

# Machine Learning (CE 40717)

## Fall 2024

Ali Sharifi-Zarchi

CE Department  
Sharif University of Technology

December 7, 2024



These slides were prepared with contributions from Aren Golazizian

## 1 Transformers

## 2 Encoder Architecture

## 3 Decoder Architecture

## 4 References

## 1 Transformers

## 2 Encoder Architecture

### 3 Decoder Architecture

## 4 References

## Attention is All You Need!

## Attention Is All You Need

Ashish Vaswani\*  
Google Brain  
avaswani@google.com

Noam Shazeer\*  
Google Brain  
[noam@google.com](mailto:noam@google.com)

Niki Parmar\*  
Google Research  
[nip@google.com](mailto:nip@google.com)

Job Uszkoreit\*  
Google Research  
google.com

Llion Jones\*  
Google Research  
[llion@google.com](mailto:llion@google.com)

Aidan N. Gomez\*  
University of Toronto  
aidan@cs.toronto.edu

Lukasz Kaiser\*  
Google Brain  
lukasz.kaiser@google.com

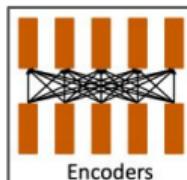
**Illia Polosukhin\*** ‡

## Abstract

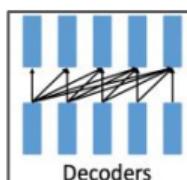
The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.0 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature.



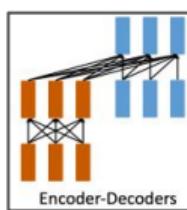
## Transformer Architectures



Encoder-only (e.g., BERT): bidirectional contextual embeddings



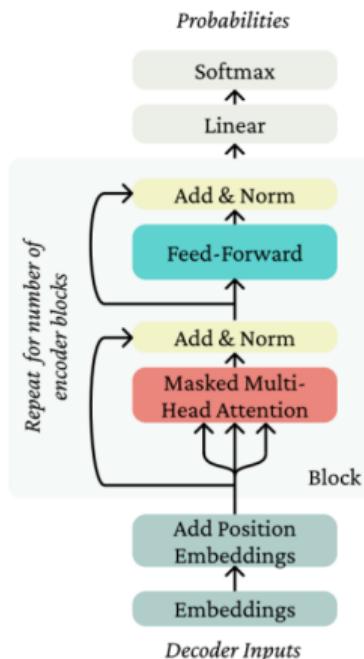
Decoder-only (e.g., GPT-x): unidirectional contextual embeddings, generate one token at a time



Encoder-decoder (e.g., T5): encode input, decode output

## Transformer block

- Each block has two "sublayers"
    1. Multihead attention
    2. Feed-forward Neural Network (with ReLU)
  - Residual:  $x + \text{Sublayer}(x)$
  - Layer Normalization changes input to have mean 0 and variance 1



## Layer normalization

**Main Idea:** Batch normalization is advantageous for stability but presents challenges with sequences of varying lengths.

**Result:** It provides a more stable input for the next layer

**Solution:** "Layer normalization" functions similarly to batch normalization but doesn't normalize across the entire batch.

## Batch norm

$$\mu = \frac{1}{B} \sum_{i=1}^B a_i, \quad \sigma = \sqrt{\frac{1}{B} \sum_{i=1}^B (a_i - \mu)^2}$$

*d-dimensional vectors for each sample in batch*

## Layer norm

$$\mu = \frac{1}{d} \sum_{i=1}^d a_j, \quad \sigma = \sqrt{\frac{1}{d} \sum_{i=1}^d (a_j - \mu)^2}$$

## *Different dimensions of a*

## Why Transformers?

# Why Transformers?

Pros:

- + Much easier to parallelize
  - + Much better long-range connections
  - + In practice, can make it much deeper (more layers) than RNN

### **Cons:**

- Attention computations are technically  $O(n^2)$
  - Somewhat more complex to implement (positional encodings, etc.)

## 1 Transformers

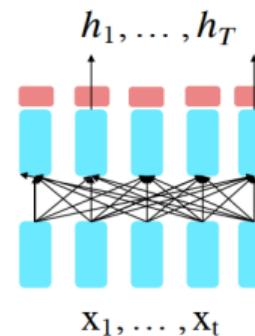
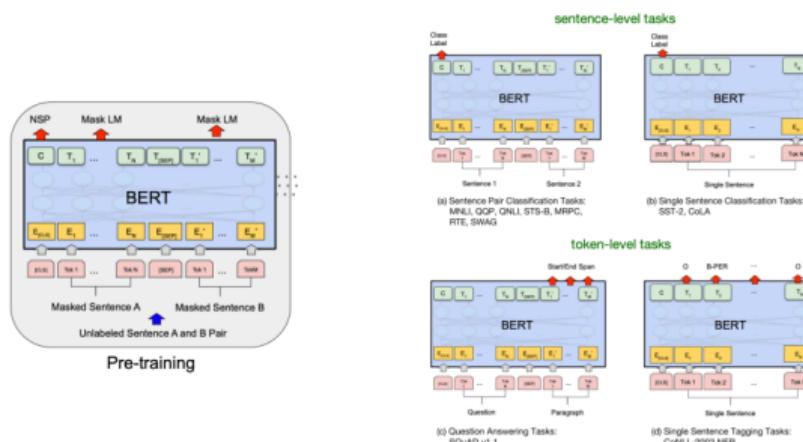
## 2 Encoder Architecture

## 3 Decoder Architecture

## 4 References

# Encoder Language Model

$$P(x) = \prod_{i=1}^n P(x_i | x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)$$



$$h_1, \dots, h_t = \text{Decoder}(x_1, \dots, x_t)$$

$$\mathbf{x}_{\text{mask}} \sim A\mathbf{h}_{\text{masked}} + b$$

# Encoder Language Model

Encoder language models, like BERT, use masked tokens to learn bidirectional representations of text.

