Research Readiness Presentation

By Tam Doan

Outline

A. Introduction

B. Presenting papers

1. "Improving Language Understanding by Generative Pre-Training " (GPT1)
    - 1.1. Problem and previous method
    - 1.2. Overview method(focus math)
    - 1.3. Result
    - 1.4. What did GPT1 achieve

2. "Language Models are Unsupervised Multitask Learners" (GPT2)
    - 2.1. Problem and previous method
    - 2.2. Overview method
    - 2.3.Result
    - 2.4. What did GPT2 achieve

3. "Language Models are Few-Shot Learners" (GPT3)
    - 3.1. Problem and proposal method
    - 3.2. Overview method
    - 3.3.Training Data
    - 3.4.Result
    - 3.5. What did GPT3 achieve

C. Discussion

D. Reference

Introduction

"Natural language processing is the set of methods for making human language accessible to computers"(NLP) (Jacob Eisenstein)

- Machine learning with NLP

- ChatGPT and GPT4

- Investigate previous versions of ChatGPT: GPT1, GPT2, and GPT3

Eisenstein, Jacob. *Introduction to natural language processing*. MIT press, 2019.

Improving Language Understanding by Generative Pre-Training (GPT1)

Published in June 2018 by

Alec Radford, Karthik Narasimhan, Tim Salimans,Ilya Sutskever

Present by Tam Doan

Background

TIME

Problem for deep learning	Approach to solve	Example
• need big amount of label data	➢ learn from unlabeled data ➢ transfer learning	✓ word embeddings ✓ phrase-level embeddings ✓ sentence-level embeddings
• which model? • which technique?	➢ learn useful universal representations from unlabeled text ➢ transfer learning with minimum changes of architecture	✓ A language model long short term memory (Dai and Le, 2015)
• A specific architecture for a specific downstream task	➢ attention blocks architecture for pretraining model ➢ input transformations for specific tasks	✓ GPT1

➔ Without a change of architecture, GPT1 can perform well many specific tasks with supervised finetuning

Overview GPT1: Unsupervised pre-training (step1)

❖Available data

❖ Given an unlabeled dataset T = {$u_1, \ldots, u_n$}

- maximize the likelihood L_1(T) :

$$L_1(T) = \sum_{i=1}^{n} \log P\,(u_i / u_{i-k}, \ldots, u_{i-1}; \theta) \qquad (1)$$

➢ u_i is the predicted next word(output).The context window has size k. The conditional probability P is GPT1 model with parameter θ.

$$Attention(Q, K, V) = [softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)]V \qquad (5)$$

(Vaswani et al., 2017)

- Let x_i is a vector embedding a token and it's position of each input word and X is matrix of N embedding vector x_i , X in $R^{N*d_{model}}$
- Transformer maps an input sequence (x_1,..., x_N) to an output sequence of the same length (y_1,...,y_N).
- Query(q) is the current focus of attention , which is compared to all of the other preceding inputs:

  $q_i = W^Q x_i$ (6) ; Q =XW^Q (7) where W^Q in $R^{d_{model}*d_k}$, Q in R^{N*d_k}
- Key (k) is a preceding input being compare with the current focus of attention:

  $k_i = W^K x_i$ (8) ; K =XW^K (9) whereW^K in $R^{d_{model}*d_k}$, K in R^{N*d_k}
- Value(v) is used to compute the output for the current focus of attention :

  $v_i = W^V x_i$ (10) ; V =XW^Q (11) $where W^Q$ in $R^{d_{model}*d_v}$, V in R^{N*d_k}
- d_k: dimension of a vector k_i

q_1k_1	q_1k_2	q_1k_3	q_1k_4
q_2k_1	q_2k_2	q_2k_3	q_2k_4
q_3k_1	q_3k_2	q_3k_3	q_3k_4
q_4k_1	q_4k_2	q_4k_3	q_4k_4

Masked

q_1k_1	$-\infty$	$-\infty$	$-\infty$
q_2k_1	q_2k_2	$-\infty$	$-\infty$
q_3k_1	q_3k_2	q_3k_3	
q_4k_1	q_4k_2	q_4k_3	q_4k_4

$$P(u) = \text{softmax}(h_m W_e^T) \qquad (4)$$

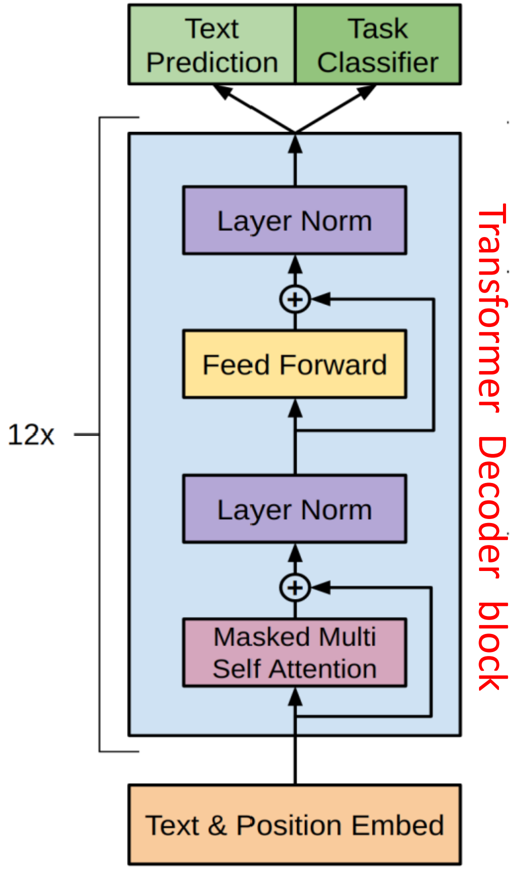
$$h_l = transformer_{block(h_{l-1})} \qquad (3)$$
$$\forall l \in [1, m]$$

Where m is the number of blocks

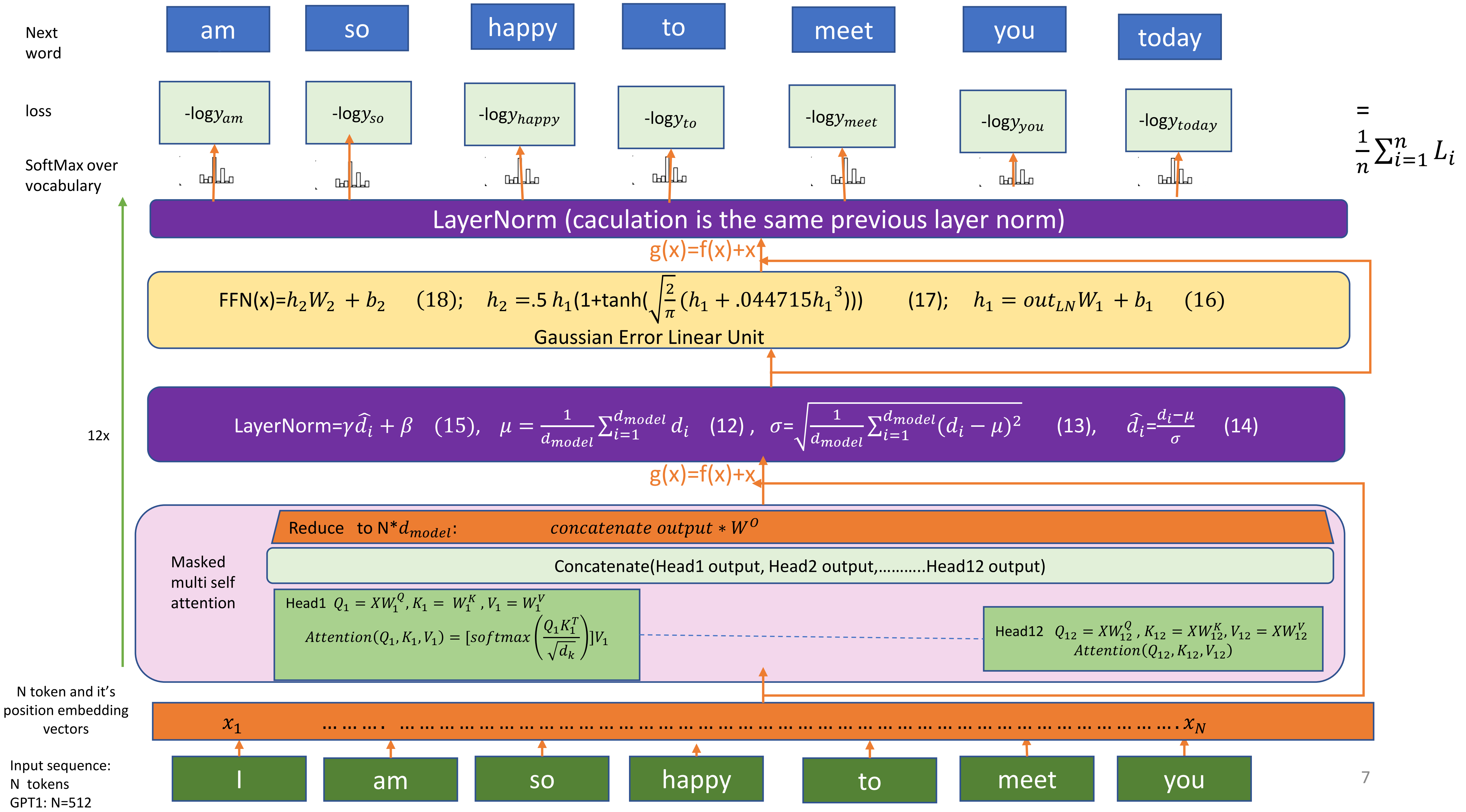
$$h_0 = U w_e + w_P \qquad (2)$$

where U = ($u_{-k}, \ldots, u_{-1}$) is the token context vector , W_e is the token embedding matrix,
and W_p is the position embedding matrix.

