PaLM: Scaling Language Modeling with Pathways

Aakanksha Chowdhery* Sharan Narang* Jacob Devlin*
Maarten Bosma Gaurav Mishra Adam Roberts Paul Barham
Hyung Won Chung Charles Sutton Sebastian Gehrmann Parker Schuh Kensen Shi
Sasha Tsvyashchenko Joshua Maynez Abhishek Rao† Parker Barnes Yi Tay
Noam Shazeer‡ Vinodkumar Prabhakaran Emily Reif Nan Du Ben Hutchinson
Reiner Pope James Bradbury Jacob Austin Michael Isard Guy Gur-Ari
Pengcheng Yin Toju Duke Anselm Levskaya Sanjay Ghemawat Sunipa Dev
Henryk Michalewski Xavier Garcia Vedant Misra Kevin Robinson Liam Fedus
Denny Zhou Daphne Ippolito David Luan‡ Hyeontaek Lim Barret Zoph
Alexander Spiridonov Ryan Sepassi David Dohan Shivani Agrawal Mark Omernick
Andrew M. Dai Thanumalayan Sankaranarayana Pillai Marie Pellat Aitor Lewkowycz
Erica Moreira Rewon Child Oleksandr Polozov† Katherine Lee Zongwei Zhou
Xuezhi Wang Brennan Saeta Mark Diaz Orhan Firat Michele Catasta† Jason Wei
Kathy Meier-Hellstern Douglas Eck Jeff Dean Slav Petrov Noah Fiedel

Google Research

Nilesh
2022.06.17

Background

- Trying to mitigate the challenge of finetuning

Model	# of Parameters (in billions)	Accelerator chips	Model FLOPS utilization
GPT-3	175B	V100	21.3%
Gopher	280B	4096 TPU v3	32.5%
Megatron-Turing NLG	530B	2240 A100	30.2%
PaLM	540B	6144 TPU v4	46.2%

- Model improvement happened in Auto-Regressive models due to:
  - scaling the size of the models in both depth and width
  - increasing the number of tokens that the model was trained on
  - training on cleaner datasets from more diverse sources
  - increasing model capacity without increasing the computational cost through sparsely activated module
    s

Model - Architecture

- Transformer model architecture with decoder-only setup
- SwiGLU Activations
- Parallel Layers
  - y = x + MLP(LayerNorm(x)) + Attention(LayerNorm(x))
  - 15% faster-training speed at large scales
- Multi Query Attention
- Rotary Position Embedding is a type of position embedding which encodes absolute positional information with a rotation matrix and naturally incorporates explicit relative position dependency in the self-attention fo rmulation
- Shared Input-Output Embedding matrices
- No Biases
- SentencePiece vocabulary with 256k tokens

Model - Hyperparameters

Model	Layers	# of Heads	d_{model}	# of Parameters (in billions)	Batch Size
PaLM 8B	32	16	4096	8.63	$256 \to 512$
PaLM 62B	64	32	8192	62.50	$512 \to 1024$
PaLM 540B	118	48	18432	540.35	$512 \to 1024 \to 2048$

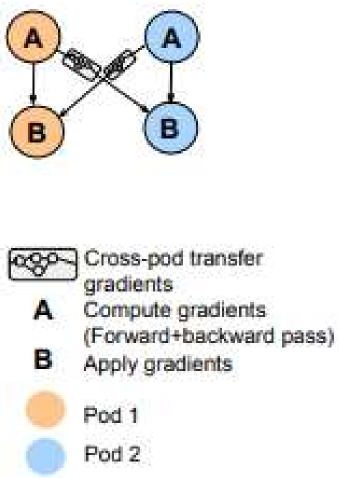
Table 1: Model architecture details. We list the number of layers, d_{model}, the number of attention heads and attention head size. The feed-forward size d_{ff} is always $4 \times d_{\text{model}}$ and attention head size is always 256.

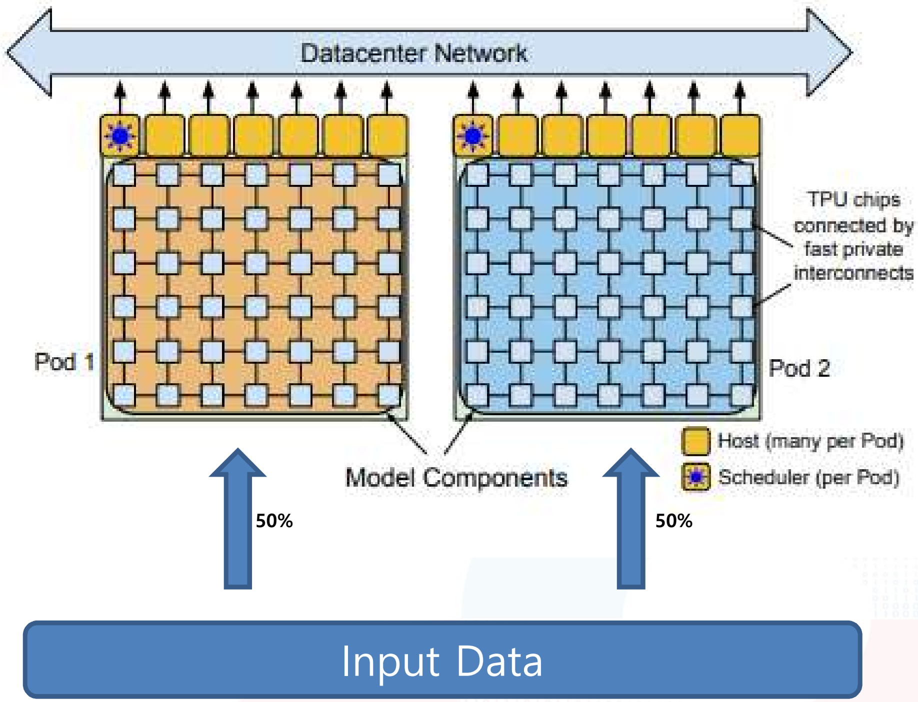
Training - Dataset

Total dataset size = 780 billion tokens	
Data source	Proportion of data
Social media conversations (multilingual)	50%
Filtered webpages (multilingual)	27%
Books (English)	13%
GitHub (code)	5%
Wikipedia (multilingual)	4%
News (English)	1%

Training - Infrastructure

- Two TPU v4 Pods
- 3072 TPU v4 chips in each Pod





Training - Setup

- **Weight initialization:** Used "fan-in variance scaling" i.e., W ~ N (0, $\frac{1}{\sqrt{n_{in}}}$)

- **Optimizer:** Adafactor without factorization

- **Optimization hyperparameters:** Adafactor learning rate of 10^{-2} for the first 10,000 steps, then 1/ √ k, where k is the step number
  β1 = 0.9 β2 = 1 − $k^{-0.8}$

- **Loss function:** standard language modeling loss function, which is the average log probability of all tokens without label smoothing, additionally use an auxiliary loss of z loss = $10^{-4} \cdot \log^2 Z$ to encourage the softmax normalizer log(Z) to be close to 0

- **Sequence length:** 2048 with Input examples concatenation and eod tokens

- **Bitwise determinism:** The model is fully bitwise reproducible from any checkpoint

- **Dropout:** The model was trained without dropout, although dropout of 0.1 is used for finetuning in most cases.