- **Masked Language Modeling (MLM):** Predicts randomly masked tokens in a sequence.
  - **Bidirectional Context:** Considers information from both directions for each token.
  - **Applications:** Used for classification, NER, and other NLP tasks.

## BERT: Key Contributions

- It is a fine-tuning approach based on a deep Transformer Encoder
  - The key: learn representations based on **bidirectional context**

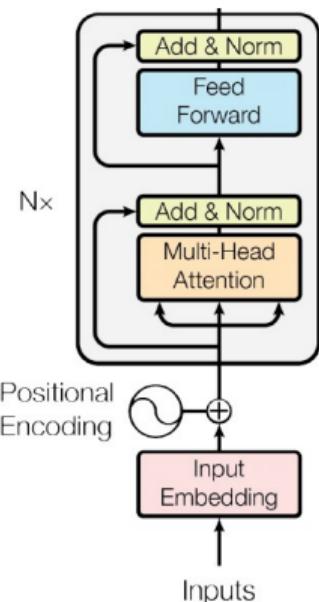
Why? Because both left and right contexts are important to understand the meaning of words.

Example #1: we went to the river **bank**

Example #2: I need to go to **bank** to make a deposit

- **Pre-training objectives:** masked language modeling + next sentence prediction
  - State-of-the-art performance on a large set of **sentence-level** and **token-level** tasks

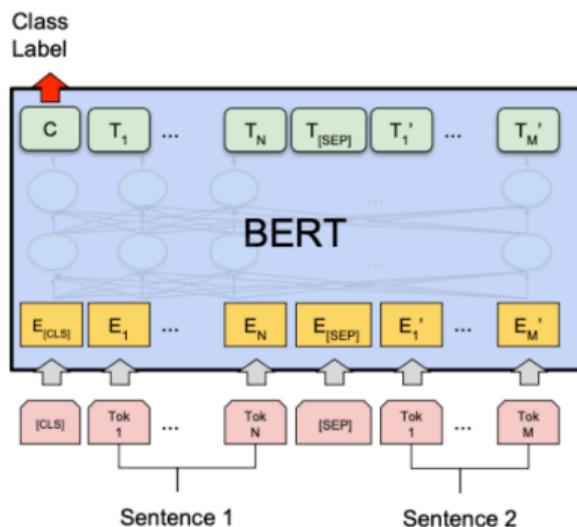
# BERT pre-training: putting together



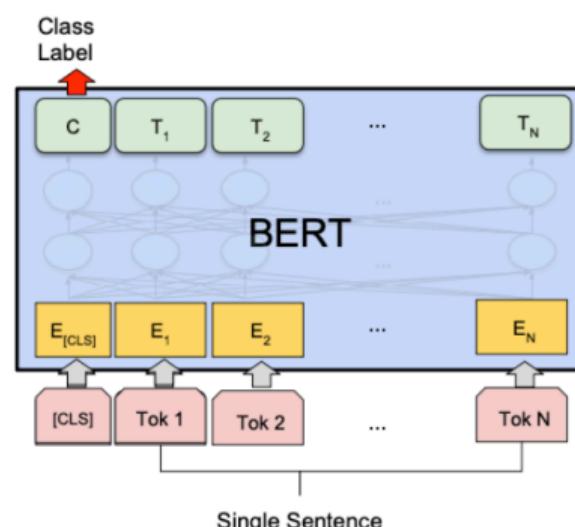
- **BERT-Base:** 12 layers, 768 hidden size, 12 attention heads, 110M parameters
- **BERT-Large:** 24 layers, 1024 hidden size, 16 attention heads, 340M parameters
- **Training corpus:** Wikipedia (2.5B) + BooksCorpus (0.8B)
- **Max sequence size:** 512 word pieces (roughly 256 and 256 for two non-contiguous sequences)
- **Trained for:** 1M steps, batch size 128k

## Sentence-level tasks

### sentence-level tasks



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



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

## Sentence-level tasks(cont.)

- Sentence pair classification tasks:

MNLI

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

QQP

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

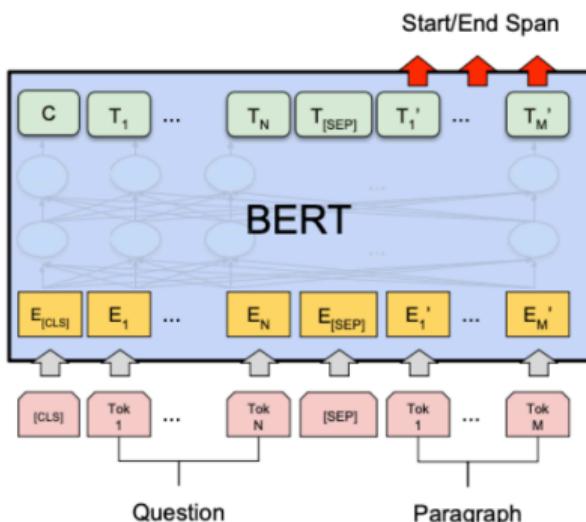
- Single sentence classification tasks

SST2

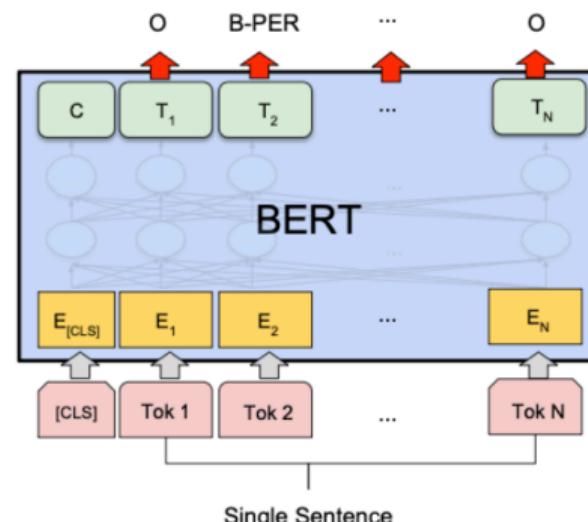
- Sentence: rich veins of funny stuff in this movie
  - Result: {positive, negative}

## Token-level tasks

### token-level tasks



(c) Question Answering Tasks:  
SQuAD v1.1



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

## Token-level tasks: Extractive Question Answering

- Extractive question answering e.g., SQuAD (Rajpurkar et al., 2016)

### 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)

Example Result: MetLife Stadium

# Token-level tasks: Named Entity Recognition

## Token-level tasks

- Named entity recognition ([Tjong Kim Sang and De Meulder, 2003](#))

### CoNLL 2003 NER

John	Smith	lives	in	New	York
B-PER	I-PER	O	O	B-LOC	I-LOC

# Masked Language Modeling (MLM)

- **Q:** Why we can't do language modeling with bidirectional models?



- **Solution:** Mask out a percentage  $k$  of the input words, and then predict the masked words.

store  
gallon

the man went to [MASK] to buy a [MASK] of milk

# MLM: Masking Rate and Strategy

- **Q: What is the value of k?**

- They always use  $k = 15\%$ .
- Too little masking: computationally expensive (we need to increase # of epochs)
- Too much masking: not enough context
- See [\(Wettig et al., 2022\)](#) for more discussion of masking rates:
  - Masking 40% outperforms 15% for BERT-large size models on GLUE and SQuAD
  - High masking rate of 80% can still preserve 95% fine-tuning performance

- **Q: How are masked tokens selected?**

- 15% tokens are uniformly sampled
- Is it optimal? See span masking [\(Joshi et al., 2020\)](#) and PMI masking [\(Levine et al., 2021\)](#)

**Example:** He [MASK] from Kuala [MASK], Malaysia.

## Next Sentence Prediction (NSP)

- Motivation: many NLP downstream tasks require understanding the relationship between two sentences (natural language inference, paraphrase detection, QA).
- NSP is designed to reduce the gap between pre-training and fine-tuning.

[CLS]: a special token  
always at the beginning

**Input** = [CLS] the man went to [MASK] store [SEP]

he bought a gallon [MASK] milk [SEP]

**Label** = IsNext

[SEP]: a special token used  
to separate two segments



They sample two contiguous  
segments for 50% of the  
time and another random  
segment from the corpus for  
50% of the time

**Input** = [CLS] the man [MASK] to the store [SEP]

penguin [MASK] are flight ##less birds [SEP]

**Label** = NotNext

# BERT Training

**Dataset:** Let  $\mathcal{D}$  be a set of examples  $(x_{1:L}, c)$  constructed as follows:

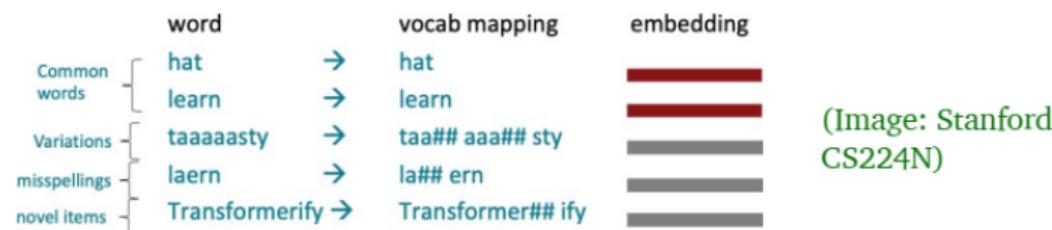
- Let  $A$  be a sentence from the corpus.
- With probability 0.5, let  $B$  be the next sentence.
- With probability 0.5, let  $B$  be a random sentence from the corpus.
- Let  $x_{1:L} = [\text{CLS}], A, [\text{SEP}], B$ .
- Let  $c$  denote whether  $B$  is the next sentence or not.

**Objective.** Then the BERT objective is:

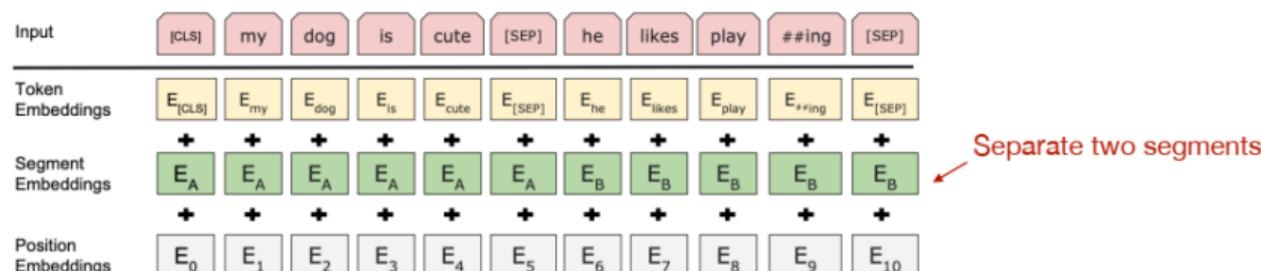
$$\mathcal{O}(\theta) = \sum_{(x_{1:L}, c) \in \mathcal{D}} \underbrace{\mathbb{E}_{I, \tilde{x}_{1:L} \sim A(\cdot | x_{1:L}, I)} \left[ \sum_{i \in I} -\log p_\theta(\tilde{x}_i | x_{1:L}) \right]}_{\text{masked language modeling}} + \underbrace{-\log p(c | \phi(x_{1:L})_1)}_{\text{next sentence prediction}}$$