GPT1 architecture



"Attention is all you need (Vaswani et al., 2017)"
"Generating wikipedia by summarizing long sequences(Liu et al., 2018)"



Next word
am
so
happy
to
meet
you
today
loss
$-\log y_{am}$
$-\log y_{so}$
$-\log y_{happy}$
$-\log y_{to}$
$-\log y_{meet}$
$-\log y_{you}$
$-\log y_{today}$
$= \frac{1}{n}\sum_{i=1}^{n} L_i$
SoftMax over vocabulary
LayerNorm (caculation is the same previous layer norm)
g(x)=f(x)+x
FFN(x)=$h_2W_2 + b_2$ (18); $h_2 = .5\, h_1(1+\tanh(\sqrt{\frac{2}{\pi}}(h_1 + .044715{h_1}^3)))$ (17); $h_1 = out_{LN}W_1 + b_1$ (16)
Gaussian Error Linear Unit
12x
LayerNorm=$\gamma\hat{d_i} + \beta$ (15), $\mu = \frac{1}{d_{model}}\sum_{i=1}^{d_{model}} d_i$ (12) , $\sigma=\sqrt{\frac{1}{d_{model}}\sum_{i=1}^{d_{model}}(d_i-\mu)^2}$ (13), $\hat{d_i}=\frac{d_i-\mu}{\sigma}$ (14)
g(x)=f(x)+x
Reduce to N*d_{model}: $concatenate\ output * W^O$
Masked multi self attention
Concatenate(Head1 output, Head2 output,..........Head12 output)
Head1 $Q_1 = XW_1^Q, K_1 = W_1^K, V_1 = W_1^V$
$Attention(Q_1,K_1,V_1) = [softmax\left(\frac{Q_1K_1^T}{\sqrt{d_k}}\right)]V_1$
Head12 $Q_{12} = XW_{12}^Q, K_{12} = XW_{12}^K, V_{12} = XW_{12}^V$
$Attention(Q_{12},K_{12},V_{12})$
N token and it's position embedding vectors
x_1 ……… . ……………………………………………………………………………… . x_N
Input sequence: N tokens GPT1: N=512
I
am
so
happy
to
meet
you

Unsupervised Training tuning parameters

- bytepair encoding (BPE) vocabulary with 40,000 merges
- Adam optimization
- learning rate : .25e-3
- Batch size: 64
- Input sequence: 512 tokens
- Dropouts: .01

➔ GPT1 was train with BooksCorpus dataset for 1 month when used 8 P600 GPU system

Overview GPT1: Supervised fine-tuning (step 2)

❖ How supervised fine-tuning works :

- adapt parameters obtained with equation $L_1(T)$ in step 1.
- let a labeled dataset $C=\{c_1, \ldots\ldots\ c_n\}$ has n instances where each instance c_i is an sequence input tokens, $c_i = \{x^1, \ldots, x^q\}$, where y_i is correspond label, $i \in [1,n]$,
- Transform all input sequence
- Compute readable form input

$$P(y_i/c_i) = \text{softmax}\left(h_m^q W_y\right), \qquad (19)$$

h_m^q : the output of the final transformer block

W_y : parameters of linear output layer

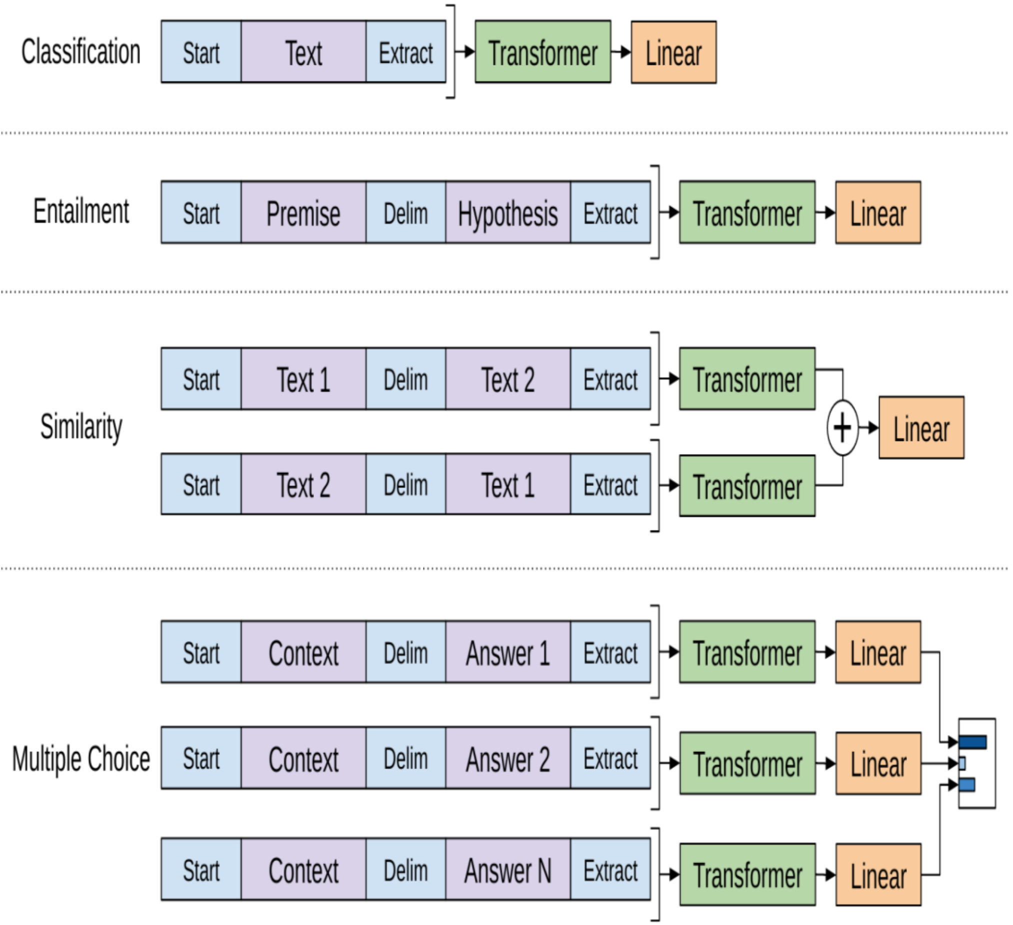
➢ Maximize $L_2(C)$ through supervised training such that :

$$L_2(C) = \sum_{i=1,(c_i,y_i)}^{n} \log P\,(y_i/c_i) \qquad (20)$$

(ci, yi) : all pair of (instance, label) in the label dataset C

- Obtain more generalization and converge faster by compute :

$$L_3(C) = L_2(C) + \lambda * L_1(C) \qquad (21)$$



- Dropout: .1 (classifier)
- learning rate: .625e-4
- Batchsize:32
- linear learning rate decay schedule .02%
- λ =.5

Result: natural language inference tasks, question answering and commonsense reasoning

Method	MNLI-m	MNLI-mm	SNLI	SciTail	QNLI	RTE
ESIM + ELMo (5x)	-	-	89.3	-	-	-
CAFE (5x)	80.2	79.0	89.3	-	-	-
Stochastic Answer Network (3x)	80.6	80.1	-	-	-	-
CAFE	78.7	77.9	88.5	83.3	-	-
GenSen	71.4	71.3	-	-	82.3	59.2
Multi-task BiLSTM + Attn	72.2	72.1	-	-	82.1	**61.7**
Finetuned Transformer LM (GPT1)	**82.1**	**81.4**	**89.9**	**88.3**	**88.1**	56.0

Method	Story Cloze	RACE-m	RACE-h	RACE
val-LS-skip	76.5	-	-	-
Hidden Coherence Model	77.6	-	-	-
Dynamic Fusion Net (9x)	-	55.6	49.4	51.2
BiAttention MRU (9x)	-	60.2	50.3	53.3
Finetuned Transformer LM (GPT1)	**86.5**	**62.9**	**57.4**	**59.0**