Training - Instability

- Observed spikes in the loss roughly 20 times during training
- Spikes occurred at highly irregular intervals
- To mitigate these spikes, re-started training from a checkpoint roughly 100 steps before the spike started, and skipped roughly 200–500 data batches

Evaluation – English NLP tasks

KISTI SINCE 1962

Task	0-shot		1-shot		Few-shot	
	Prior SOTA	PaLM 540B	Prior SOTA	PaLM 540B	Prior SOTA	PaLM 540B
TriviaQA (EM)	71.3[a]	**76.9**	75.8[a]	**81.4**	75.8[a] (1)	**81.4** (1)
Natural Questions (EM)	**24.7**[a]	21.2	26.3[a]	**29.3**	32.5[a] (1)	**39.6** (64)
Web Questions (EM)	**19.0**[a]	10.6	**25.3**[b]	22.6	41.1[b] (64)	**43.5** (64)
Lambada (EM)	77.7[f]	**77.9**	80.9[a]	**81.8**	87.2[c] (15)	**89.7** (8)
HellaSwag	80.8[f]	**83.4**	80.2[c]	**83.6**	82.4[c] (20)	**83.8** (5)
StoryCloze	83.2[b]	**84.6**	84.7[b]	**86.1**	87.7[b] (70)	**89.0** (5)
Winograd	88.3[b]	**90.1**	**89.7**[b]	87.5	88.6[a] (2)	**89.4** (5)
Winogrande	74.9[f]	**81.1**	73.7[c]	**83.7**	79.2[a] (16)	**85.1** (5)
Drop (F1)	57.3[a]	**69.4**	57.8[a]	**70.8**	58.6[a] (2)	**70.8** (1)
CoQA (F1)	**81.5**[b]	77.6	**84.0**[b]	79.9	**85.0**[b] (5)	81.5 (5)
QuAC (F1)	41.5[b]	**45.2**	43.4[b]	**47.7**	44.3[b] (5)	**47.7** (1)
SQuADv2 (F1)	71.1[a]	**80.8**	71.8[a]	**82.9**	71.8[a] (10)	**83.3** (5)
SQuADv2 (EM)	64.7[a]	**75.5**	66.5[a]	**78.7**	67.0[a] (10)	**79.6** (5)
RACE-m	64.0[a]	**68.1**	65.6[a]	**69.3**	66.9[a†] (8)	**72.1** (8)
RACE-h	47.9[c]	**49.1**	48.7[a]	**52.1**	49.3[a†] (2)	**54.6** (5)

Task	0-shot		1-shot		Few-shot	
	Prior SOTA	PaLM 540B	Prior SOTA	PaLM 540B	Prior SOTA	PaLM 540B
PIQA	82.0[c]	**82.3**	81.4[a]	**83.9**	83.2[c] (5)	**85.2** (5)
ARC-e	76.4[e]	**76.6**	76.6[a]	**85.0**	80.9[e] (10)	**88.4** (5)
ARC-c	51.4[b]	**53.0**	53.2[b]	**60.1**	52.0[a] (3)	**65.9** (5)
OpenbookQA	**57.6**[b]	53.4	**55.8**[b]	53.6	65.4[b] (100)	**68.0** (32)
BoolQ	83.7[f]	**88.0**	82.8[a]	**88.7**	84.8[c] (32)	**89.1** (8)
Copa	91.0[b]	**93.0**	**92.0**[a]	91.0	93.0[a] (16)	**95.0** (5)
RTE	**73.3**[e]	72.9	71.5[a]	**78.7**	76.8 (5)	**81.2** (5)
WiC	50.3[a]	**59.1**	52.7[a]	**63.2**	58.5[c] (32)	**64.6** (5)
Multirc (F1a)	73.7[a]	**83.5**	74.7[a]	**84.9**	77.5[a] (4)	**86.3** (5)
WSC	85.3[a]	**89.1**	83.9[a]	**86.3**	85.6[a] (2)	**89.5** (5)
ReCoRD	90.3[a]	**92.9**	90.3[a]	**92.8**	90.6 (2)	**92.9** (2)
CB	48.2[a]	**51.8**	73.2[a]	**83.9**	84.8[a] (8)	**89.3** (5)
ANLI R1	39.2[a]	**48.4**	42.4[a]	**52.6**	44.3[a] (2)	**56.9** (5)
ANLI R2	39.9[e]	**44.2**	40.0[a]	**58.7**	41.2[a] (10)	**56.1** (5)
ANLI R3	41.3[a]	**45.7**	40.8[a]	**52.3**	44.7[a] (4)	**51.2** (5)

Model	Avg NLG	Avg NLU
GPT-3 175B	52.9	65.4
GLaM 64B/64E	58.4	68.7
PaLM 8B	41.5	59.2
PaLM 62B	57.7	67.3
PaLM 540B	63.9	74.7

Evaluation - Finetuning

Results on the SuperGLUE dev set

Encoder-Decoder Model

Model	Avg	BoolQ	CB	CoPA	MultiRC	Record	RTE	WiC	WSC
T5-11B	89.9	90.8	94.9/96.4	98.0	87.4/66.1	93.8/93.2	93.9	77.3	96.2
ST-MoE-32B	93.2	93.1	100/100	100	90.4/69.9	95.0/95.6	95.7	81.0	100
PaLM 540B (*finetuned*)	92.6	92.2	100/100	100	90.1/69.2	94.0/94.6	95.7	78.8	100

Results on SuperGLUE dev set comparing PaLM-540B few-shot and finetuned

Model	BoolQ	CB	CoPA	MultiRC	Record	RTE	WiC	WSC
Few-shot	89.1	89.3	95	86.3/-	92.9/-	81.2	64.6	89.5
Finetuned	92.2	100/100	100	90.1/69.2	94.0/94.6	95.7	78.8	100

Evaluation – Big Bench

- **goal step wikihow -** The goal is to reason about the goal-step relationship between events.
  Input: In order to "clean silver," which step should be done first?
  (a) dry the silver (b) handwash the silver
  Answer: (b) handwash the silver

- **logical args –** The goal is to predict the correct logical inference from a passage.
  Input: Students told the substitute teacher they were learning trigonometry. The substitute told them that instead of teaching them useless facts about triangles, he would instead teach them how to work with probabilities. What is he implying?