# BERT Pre-training: Putting Together

- Vocabulary size:** 30,000 wordpieces (common sub-word units) (Wu et al., 2016)



- Input embeddings:**

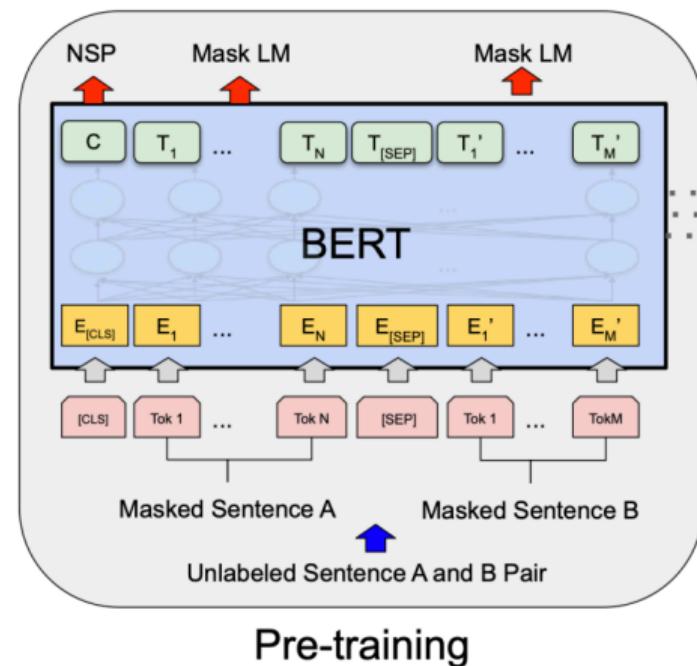


- Just two possible "segment embeddings":  $EA$  and  $EB$ .

- Positional embeddings are learned vectors for every possible position between 0 and 512-1.

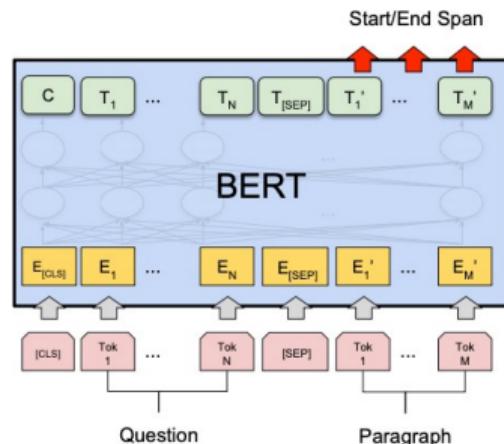
# BERT Pre-training: Putting Together

- MLM and NSP are trained together
- [CLS] is pre-trained for NSP
- Other token representations are trained for MLM

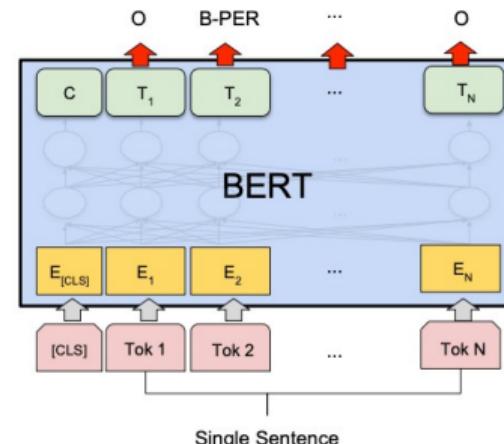


## Fine-tuning BERT

“Pre-train once, finetune many times.”  
**token-level tasks**



(c) Question Answering Tasks:  
SQuAD v1.1



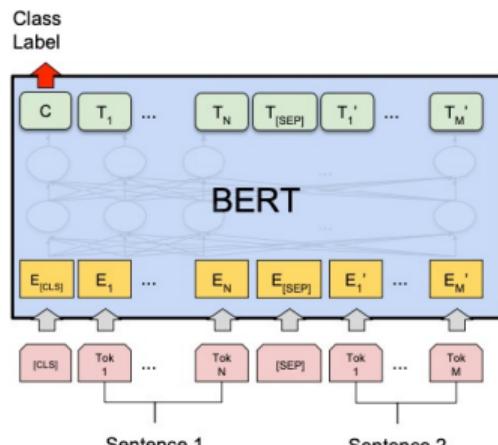
(d) Single Sentence Tagging Tasks:  
CoNLL-2003 NER

For token-level prediction tasks, add linear classifier on top of hidden representations

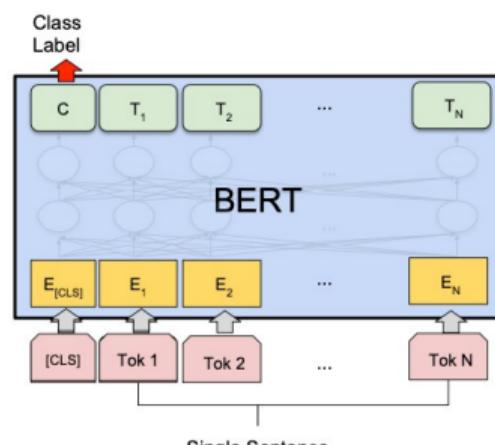
Q: How many new parameters?