Result : Semantic similarity and classification

Method	Classification		Semantic Similarity			GLUE
	CoLA (mcc)	SST2 (acc)	MRPC (F1)	STSB (pc)	QQP (F1)	
Sparse byte mLSTM	-	**93.2**	-	-	-	-
TF-KLD	-	-	**86.0**	-	-	-
ECNU (mixed ensemble)	-	-	-	81.0	-	-
Single-task BiLSTM + ELMo + Attn	35.0	90.2	80.2	55.5	66.1	64.8
Multi-task BiLSTM + ELMo + Attn	18.9	91.6	83.5	72.8	63.3	68.9
Finetuned Transformer LM (GPT1)	**45.4**	91.3	82.3	**82.0**	**70.3**	**72.8**

Semantic similarity and classification results, comparing our model with current state-of-the-art methods. All task evaluations in this table were done using the GLUE benchmark. (mcc= Mathews correlation, acc=Accuracy, pc=Pearson correlation)

$$MCC=\frac{TP.TN-FP.FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

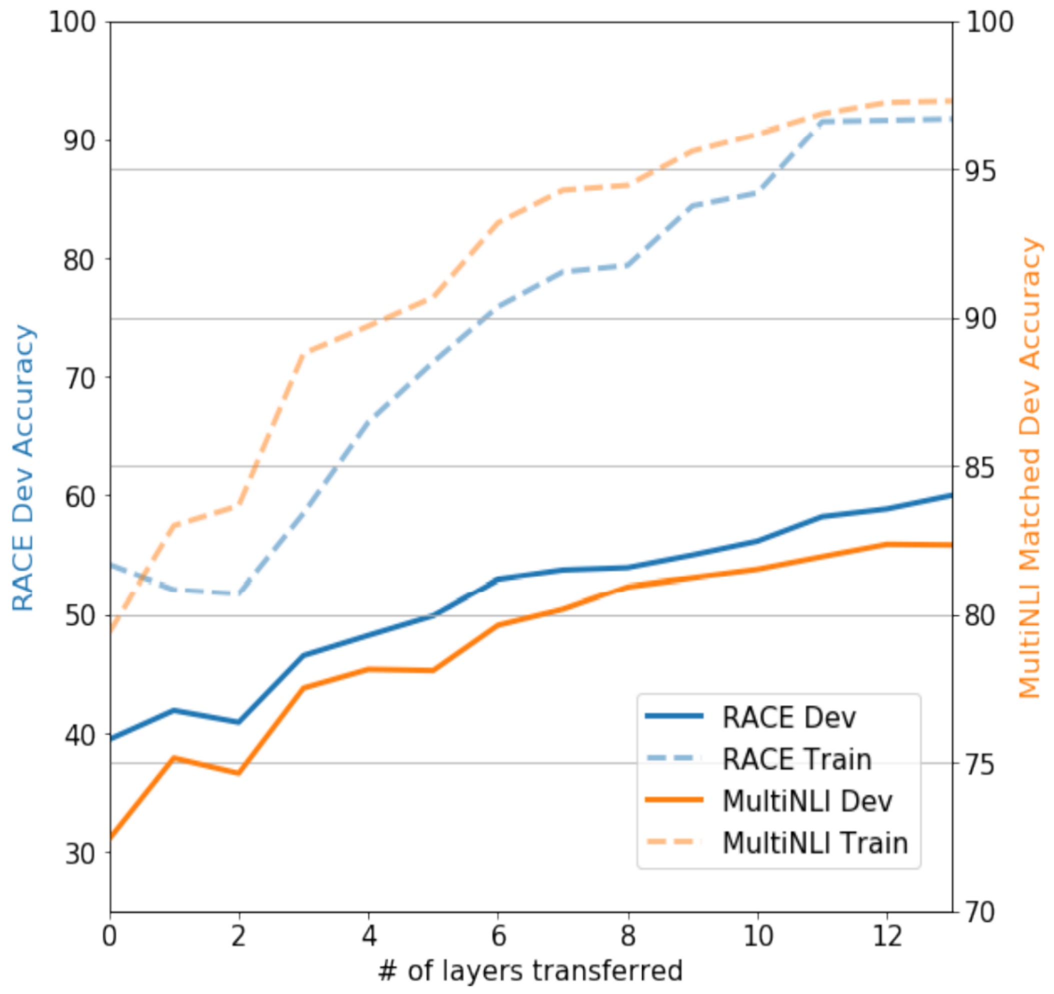
$$ACC=\frac{TP+TN}{TP+TN+FP+FN}$$

$$PCxy=\frac{n.\sum_{i=1}^{n} x_i y_i-(\sum_i^n x_i).(\sum_i^n y)}{\sqrt{n\sum_{i=1}^{n} x_i^2-(\sum_i^n x_i)^2}\cdot\sqrt{n\sum_{i=1}^{n} y_i^2-(\sum_i^n y_i)^2}}$$

$$F1=\frac{2Precision.Recall}{Precision+Recall}=\frac{2TP}{2TP+FP+FN}$$

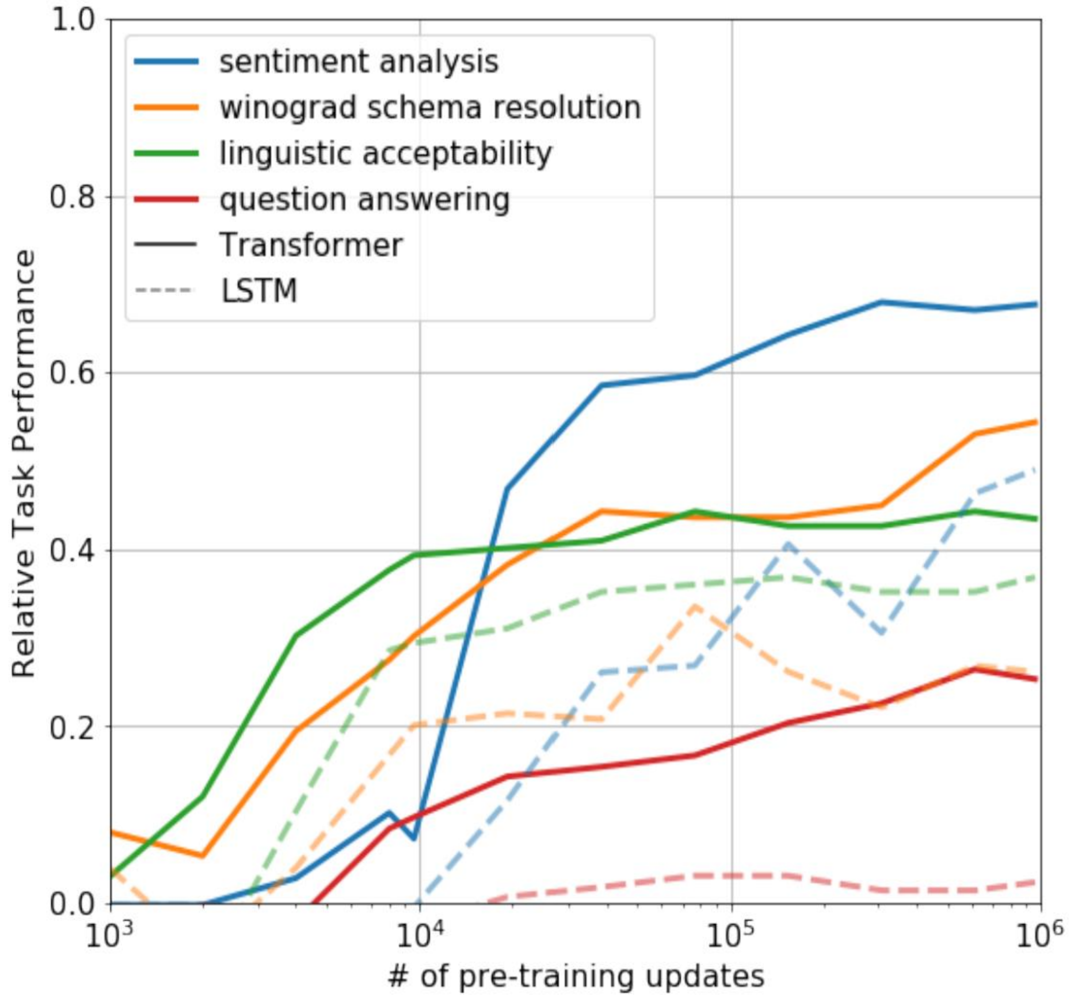
Analysis

Impact of number of layers in pretrain model to specific tasks



[20]

Zero-shot Behaviors



What did GPT1 achieve?