(a) He believes that mathematics does not need to be useful to be interesting.
(b) He thinks understanding probabilities is more useful than trigonometry.
(c) He believes that probability theory is a useless subject.

Answer: (b) He thinks understanding probabilities is more useful than trigonometry.

Evaluation – Big Bench

- **english proverbs –** The goal is to guess which proverb best describes a text passage.
  Input: Vanessa spent lots of years helping out on weekends at the local center for homeless aid. Recentl y, when she lost her job, the center was ready to offer her a new job right away. Which of the following proverbs best apply to this situation?
  (a) Curses, like chickens, come home to roost.
  (b) Where there is smoke there is fire
  (c) As you sow, so you shall reap
  Answer: (c) As you sow, so you shall reap

- **logical sequence –** The goal is to order a set of "things" (months, actions, numbers, letters, etc.) into thei r logical ordering
  Input: Which of the following lists is correctly ordered chronologically?
  (a) drink water, feel thirsty, seal water bottle, open water bottle
  (b) feel thirsty, open water bottle, drink water, seal water bottle
  (c) seal water bottle, open water bottle, drink water, feel thirsty

Answer: (b) feel thirsty, open water bottle, drink water, seal water bottle

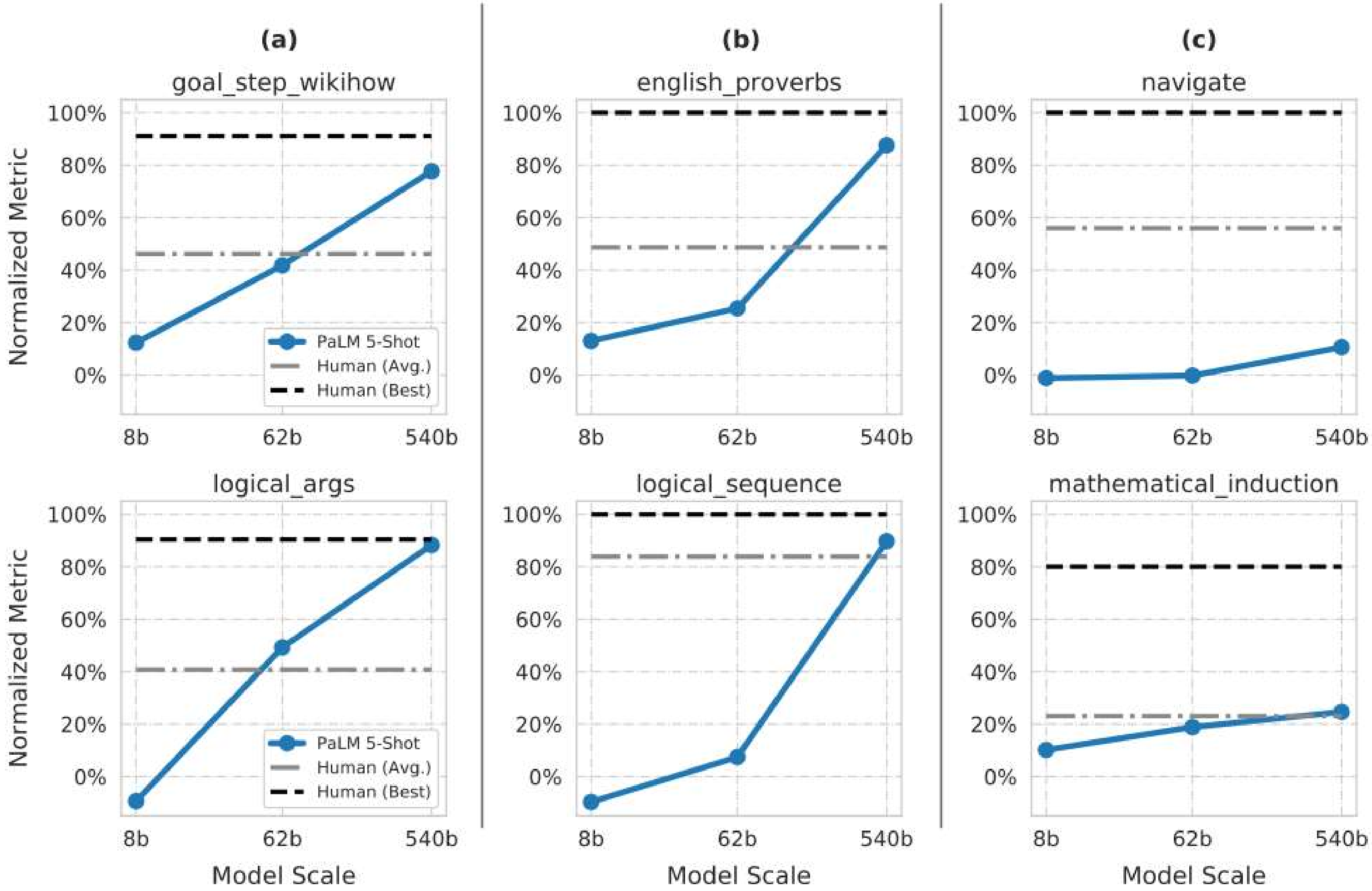
Evaluation – Big Bench

- **navigate –** The goal is to follow a set of simple navigational instructions, and figure out where you woul d end up.
  Input: If you follow these instructions, do you return to the starting point? Always face forward. Take 6 steps left. Take 7 steps forward. Take 8 steps left. Take 7 steps left. Take 6 steps forward. Take 1 ste p forward. Take 4 steps forward.
  Answer: No