# Fine-tuning BERT

“Pre-train once, finetune many times.”  
sentence-level tasks



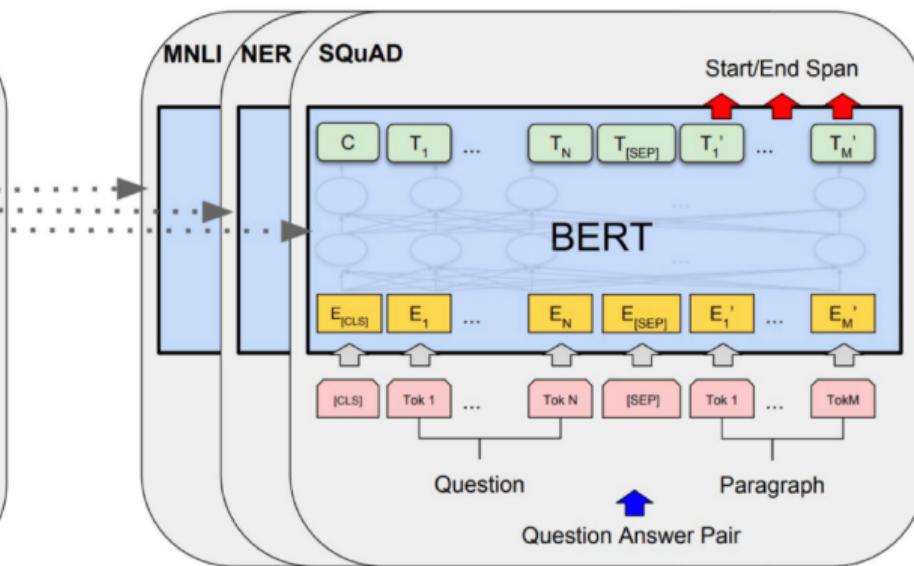
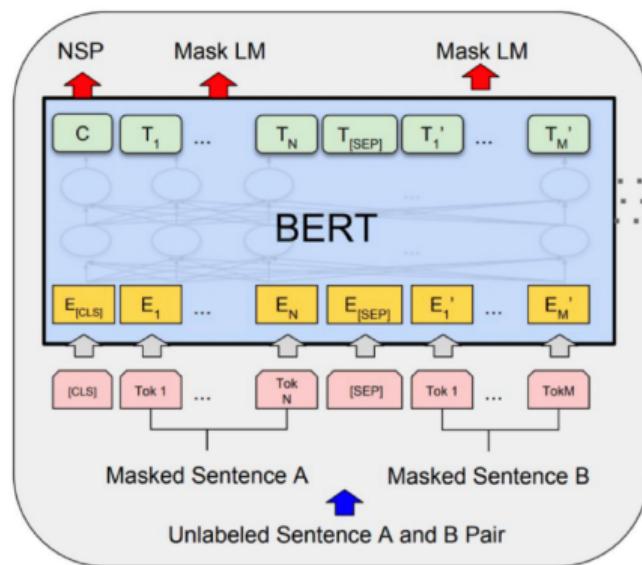
(a) Sentence Pair Classification Tasks:  
MNLI, QQP, QNLI, STS-B, MRPC,  
RTE, SWAG



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

For sentence pair tasks, use [SEP] to separate the two segments with segment embeddings and add a linear classifier on top of [CLS] representation.

## Finetuning Paradigm in NLP



## BERT Extensions

- Models that handle long contexts (> 512 tokens)
    - Longformer, Big Bird, ...
  - Multilingual BERT
    - Trained single model on 104 languages from Wikipedia. Shared 110k WordPiece vocabulary
  - BERT extended to different domains
    - SciBERT, BioBERT, FinBERT, ClinicalBERT, ...
  - Making BERT smaller to use
    - DistillBERT, TinyBERT, ...

# BERT Extensions

- **RoBERTa** (Liu et al., 2019)
  - Trained on 10x data & longer, no NSP
  - Much stronger performance than BERT (e.g., 94.6 compared to 90.9 on SQuAD)
  - Still one of the most popular models to date
- **ALBERT** (Lan et al., 2020)
  - Increasing model sizes by sharing model parameters across layers
  - Less storage, much stronger performance but runs slower

## ① Transformers

## ② Encoder Architecture

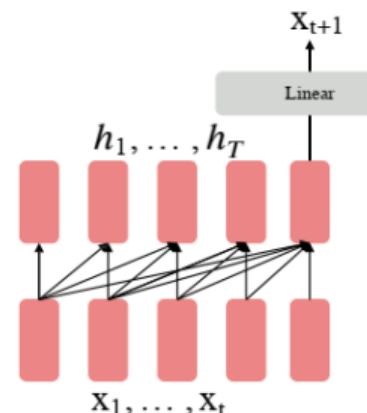
## ③ Decoder Architecture

## ④ References

## Decoder Language Model

**Autoregressive (AR)** models use Decoder stacks in generation, aiming to maximize log-likelihood via forward autoregressive factorization:

$$\max_{\theta} \log p_{\theta}(x_1, \dots, x_T) \approx \sum_{t=1}^T \log p_{\theta}(x_t | x_1, \dots, x_{t-1})$$