GPT1

- Uses Transformer decoder only
- Learns from 5 GB of text
- Transforms Input for specific task

Result:

- without a change of architecture to perform downstream tasks
- improved the SOTA on 9 of the 12 datasets in 2018

===>set a direction for GPT2

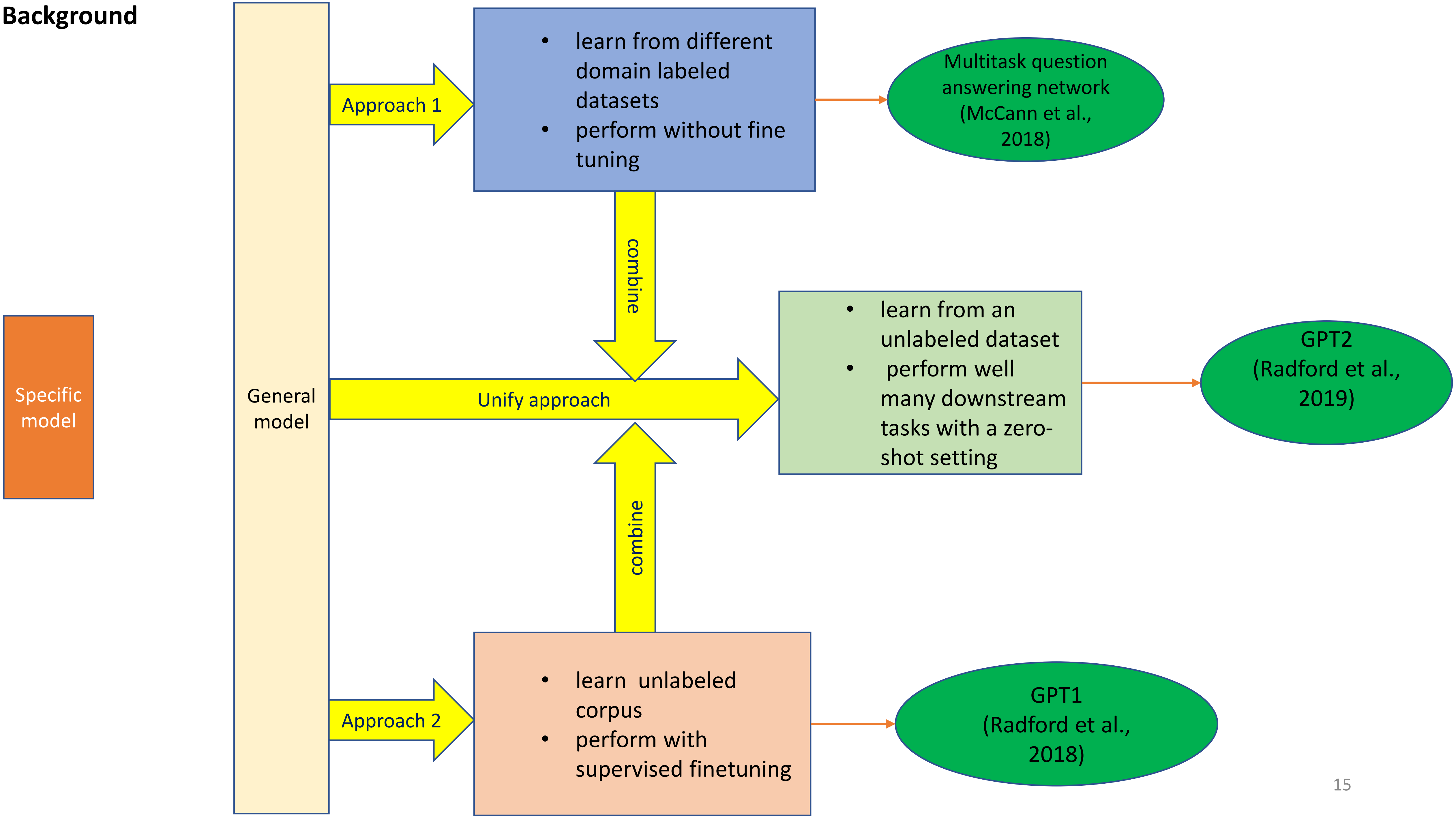
Language Models are Unsupervised Multitask Learners (GPT2)

Published in 2019 by

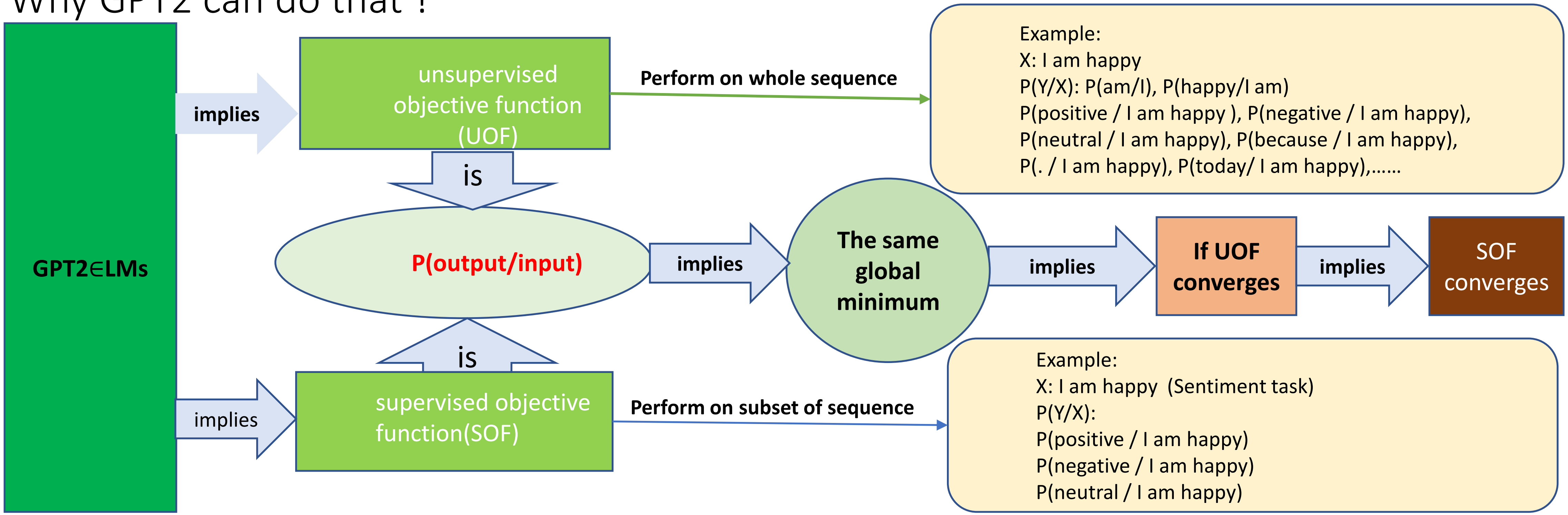
Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever

Present by Tam Doan

Background


Specific model
General model
Approach 1
learn from different domain labeled datasets
perform without fine tuning
Multitask question answering network (McCann et al., 2018)
combine
Unify approach
learn from an unlabeled dataset
perform well many downstream tasks with a zero-shot setting
GPT2 (Radford et al., 2019)
combine
Approach 2
learn unlabeled corpus
perform with supervised finetuning
GPT1 (Radford et al., 2018)