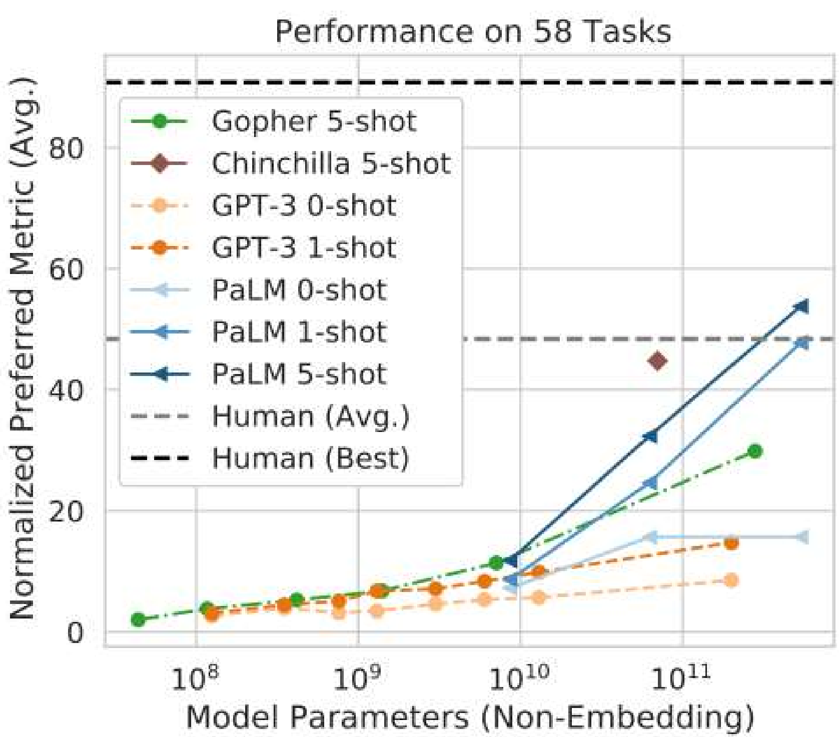
- **mathematical induction –** The goal is to perform logical inference mathematical induction rules, even if they contradict real-world math.
  Input: It is known that adding 2 to any odd integer creates another odd integer. 2 is an odd integer. The refore, 6 is an odd integer. Is this a correct induction argument (even though some of the assumpti ons may be incorrect)?