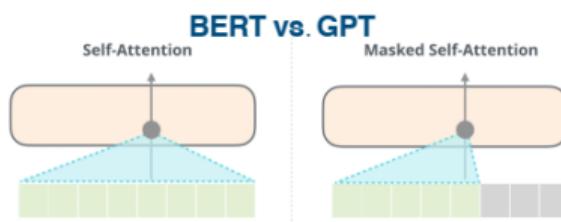
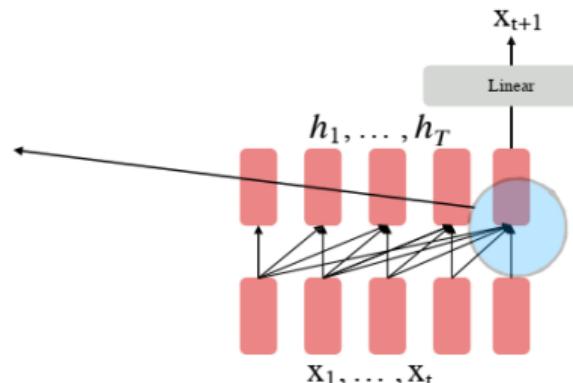
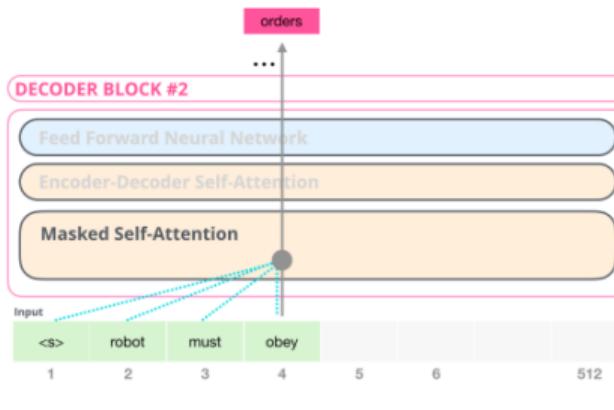


$$h_1, \dots, h_t = \text{Decoder}(x_1, \dots, x_t)$$

$$x_{t+1} \sim A h_t + b$$

## Decoder Language Model

$$\max_{\theta} \log p_{\theta}(x_1, \dots, x_T) \approx \sum_{t=1}^T \log p_{\theta}(x_t | x_1, \dots, x_{t-1})$$