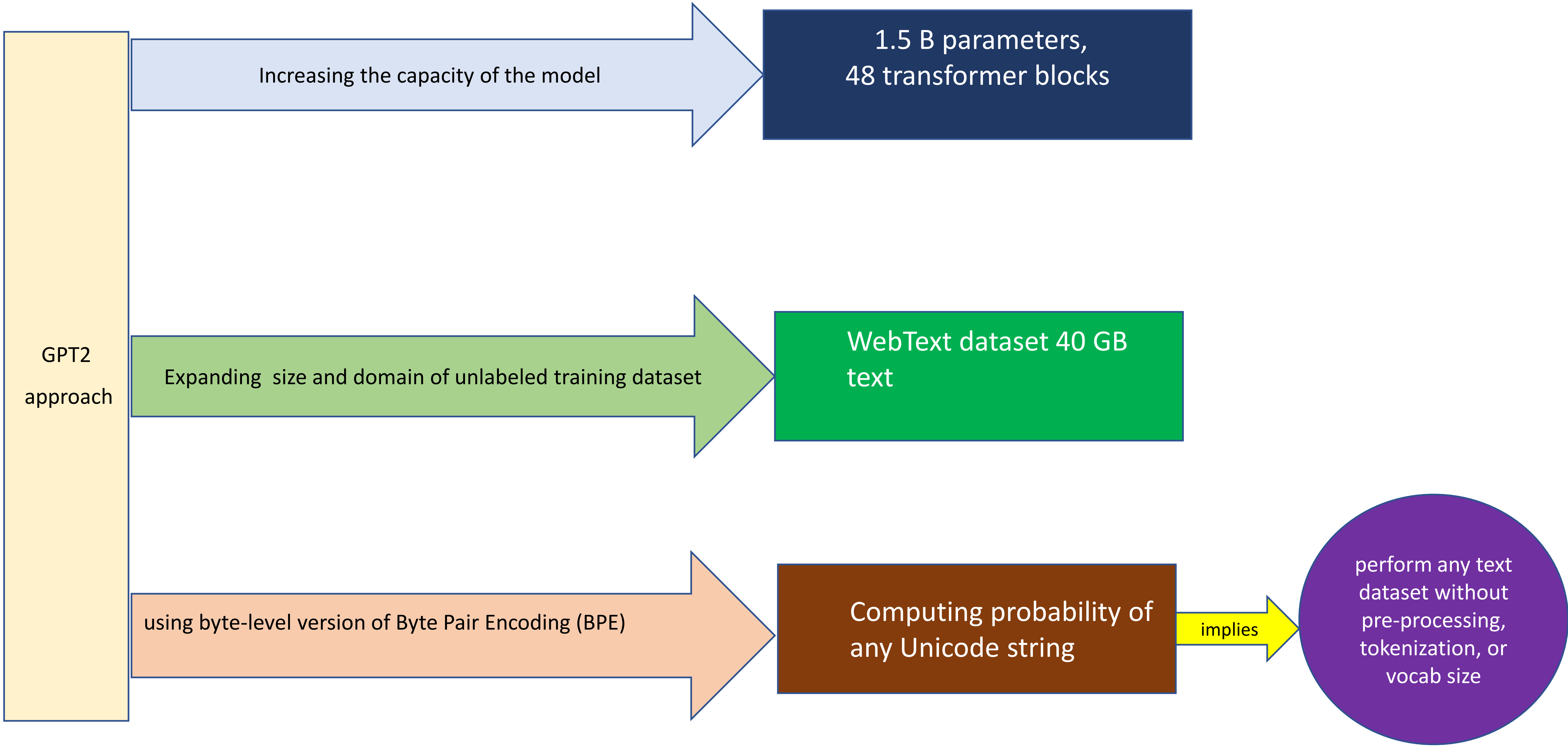
Why GPT2 can do that ?



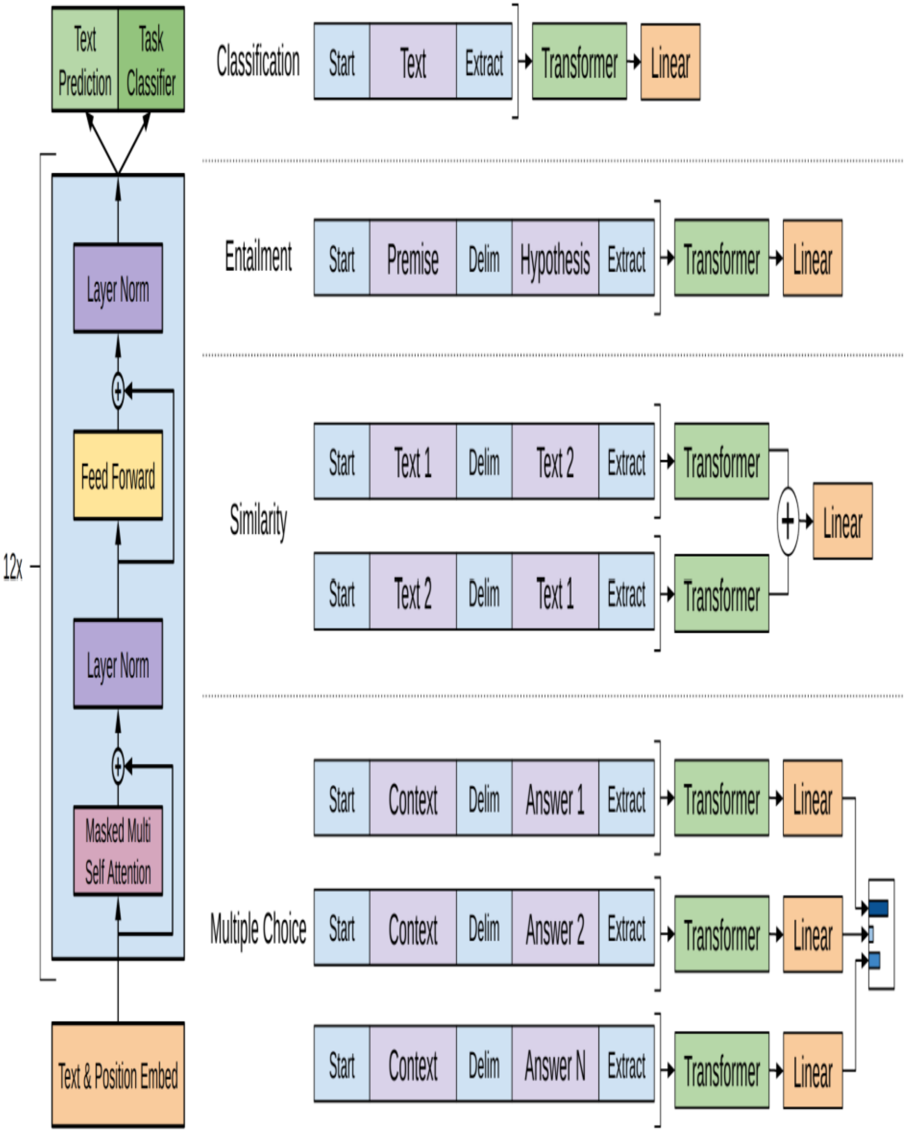
➔if we can optimize the unsupervised objective function to converge, GPT2 can perform downstream tasks without supervised finetuning

GPT1 showed that a large enough language model can perform many NLP tasks with zero setting

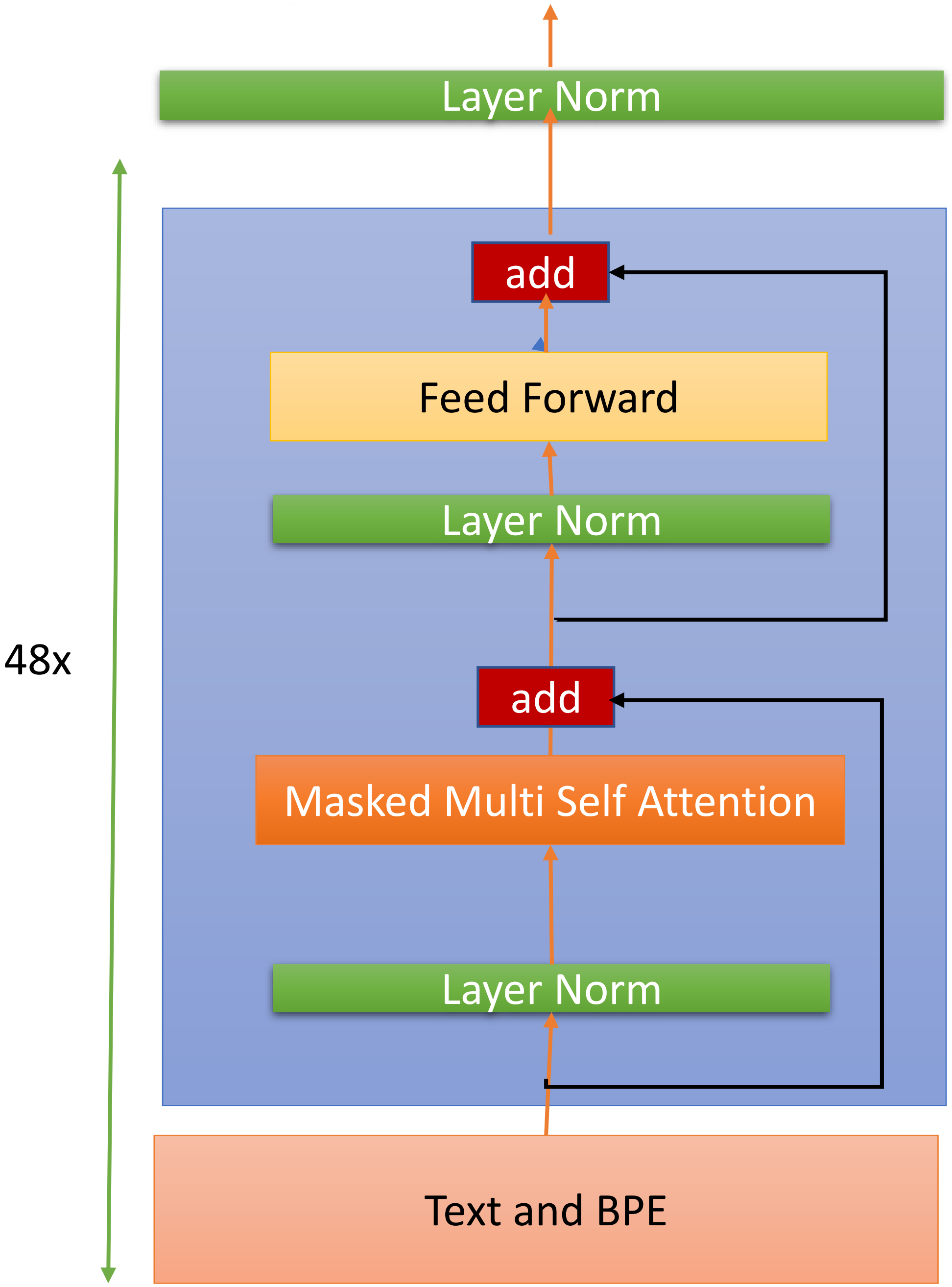
Overview of GPT2



GPT2 Architecture



GPT1 architecture



Parameters	Layers	d_{model}
117M	12	768
345M	24	1024
762M	36	1280
1542M	48	1600

Architecture hyperparameters for the 4 model sizes.