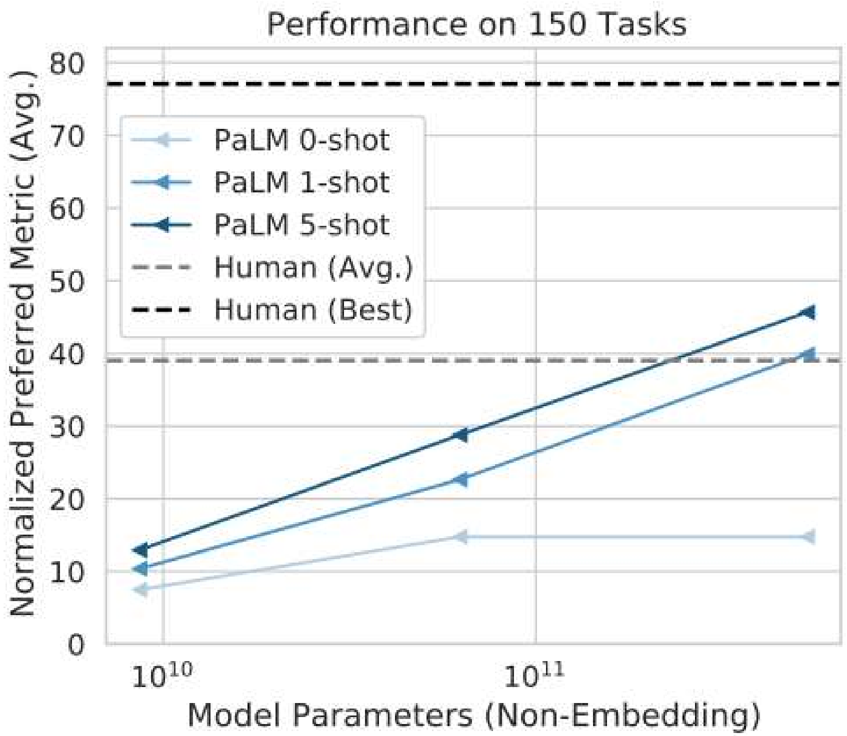
Answer: Yes

Evaluation – Big Bench

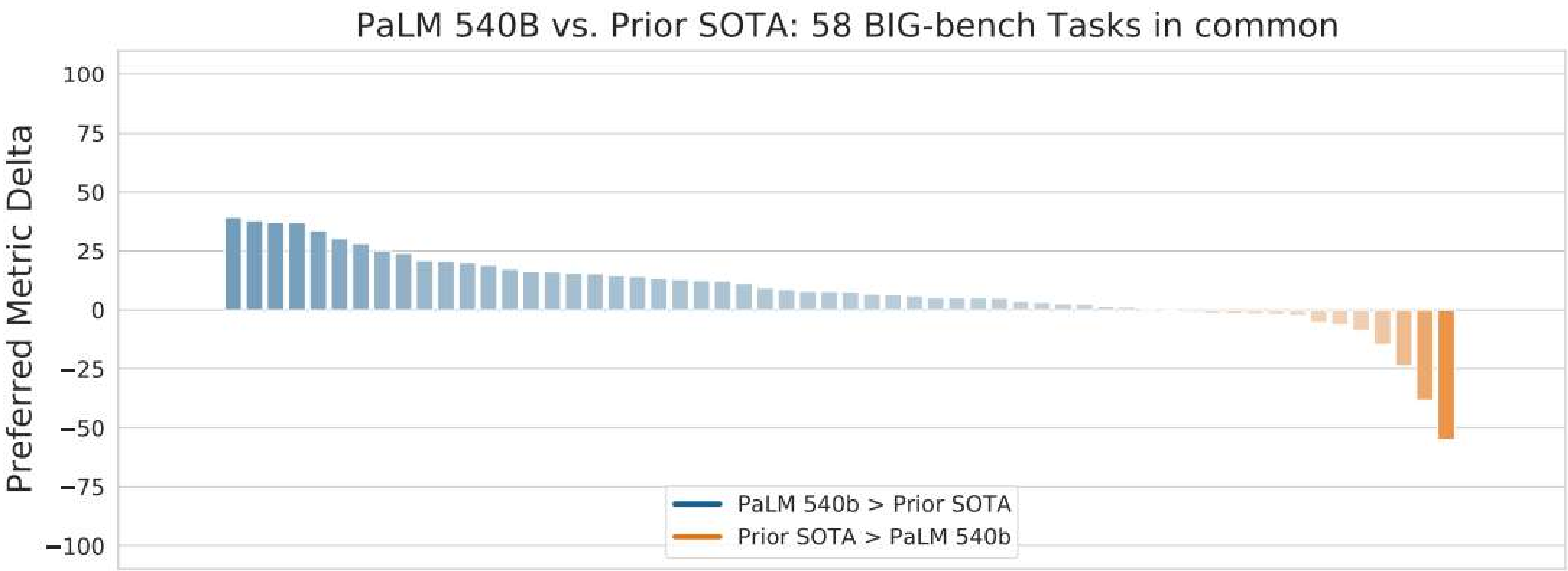


Evaluation – Big Bench



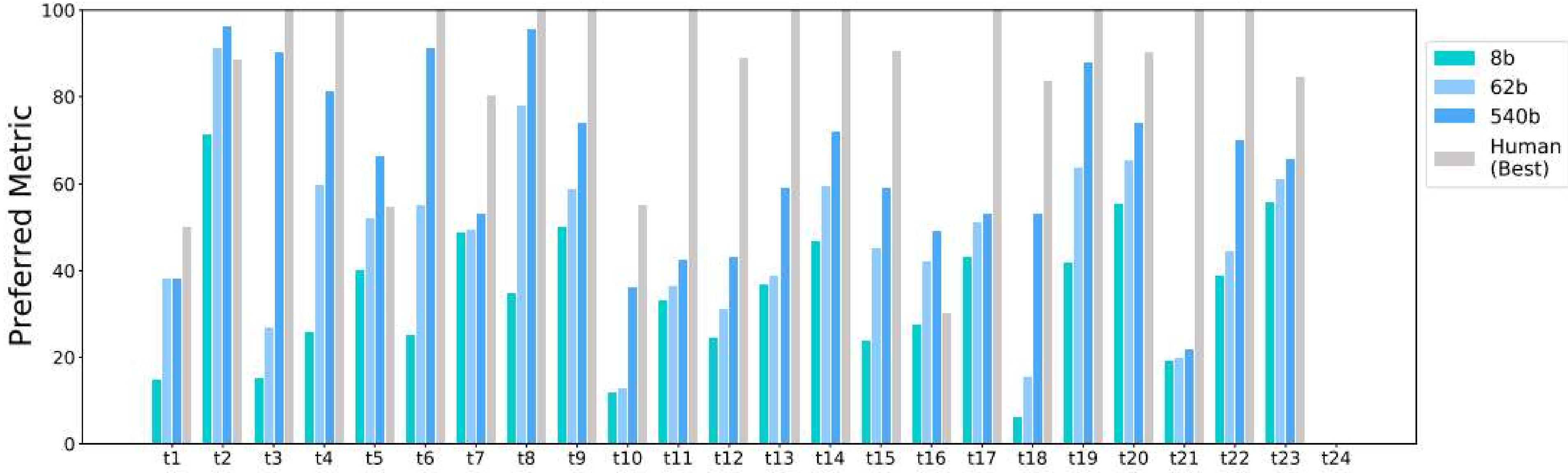


Evaluation – Big Bench



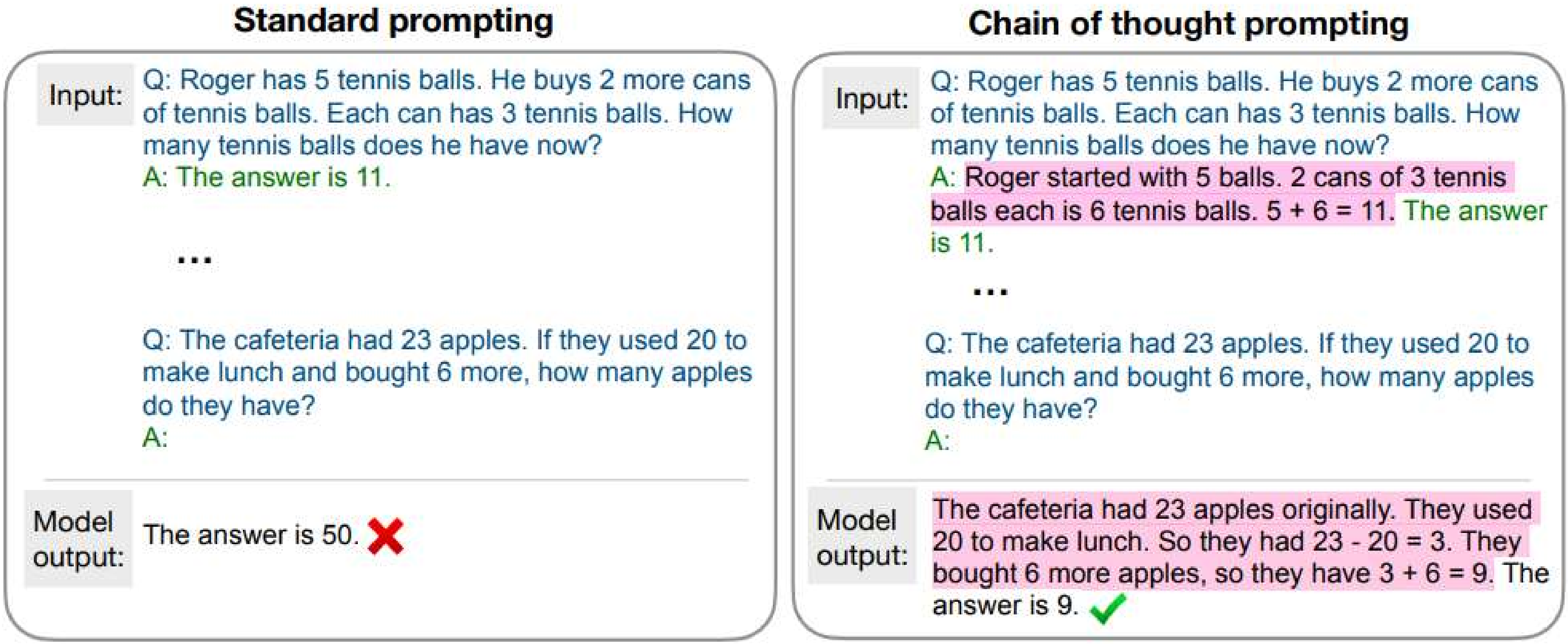
Evaluation – cause and effect

KISTI SINCE 1962



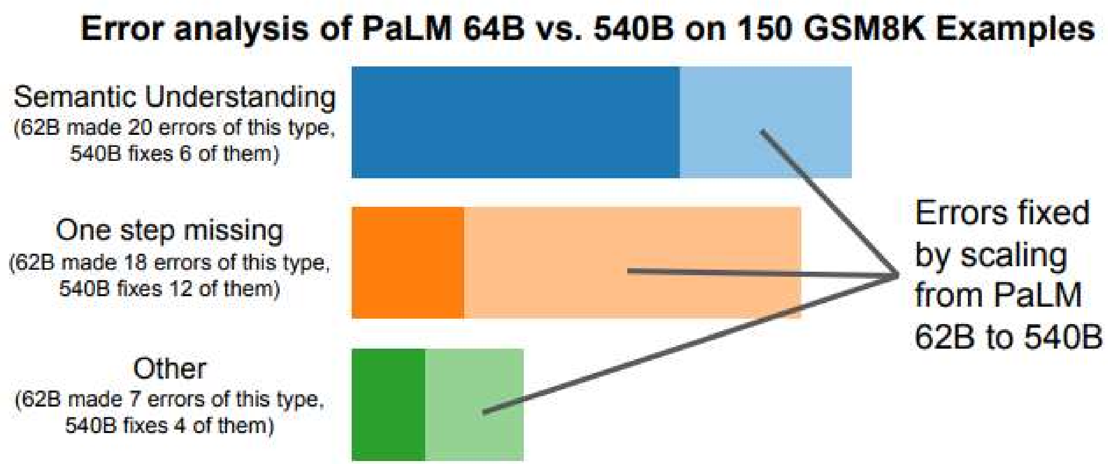
t1	auto debugging	**t2**	bbq lite json	**t3**	code line description	**t4**	conceptual combinations
t5	conlang translation	**t6**	emoji movie	**t7**	formal fallacies syllogisms negation	**t8**	hindu knowledge
t9	known unknowns	**t10**	language identification	**t11**	logic grid puzzle	**t12**	logical deduction
