG) have facilitated technologies such as digital voice assistants and chatbots. In this research, we demonstrate how NLG can support content marketing by providing automated content generation solutions in search engine optimization (SEO). Traditional SEO projects rely on hand-crafted content creation, which is time-consuming and costly. To address this challenge, we propose an automated SEO content generation process. As part of our research, we demonstrate that although the machine-generated content is designed to perform well in search engines, the role of the human editor cannot be replaced. We conducted field experiments in two industries to demonstrate how AI-generated content can outperform SEO experts in search engine rankings. Additionally, we illustrate how our approach can reduce the production costs associated with content marketing, increasing their return on investment.

**History:** K. Sudhar served as the senior editor and Olivier Trubitsch served as associate editor for this article. This paper was accepted through the Marketing Science Frontiers review process.

**Supplemental material:** Supplemental material for this article is available at <https://journals.sagepub.com/doi/10.1177/1525800022109297>.

**Keywords:** SEO • content marketing • natural language generation • transfer learning

30

15

## GPT-3

Stanford CS224N: <https://web.stanford.edu/class/cs224n/>  
Princeton COS 5976: <https://www.cs.princeton.edu/courses/archive/fall22/cos597G/lectures/lec04.pdf>

- 175B Parameters: 96 decoder blocks, context size = 2,048, embedding-dim=12,288
- Trained on 300B tokens and 10,000 A100s for dozens of days.
- In context learning: Learning without updating gradients.

No Prompt	Prompt
Zero-shot (os)	Please unscramble the letters into a word, and write that word: skicts = sticks
1-shot (1s)	Please unscramble the letters into a word, and write that word: chiar = chair skicts = sticks

**Language models are few-shot learners**  
T Brown, B Mann, N Ryder... - Advances in neural ..., 2020 - proceedings.neurips.cc  
... up language models greatly improves task-agnostic, few-shot ... GPT-3, an autoregressive language model with 175 billion ... language model, and test its performance in the few-shot ...  
☆ Save ⚡ Cite Cited by 21815 Related articles All 27 versions ☰

31

31

## Pretraining Data for LLMs

Stanford CS224N: <https://web.stanford.edu/class/cs224n/>; FineWeb: <https://huggingface.co/spaces/HuggingFaceFW/blogpost-fineweb-v1>

- Orders of magnitude difference in sizes between labeled and unlabeled data:

Dataset	Tokens (~0.75 words)
SQuAD 2.0 [Rajpurkar+ 2018]	< 50 Million
DCLM-pool [Li+ 2024]	240 Trillion
Estimated ‘internet text’ [Villalobos 2024]	510T (indexed), 3100T (total)

A 10 million times gap in QA to indexed internet

The FineWeb pipeline

- Common Crawl: <https://commoncrawl.org/> (750T)
- SOTA models do not release their pretraining data (10~15T).
- Data ablation: Train language models on a dataset following a specific data curation decision.
- Evaluation of curated datasets: zero-shot in-context prompting.

<https://arxiv.org/abs/2402.00159>

32

32

## Tokenization

Stanford CS224N: <https://web.stanford.edu/class/cs224n/>; Ticktokenizer: <https://tiktok tokenizer.vercel.app/>

- Most used tokenization (except Google): Byte Pair Encoding (BPE)
- Efficiency vs. Effectiveness

**Algorithm 1** Byte-pair encoding (Sennrich et al., 2016; Gage, 1994)

```

1: Input: set of strings  $D$ , target vocab size  $k$ 
2: procedure BPE( $D, k$ )
3:    $V \leftarrow$  all unique characters in  $D$ 
4:   (about 4,000 in English Wikipedia)
5:   while  $|V| < k$  do            $\triangleright$  Merge tokens
6:      $t_L, t_R \leftarrow$  Most frequent bigram in  $D$ 
7:      $t_{\text{NEW}} \leftarrow t_L + t_R$      $\triangleright$  Make new token
8:      $V \leftarrow V + [t_{\text{NEW}}]$ 
9:     Replace each occurrence of  $t_L, t_R$  in
10:     $D$  with  $t_{\text{NEW}}$ 
11:   end while
12:   return  $V$ 
13: end procedure

```

Stanford CS336: <https://stanford-cs336.github.io/spring2024/>

Iteration	Corpus	Vocabulary
0	AACGCACTATATA	{A,T,C,G}
1	A A C G C A C T A T A T A	{A,T,C,G,TA}
2	A A C G C A C T A T A T A	{A,T,C,G,TA, AC}
3	A A C G C A C T A T A	.....