Identity mappings in deep residual networks(He et al., 2016)

GPT2 architecture

Result of 8 datasets

	LAMBADA (PPL)	LAMBADA (ACC)	CBT-CN (ACC)	CBT-NE (ACC)	WikiText2 (PPL)	PTB (PPL)	enwik8 (BPB)	text8 (BPC)	WikiText103 (PPL)	1BW (PPL)
SOTA	99.8	59.23	85.7	82.3	39.14	46.54	0.99	1.08	18.3	**21.8**
117M	**35.13**	45.99	**87.65**	**83.4**	**29.41**	65.85	1.16	1.17	37.50	75.20
345M	**15.60**	55.48	**92.35**	**87.1**	**22.76**	47.33	1.01	**1.06**	26.37	55.72
762M	**10.87**	**60.12**	**93.45**	**88.0**	**19.93**	**40.31**	**0.97**	**1.02**	22.05	44.575
1542M	**8.63**	**63.24**	**93.30**	**89.05**	**18.34**	**35.76**	**0.93**	**0.98**	**17.48**	42.16

$$\text{PPL(W)} = P(w_1, w_{2,\ldots.}w_N)^{\frac{-1}{N}} = \sqrt[N]{\frac{1}{P(w_1, w_{2,\ldots.}w_N)}} = \sqrt[N]{\prod_{i=1}^{N} P(w_i / w_{1,\ldots.}w_{i-1})}$$

Summarization task

- ❑ CNN and Daily Mail dataset
- 3 generated sentences as the summary.

	R-1	R-2	R-L	R-AVG
Bottom-Up Sum	**41.22**	**18.68**	**38.34**	**32.75**
Lede-3	40.38	17.66	36.62	31.55
Seq2Seq + Attn	31.33	11.81	28.83	23.99
GPT-2 TL;DR:	29.34	8.27	26.58	21.40
Random-3	28.78	8.63	25.52	20.98
GPT-2 no hint	21.58	4.03	19.47	15.03

Translation task

- ❑ WMT-14 French-English test set:
  - GPT-2: 11.5 BLEU
  - SOTA unsupervised machine translation (Artetxe et al., 2019): 33.5 BLEU

Commonsense reasoning and Reading Comprehension

❑Winograd Schema challenge dataset: GPT-2 improves SOTA to achieve 70.70%

❑Conversation Question Answering dataset (CoQA) :

GPT2 : 55 F1

supervised SOTA (2018) BERT: 89 F1

Question Answering

Natural Questions dataset
GPT-2: 4.1% correctly

What did GPT2 achieve?

❖ what's new in GPT2 :

- Capacity increased to 1.5 B parameters
- Normalize input before feeding to Attention layer, Feed Forward layer, and linear layer
- using byte-level version of Byte Pair Encoding
- 40 GB unlabeled text of training data

❖Result:

- zero-shots in downstream tasks
- achieved new state of the art on 7 out of 8 language modeling datasets

Language Models are Few-Shot Learners

Published in Jul 2020 by

Tom B. Brown, Benjamin Mann, Nick Ryder,Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam ,Girish Sastry, Amanda Askell ,Sandhini Agarwal ,Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler ,Mateusz Litwin ,Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, Dario Amodei

Present by Tam Doan

Background

The best language model 2019
(GPT2)

Perform poorly on summarization task, question answering task, translation task…

Need improve the accuracy of prediction

- learn from unlabeled various corpus of text
- perform many down stream tasks without finetuning

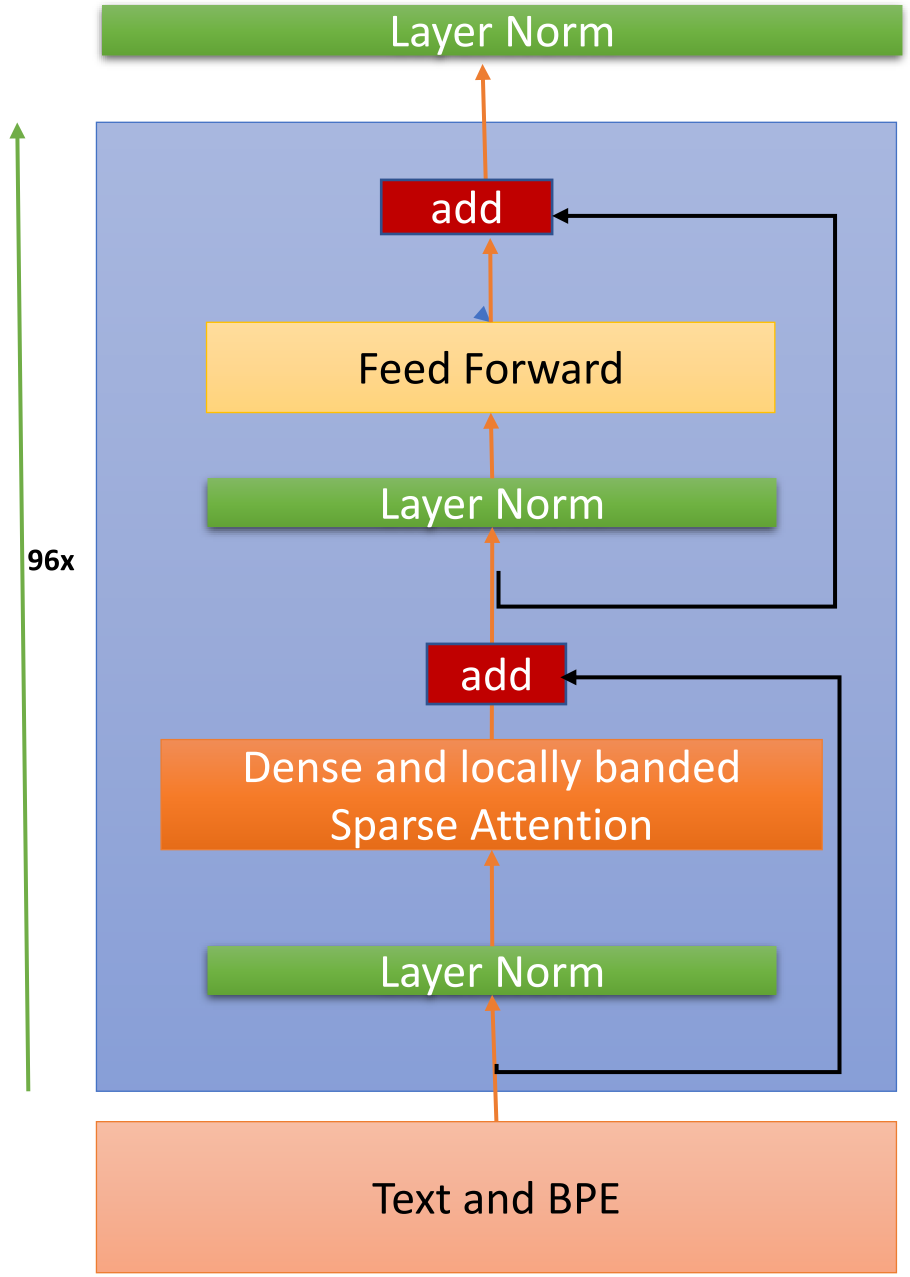
GPT3 (2020)

- Increasing capacity of model
- Growing size and variety of unlabeled training set
- Applying few shot learners

Architecture

- most the same GPT2 except using the Sparse Transformer
- the original self-attention mechanism: $O(n^2)$ with n input tokens.
- Sparse Transformer :$O(n\sqrt{n})$
- 96 transformer blocks
- Each attention layer: 96 attention heads
- Each attention head : 128 dimensions
- Bottleneck layer: 12,288 hidden unit
- The feedforward layer: 49,152 hidden unit

➔175 B learnable parameters



Generating long sequences with sparse transformers(Child et al. 2019)

Different settings for learning

Zero shot learning

Translate English to Vietnamese
Rice =>

One shot learning

Translate English to Vietnamese
Red apple => táo đỏ
Rice =>

Few shot learning

Translate English to Vietnamese
Red apple => táo đỏ
Cashew => hạt điều
Mango => trái xoài
Rice =>

in-context learning

Fine tuning (GPT1)

Stochastic gradient descent

Red apple => táo đỏ

Update gradient

Cashew => hạt điều

Update gradient

. . .