t13	misconceptions russian	**t14**	novel concepts	**t15**	operators	**t16**	parsinlu reading comprehension
t17	play dialog same or different	**t18**	repeat copy logic	**t19**	strange stories	**t20**	strategyqa
t21	symbol interpretation	**t22**	vitaminc fact verification	**t23**	winowhy	**t24**	linguistics puzzles.

Evaluation – Reasoning



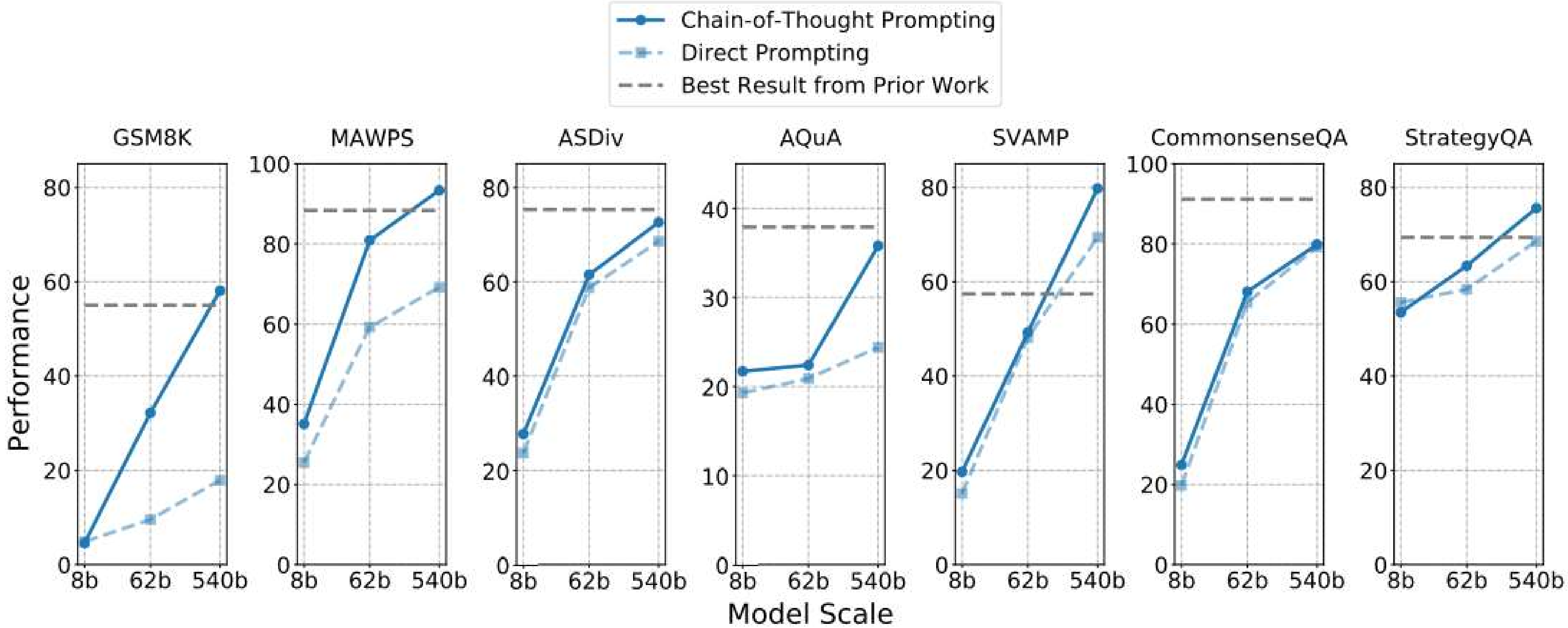
Evaluation – Reasoning

Model+Technique	Accuracy
PaLM 540B+chain-of-thought+calculator	**58%**
PaLM 540B+chain-of-thought	54%
PaLM 540B w/o chain-of-thought	17%
PaLM 62B+chain-of-thought	33%
GPT-3+finetuning+chain-of-thought+calculator	34%
GPT-3+finetuning+chain-of-thought+calculator+verifier	55%