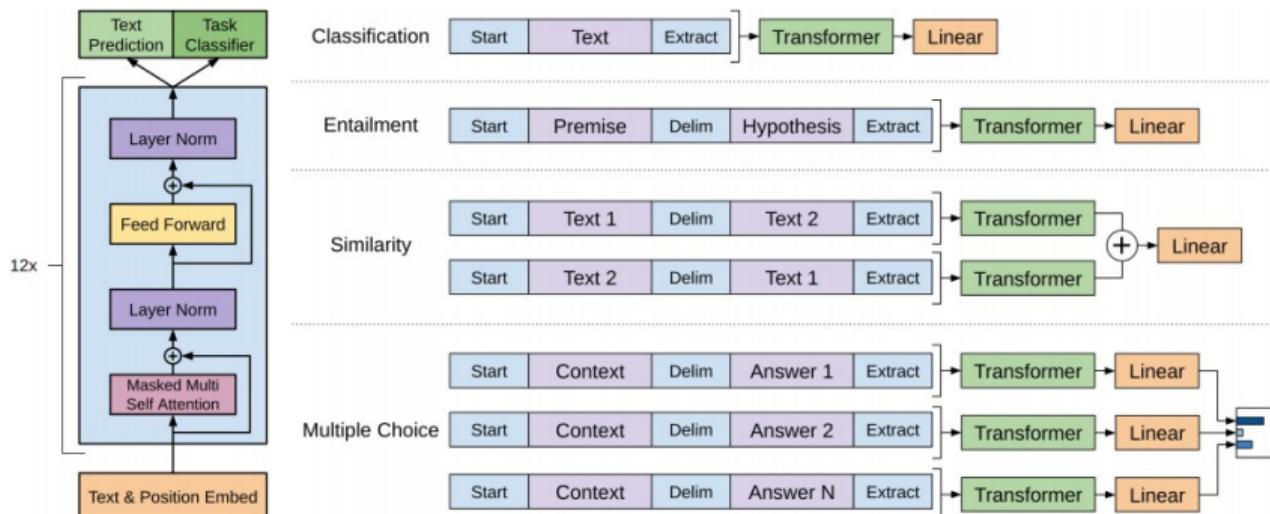


$$h_1, \dots, h_t = \text{Decoder}(x_1, \dots, x_t)$$

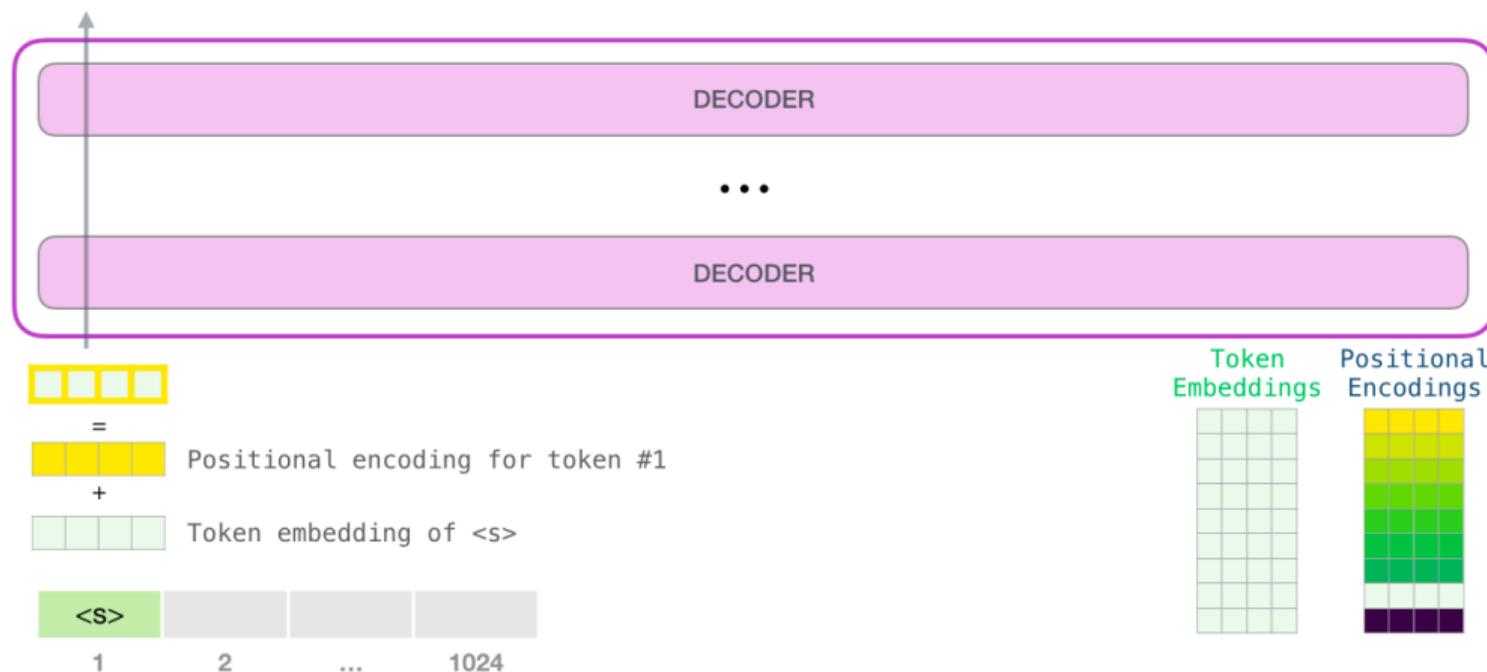
$$x_{t+1} \sim Ah_t + b$$

# Generative Pre-Trained Transformer (GPT)

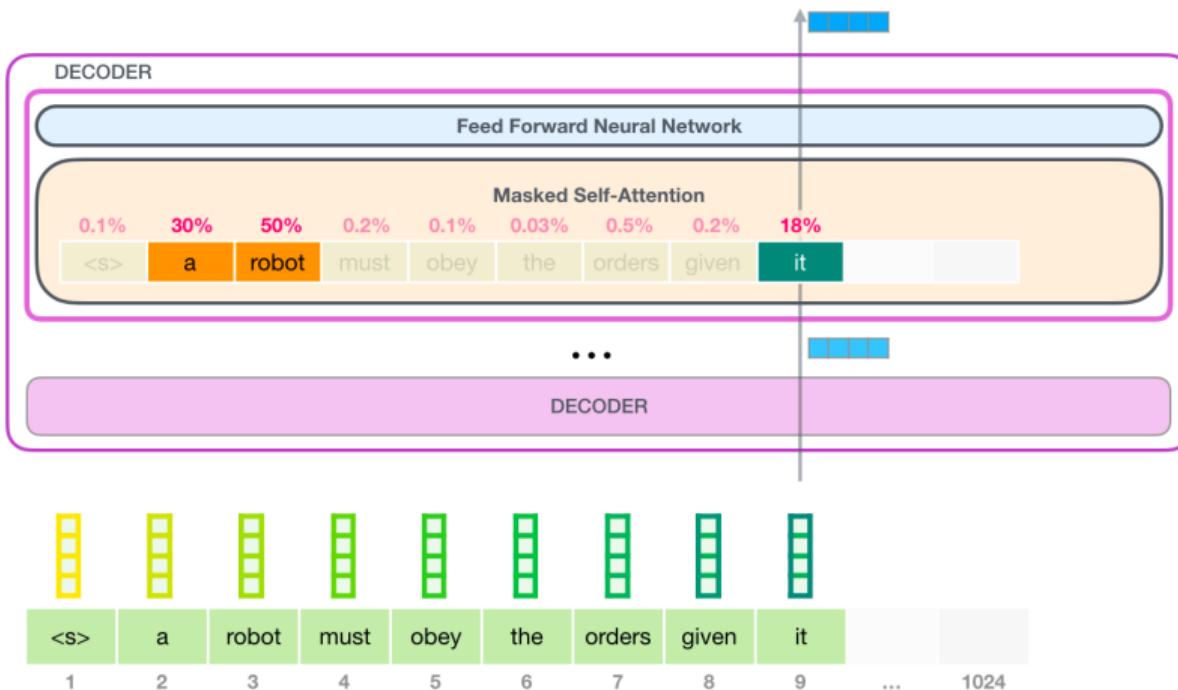
- Transformer Decoder with 12 layers.
- Byte Pair Encoding (BPE) with 40,000 merges.
- Trained on BooksCorpus: over 7000 unique books.
  - Contains long spans of contiguous text, for learning long-distance dependencies.