Mango => trái xoài

Update gradient

Rice =>

Training data

Dataset	Quantity (tokens)	Weight in training mix	Epochs elapsed when training for 300B tokens
Common Crawl (filtered)	410 billion	60%	0.44
WebText2	19 billion	22%	2.9
Books1	12 billion	8%	1.9
Books2	55 billion	8%	0.43
Wikipedia	3 billion	3%	3.4

- 570GB after filtering
- 93%English, 7% in other languages.

Training and Hardware detail

- Training :
  - total 300 billion tokens.
  - sample data without replacement
  - Optimizer :Adam with β1 = 0.9, β2 = 0.95, and $\epsilon = 10^{-8}$
  - learning rate: 0.6 × 10^{-4},after 260 billion tokens LR=.1LR
  - Batch size: 3.2M
  - Length of input sequences : 2048 token
  - Trained parallel both matrix multiply and layers of model
- Hardware:
  - Microsoft high-bandwidth cluster (V100 GPU)

Result: traditional language modeling tasks, Cloze tasks, sentence , paragraph completion tasks

- Penn Tree Bank (PTB) dataset : GPT3 achieved new SOTA with a perplexity of 20.50 (increase 15 points) in zero-shot.

Setting	LAMBADA (acc)	LAMBADA (ppl)	StoryCloze (acc)	HellaSwag (acc)
SOTA	68.0	8.63	**91.8**	**85.6**
GPT-3 Zero-Shot	**76.2**	3.00	83.2	78.9
GPT-3 One-Shot	**72.5**	3.35	84.7	78.1
GPT-3 Few-Shot	**86.4**	**1.92**	87.7	79.3

Setting	NaturalQS	WebQS	TriviaQA
RAG (Fine-tuned, Open-Domain)	**44.5**	**45.5**	**68.0**
T5-11B+SSM (Fine-tuned, Closed-Book)	36.6	44.7	60.5
T5-11B (Fine-tuned, Closed-Book)	34.5	37.4	50.1
GPT-3 Zero-Shot	14.6	14.4	64.3
GPT-3 One-Shot	23.0	25.3	**68.0**
GPT-3 Few-Shot	29.9	41.5	**71.2**

Result : Closed Book Question Answering tasks

Result: translation tasks

Setting	WMT’14: En→Fr	WMT’14: Fr→En	WMT’16: En→De	WMT’16: De→En	WMT’16: En→Ro	WMT’16 : Ro→En
SOTA (Supervised)	**45.6**	35.0	**41.2**	40.2	**38.5**	**39.9**
XLM	33.4	33.3	26.4	34.3	33.3	31.8
MASS	37.5	34.9	28.3	35.2	35.2	33.1
mBART	-	-	29.8	34.0	35.0	30.5
GPT-3 Zero-Shot	25.2	21.2	24.6	27.2	14.1	19.9
GPT-3 Few-Shot	28.3	33.7	26.2	30.4	20.6	38.6
GPT-3 Few-Shot	32.6	**39.2**	29.7	**40.6**	21.0	39.5

Setting	Winograd	Winogrande (XL)
Fine-tuned SOTA	**90.1**	**84.6**
GPT-3 Zero-Shot	88.3	70.2
GPT-3 One-Shot	89.7	73.2
GPT-3 Few-Shot	88.6	77.7

Result: Winograd-Style Tasks

Result: Common Sense Reasoning and Reading Comprehension

Setting	PIQA	ARC (Easy)	ARC (hard)	OpenBookQA
Fine-tuned SOTA	79.4	**92.0**	**78.5**	**87.2**
GPT-3 Zero-Shot	**80.5***	68.8	51.4	57.6
GPT-3 One-Shot	**80.5***	71.2	53.2	58.8
GPT-3 Few-Shot	**82.8***	70.1	51.5	65.4

Setting	CoQA	DROP	QuAC	SQuADv2	RACE-h	RACE-m
Fine-tuned SOTA	**90.7**[a]	**89.1**[b]	**74.4**[c]	**93.0**[d]	**90.0**[e]	**93.1**[e]
GPT-3 Zero-Shot	81.5	23.6	41.5	59.5	45.5	58.4
GPT-3 One-Shot	84.0	34.3	43.3	65.4	45.9	57.4
GPT-3 Few-Shot	85.0	36.5	44.3	69.8	46.8	58.1

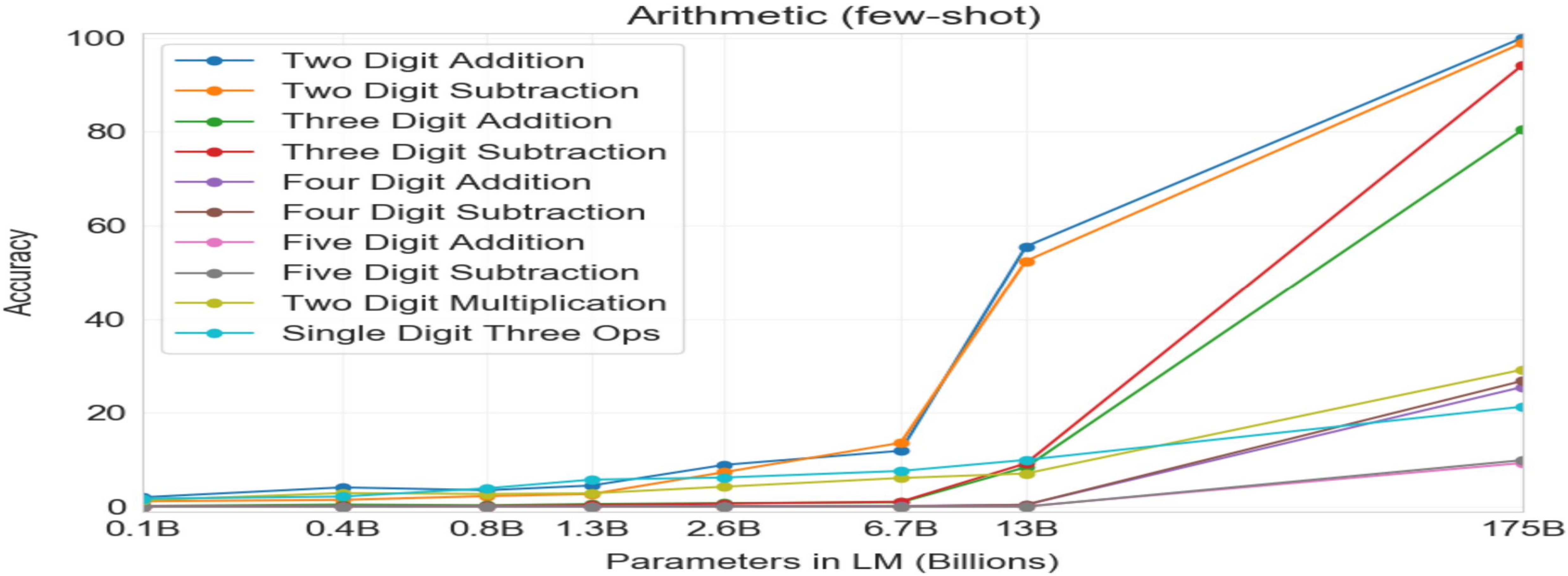
Result: SuperGLUE and Natural Language Inference

	SuperGLUE Average	BoolQ Accuracy	CB Accuracy	CB F1	COPA Accuracy	RTE Accuracy
Fine-tuned SOTA	**89.0**	**91.0**	**96.9**	**93.9**	**94.8**	**92.5**
Fine-tuned BERT-Large	69.0	77.4	83.6	75.7	70.6	71.7
GPT-3 Few-Shot	71.8	76.4	75.6	52.0	92.0	69.0

	WiC Accuracy	WSC Accuracy	MultiRC Accuracy	MultiRC F1a	ReCoRD Accuracy	ReCoRD F1
Fine-tuned SOTA	**76.1**	**93.8**	**62.3**	**88.2**	**92.5**	**93.3**
Fine-tuned BERT-Large	69.6	64.6	24.1	70.0	71.3	72.0
GPT-3 Few-Shot	49.4	80.1	30.5	75.4	90.2	91.1