Evaluation – Reasoning

8-shot prediction with PaLM 540B+chain-of-thought



Evaluation – Code tasks

- Text to Code
- Code to Code

	Code tokens		Code web docs
	Total code	Python	
LaMDA 137B	–	–	18B
Codex 12B	100B	100B	–
PaLM 540B	39B	2.7B	–
PaLM-Coder 540B	46.8B	8.7B	–

prompt

```
def exercise5():
    """Marie ordered one chicken meal that costs $12, 5
    packs of milk that costs $3 each, 4 apples that cost
    $1.50 each, and some boxes of pizza. Marie paid a total
    of $50. How many boxes of pizza did Marie order if each
    box costs $8.50?"""
```

model

```
    total_paid = 50
    cost_of_pizza_per_box = 8.50
    cost_of_meal = 12
    cost_of_milk_per_pack = 3
    cost_of_apples_per_apple = 1.50
    packages_of_milk = 5

    return float(total_paid - cost_of_meal - cost_of_milk_per_pack *
            packages_of_milk - cost_of_apples_per_apple *
            4) / cost_of_pizza_per_box
```

prompt

```
// Translate from C to Python:
long long int sum_pairwise_products ( int n ) {
 long long int sum = 0;
 for ( int i = 1; i <= n; i ++ ) {
  for ( int j = i; j <= n; j ++ ) {
   sum = sum + i * j;
  }
 }
 return sum; }
```

model

```
def sum_pairwise_products(n):
    sm = 0
    for i in range(1, (n + 1)):
        for j in range(i, (n + 1)):
            sm = (sm + (i * j))
    return sm
```

prompt

```
# Given a string representing musical notes in a
special ASCII format,
# parse this string and compute a total length of the
music in beats.
# Legend:
#     'o' - whole note, lasts four beats
#     'o|' - half note, lasts two beats
#     '.|' - quarter note, lasts one beat

notes = 'o o| .| o| o| .| .| .| .| o o'
```

model

```
# 1. Split the string into a list of notes.
# 2. Iterate over the list and compute the total
length of the music.
# 3. Print the result.

# 1.
notes = notes.split()

# 2.
total_length = 0
for note in notes:
    if note == 'o':
        total_length += 4
    if note == 'o|':
        total_length += 2
    if note == '.|':
        total_length += 1

# 3.
print(total_length)
```

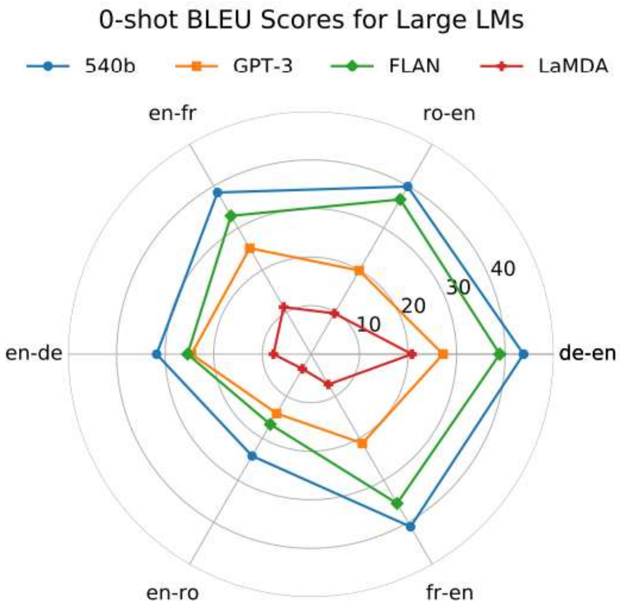
Evaluation – Code tasks

		Pretraining only		Code Finetuning			
		LaMDA 137B	PaLM 540B	Codex 12B[a]	Davinci Codex*	PaLM Coder 540B	Other Work
HumanEval (0)	pass@100	47.3	76.2	72.3	81.7	**88.4**	–
MBPP (3)	pass@80	62.4[b]	75.0	–	**84.4**	80.8	–
TransCoder (3)	pass@25	–	79.8	–	71.7	**82.5**	67.2[c]
HumanEval (0)	pass@1	14.0	26.2	28.8	**36.0**	**36.0**	–
MBPP (3)	pass@1	14.8[b]	36.8	–	**50.4**	47.0	–
GSM8K-Python (4)	pass@1	7.6	**51.3**	–	32.1	50.9	–
TransCoder (3)	pass@1	30.2	51.8	–	54.4	**55.1**	44.5[c]
DeepFix (2)	pass@1	4.3	73.7	–	81.1	**82.1**	71.7[d]

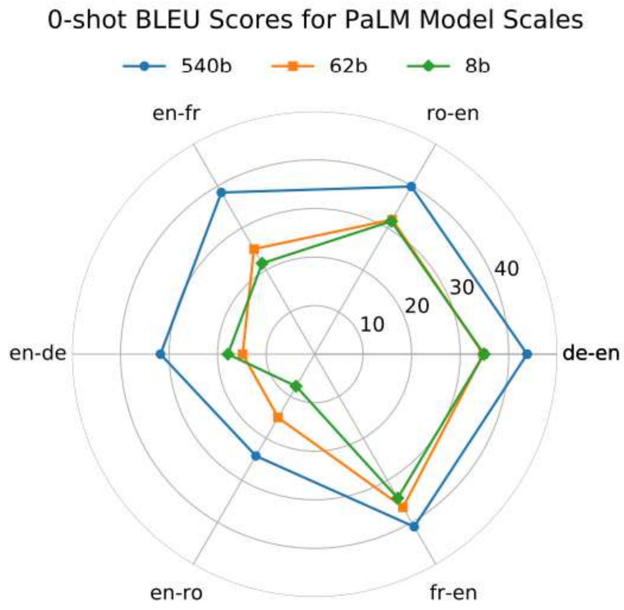
Evaluation – Translation

BLEU Scores

		0-shot		1-shot		Few-shot		Supervised
Src	Tgt	Prior SOTA	PaLM 540B	Prior SOTA	PaLM 540B	Prior SOTA	PaLM 540B	Finetuned SOTA
en	fr	32.9[a]	**38.5**	28.3[b]	**37.5**	33.9[a] (9)	**44.0**	45.6[c]
en	de	25.4[a]	**31.8**	26.2[b]	**31.8**	26.8[a] (11)	**37.4**	41.2[d]
en	ro	16.7[a]	**24.2**	20.6[b]	**28.2**	20.5[a] (9)	**28.7**	33.4[e]
fr	en	35.5[a]	**41.1**	33.7[b]	**37.4**	38.0[a] (9)	**42.8**	45.4[f]
de	en	38.9[a]	**43.8**	30.4[b]	**43.9**	40.6[a] (11)	**47.5**	41.2[g]
ro	en	36.8[a]	**39.9**	38.6[b]	**42.1**	37.3[a] (9)	**43.8**	39.1[h]