# Generative Pre-Trained Transformer (GPT)

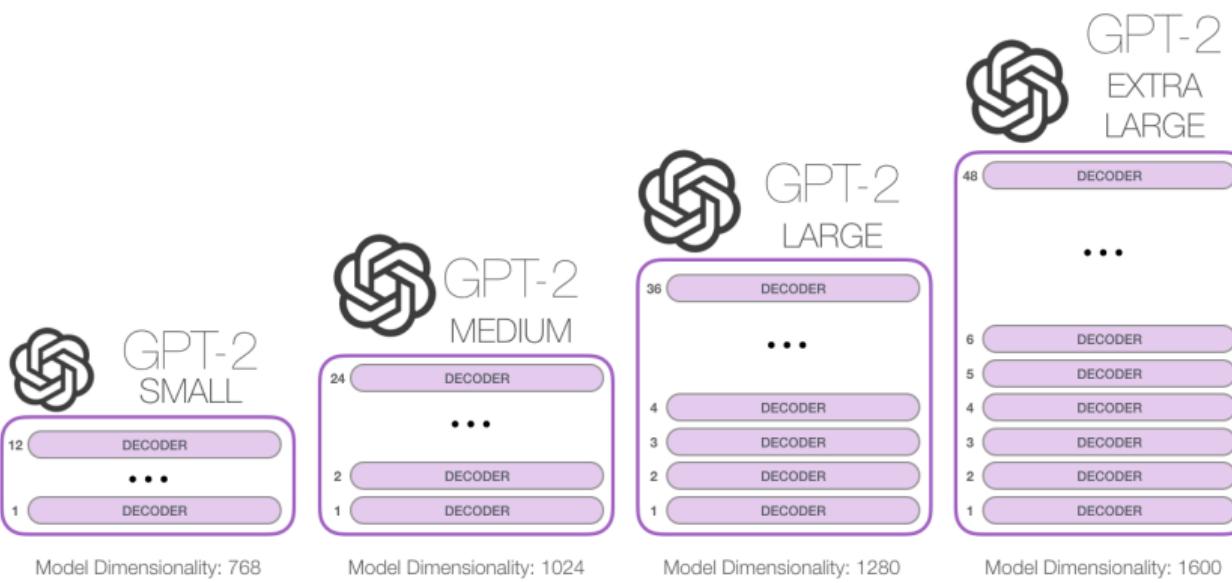


## Generative Pre-Trained Transformer (GPT)



# Generative Pre-Trained Transformer (GPT)

- GPT released June 2018
- GPT-2 released Nov. 2019 with 1.5B parameters
- GPT-3: 175B parameters trained on 45TB texts



## GPT Model Comparison

Model	Description	Data
<b>GPT-2</b> (Radford et al., 2019)	Context size: 1024 tokens, 117M-1.5B parameters	WebText (45 million outbound links from Reddit with 3+ karma); 8 million documents (40GB)
<b>GPT-3</b> (Brown et al., 2020)	Context size: 2048 tokens, 125M-175B parameters	Common Crawl + WebText + “two internet-based books cor- pora” + Wikipedia (400B to- kens, 570GB)

# Few-Shot, One-Shot, and Zero-Shot Learning in Language Models

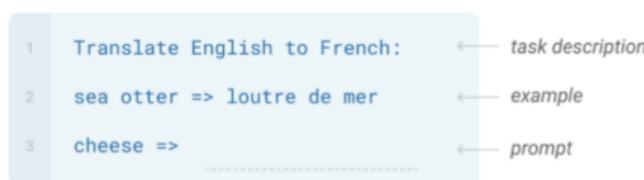
## Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.



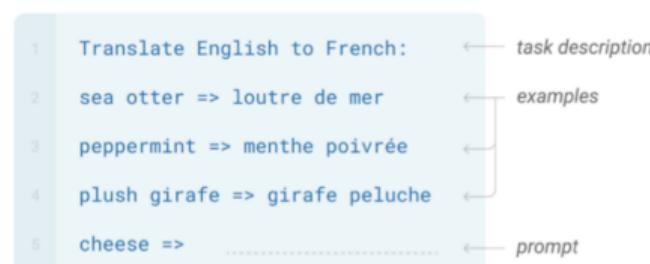
One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.



Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.



Brown et al. (2020, "Language Models are Few-Shot Learners"  
<https://arxiv.org/pdf/2005.14165.pdf>

## 1 Transformers

## 2 Encoder Architecture

## 3 Decoder Architecture

## 4 References

## References I

- Asgari, E. "Natural language processing." Sharif University of Technology.
  - Soleymani, M. "Machine learning." Sharif University of Technology.
  - Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I. (2019). "Language Models are Unsupervised Multitask Learners." OpenAI Blog. Retrieved from <https://openai.com/research/language-unsupervised>
  - Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... Amodei, D. (2020). "Language Models are Few-Shot Learners." *arXiv preprint arXiv:2005.14165*.
  - Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... Stoyanov, V. (2019). "RoBERTa: A Robustly Optimized BERT Pre-training Approach." *arXiv preprint arXiv:1907.11692*.
  - Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R. (2020). "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations." In *Proceedings of the International Conference on Learning Representations (ICLR)*.
  - Joshi, M., Chen, D., Liu, Y., Weld, D. S., Zettlemoyer, L., Levy, O. (2020). "SpanBERT: Improving Pre-training by Representing and Predicting Spans." *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 64-77.

## References II

- Wettig, A., Baykal, C., Ruder, S., Søgaard, A. (2022). "Should All Tokens be Masked? A Pilot Study of Masked Language Model Performance on Diagnostic Classifiers." *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 10, pp. 10993-11001.
  - Tjong Kim Sang, E. F., De Meulder, F. (2003). "Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition." In *Proceedings of the 7th Conference on Natural Language Learning at HLT-NAACL 2003*.
  - Radford, A., Narasimhan, K., Salimans, T., Sutskever, I. (2018). "Improving Language Understanding by Generative Pre-Training." Retrieved from <https://www.cs.ubc.ca/~amuham01/LING530/papers/radford2018improving.pdf>