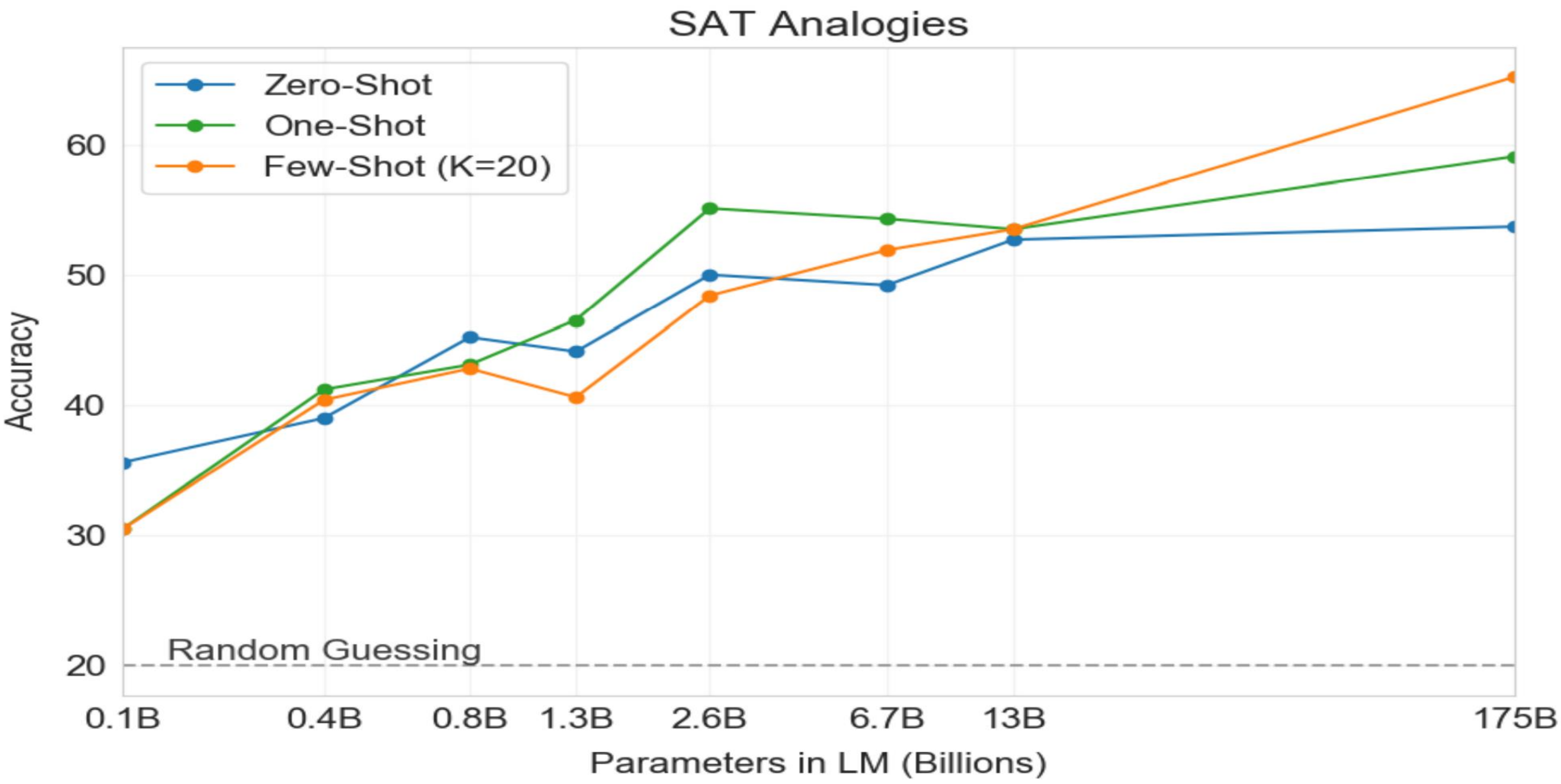
All results are reported on the test set. GPT-3 few-shot is given a total of 32 examples

Arithmetic tasks



Result : SAT Analogies

- Dataset includes 374 “Scholastic Aptitude Test(SAT) analogy” problems



News Article Generation

80 US people selected:

1. “very likely written by a human”,
2. “more likely written by a human”,
3. “I don’t know”,
4. “more likely written by a machine”,
5. “very likely written by a machine”

	Mean accuracy	**95% Confidence Interval (low, hi)**	**t compared to control (p-value)**	**“I don’t know” assignments**
Control model	86%	83%–90%	-	3.6 %
GPT-3 175B Few shot	52%	49%–54%	16.9 (1e-34)	7.8%

Result in 200 word news articles generating

	Mean accuracy	**95% Confidence Interval (low, hi)**	**t compared to control (p-value)**	**t compared to control (p-value**	**“I don’t know” assignments**
Control model	88%	84%–91%		-	2.7%
GPT-3 175B Few shot	52%	48%–57%		12.7 (3.2e-23)	10.6%

Result in 500 word news articles generating

- ❑Learning and Using Novel Words tasks
- ❑Correcting English Grammar tasks

What did GPT3 achieve ?

- mechanisms new GPT3:
  - ✓Sparse transformer
  - ✓570 Gb
  - ✓In context learning
- GPT3 is SOTA language model in 2020
- Social impact:
  - Positive impact: code and writing auto-completion, grammar assistance, game narrative generation, improving search engine responses, chatbots, and language education
  - Negative impact:
    - GPT3 is not equal gender identified.
    - GPT3 associate more to some race and religion
    - Tool for hacker

Discussion

Time

GPT1 2018
- Transformers decoding only architecture
- Parameter: 117 M; Layers:12; d_{model}: 768
- Training data: 5 GB
- Perform downstream tasks: input transformations and supervised fine tune
- Input text sequence: n=512
- Output: text

GPT2 2019
- Similar GPT1 architecture except normalize input before feeding to Attention layer, Feed Forward layer, and linear layer
- Parameter: 1.5B; Layers:48; d_{model}: 1600
- Training data: 40 GB WebText dataset
- Perform downstream tasks: zero shot setting
- Input text sequence: n=1024
- Output: text

GPT3 2020
- Similar GPT2 architecture except to use combination of dense and locally banded sparse attention in transformer blocks
- Parameter: 175B; Layers:96; d_{model}: 12,288
- Training data: 570GB text
- Perform downstream tasks: zero shot, one shot, few shot
- Input text sequence: n=2048
- Output: text

InstructGPT Jan 2022
- GPT3 +RFHF
- Input: text
- Output: text

ChatGPT Nov 2022
- GPT3 +RFHF
- Input: text
- Output: text

GPT 4 Mar 2023

GPT4
- Input: text +image
- Output: text

❑ Positive Social impact of chatGPT and later versions
- Providing a powerful tool for analyzing , understanding , and learning many topics
- New way to create process
- Available to Microsoft Bing's users
- Competition between big tech company: Google: Bart, Baidu: Ernie Bot, Facebook:LLaMA
- Support teaching , office work, theraphy chat ….
- Apply to medical field to save life

❑ Negative Social impact of chatGPT and later versions
- Need billions of dollars to build and train
- Cost more than $100,000 to run chatGPT per day
- Require access to internet to use
- Use for bad purposes : hackers, Plagiarism,..
- Affect the labor market, increase unemployment ….

❑ What is the approach of the future?
- Can we apply GPT4 to medical field to save life when it can learn from both text and images?
- How can we help 3 billion people without internet benefit from the development of NLP?
- Should we spend more billions of dollars for bigger model?
- Another method with the same performance , lower cost , and more friendly with environment is future research?

Reference

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, *33*, 1877-1901.

Child, R., Gray, S., Radford, A., & Sutskever, I. (2019). Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*.

Dai, A. M., & Le, Q. V. (2015). Semi-supervised sequence learning. *Advances in neural information processing systems*, *28*.

Eisenstein, Jacob. *Introduction to natural language processing*. MIT press, 2019.

He, K., Zhang, X., Ren, S., and Sun, J. Identity mappings in deep residual networks. In European conference on computer vision, pp. 630–645. Springer, 2016.

Liu, P. J., Saleh, M., Pot, E., Goodrich, B., Sepassi, R., Kaiser, L., & Shazeer, N. (2018). Generating wikipedia by summarizing long sequences. *arXiv preprint arXiv:1801.10198*.

McCann, B., Keskar, N. S., Xiong, C., and Socher, R. The natural language decathlon: Multitask learning as question answering. arXiv preprint arXiv:1806.08730, 2018

OpenAI. GPT-4 Technical Report. arXiv:2303.08774

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. Improving language understanding by generative pre-training. 2018.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, *1*(8), 9.

Sennrich, R., Haddow, B., and Birch, A. Neural machine translation of rare words with subword units. arXiv preprint arXiv:1508.07909, 2015.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Thank you so much for your help ！

Thank you so much for your time ！

Thank you so much for being here ！

Thank you so much for your attention ！

Have a great weekend ☺