Figure 2: Illustration of the BPE vocabulary constructions.

Monolingual models – 30-50k vocab

Model	Token count
Original transformer	37000
GPT	40257
GPT2/3	50257
T5/T5v1.1	32128
LLaMA	32000

Multilingual / production systems 100-250k

Model	Token count
mT5	250000
PaLM	256000
GPT4	100276
BLOOM	250680
DeepSeek	100000
Qwen 15B	152064
Yi	64000

Multilingual vocabularies are much larger.

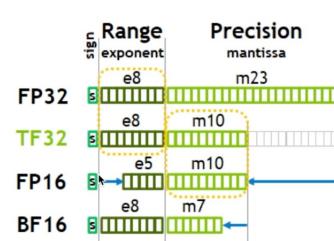
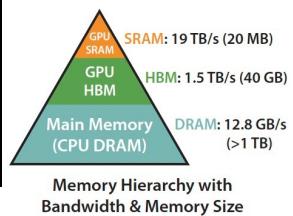
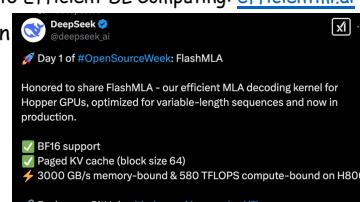
33

33

## Compute Efficient Training with GPUs

Andrej Karpathy's lecture: <https://www.youtube.com/watch?v=l8pRSuU81PU>; Stanford CS336: <https://stanford-cs336.github.io/spring2025/>; Flash Attention Paper: <https://arxiv.org/pdf/2205.14135.pdf>; MIT 6.5940 Efficient DL Computing: [efficientml.ai](https://efficientml.ai)

- Make sure you implement your model in PyTorch and train it on GPUs; be hardware-aware.
  - Data parallelism: Distributed Data Parallel (DDP)
- Choose a nice number: Multiples of  $2^k$ 
  - CUDA's are implemented in blocks of  $2^k$ .
- Do not reinvent the wheels (e.g., Flash Attention as a function in PyTorch).
- Choose the right batch size: The largest supported by your hardware.
  - Use gradient accumulation to increase the equivalent batch size.
  - Scale learning rate accordingly.
- Reduce precision: Use BF16 or even FP8.
- Apply learning rate scheduler (AdamW).
- Initialize the training properly.
  - He Initialization.



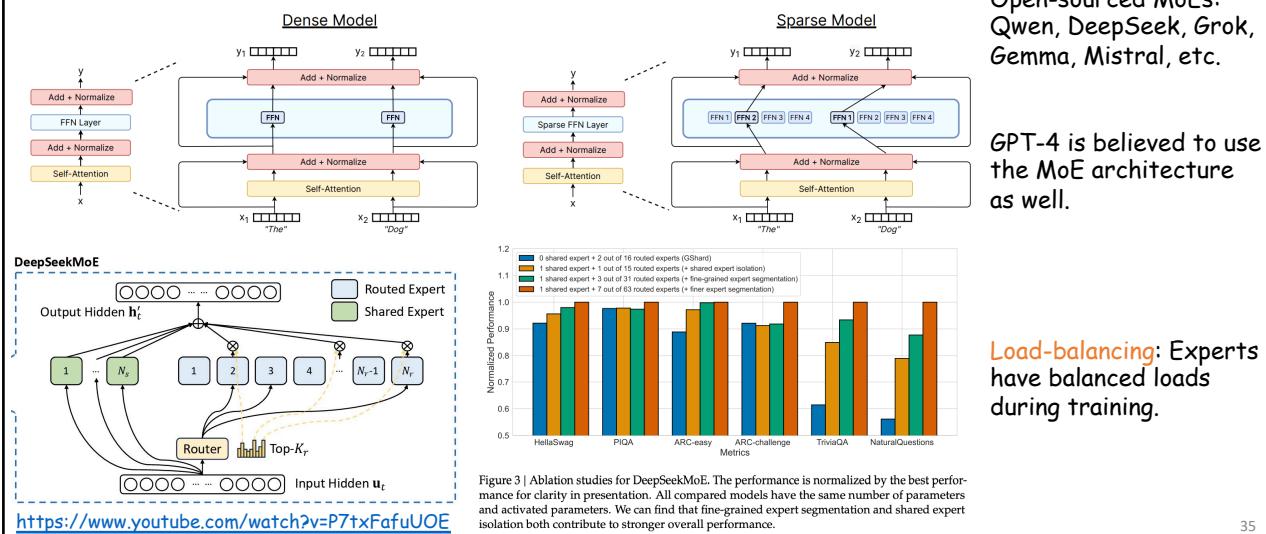
34

34

## Mixture of Experts (MoE)

Stanford CS336: <https://stanford-cs336.github.io/spring2024/>; DeepSeek-V3 & MoE: <https://arxiv.org/pdf/2412.19437v1.pdf>, <https://arxiv.org/pdf/2401.06066.pdf>; Princeton COS 597G: <https://www.cs.princeton.edu/courses/archive/fall22/cos597G/lectures/lec04.pdf>

- MoE: Leverage sparsity for more efficient training
  - DeepSeek-V3 has 671B parameters, but each token only activates 37B.



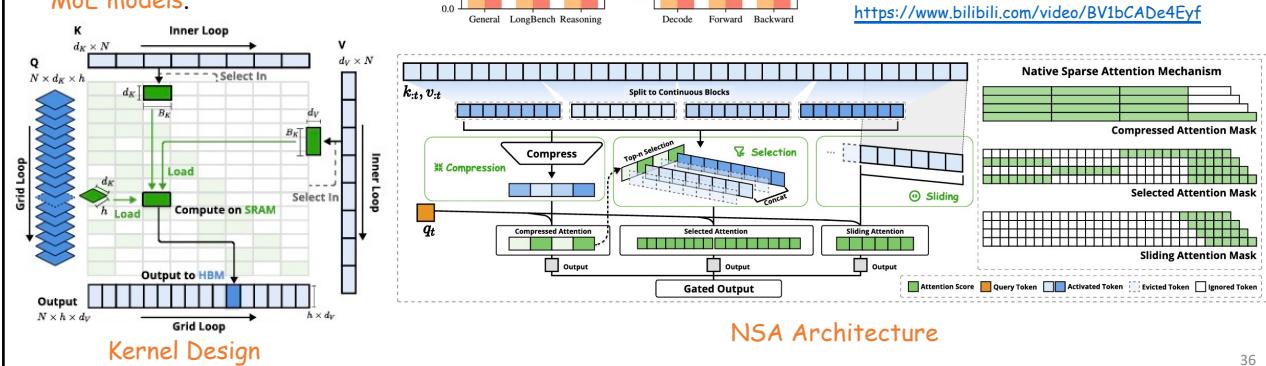
35

35

## Native Sparse Attention (NSA)

DeepSeek-NSA: <https://arxiv.org/pdf/2502.11089.pdf>; OpenAI Sparse Attention: <https://arxiv.org/pdf/1904.10509.pdf>

- Sparse Attention has been implemented at the inference stage to address long context-window.
- But NSA is hardware-aligned (redesigned kernel on Triton), trainable and powerful.
- Experimented on 24B MoE models.



36

36

A pretrained LLM is called a **base model**, which is essentially a **next-token simulator** that approximates **ALL** the texts on the Internet.

## Hallucination is inevitable!!!

To make the base model **aligned with human behaviors** in specific contexts, we need **posttraining**.

37

37