(a)



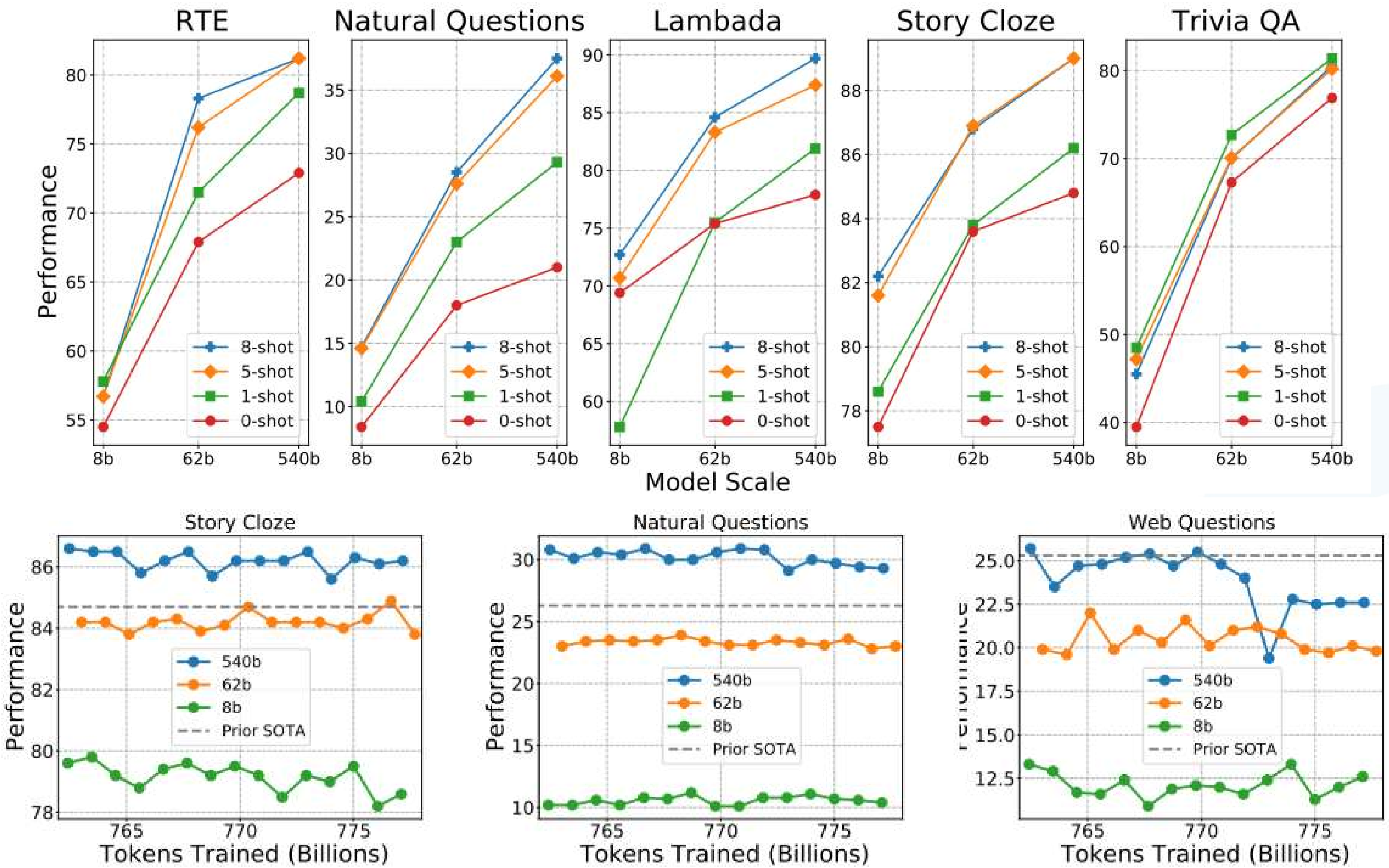
(b)

Evaluation – Multilingual QA

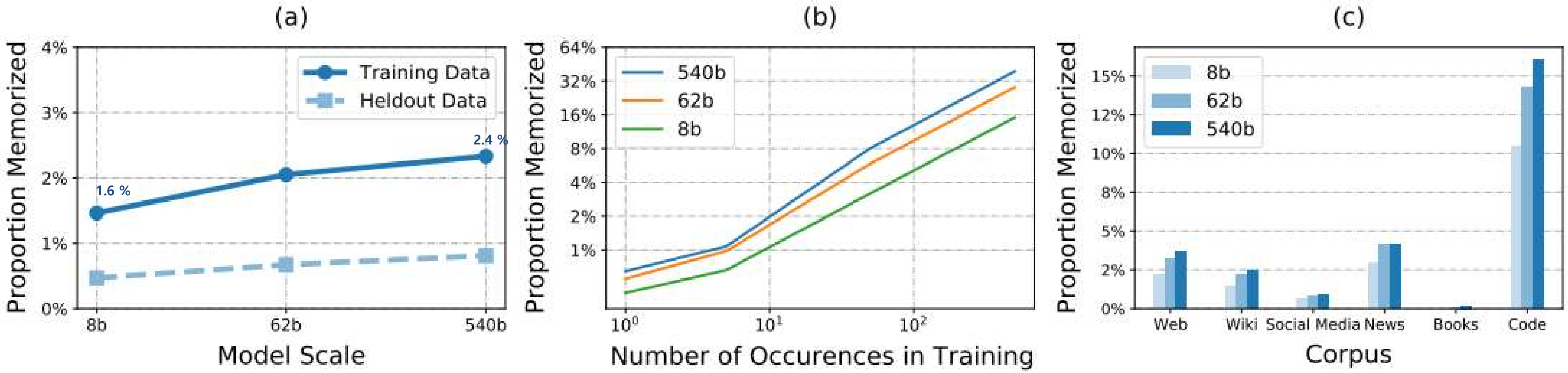
Model	Ar	Bn	En	Fi	Id	Ko	Ru	Sw	Te	Avg
mT5 XXL	76.9	80.5	75.5	76.3	81.8	75.7	76.8	84.4	83.9	79.1
ByT5 XXL	**80.0**	**85.0**	**77.7**	78.8	**85.7**	**78.3**	**78.2**	84.0	**85.5**	**81.4**
PaLM 540B (*finetuned*)	75.0	83.2	75.5	**78.9**	84.1	75.7	77.1	**85.2**	84.9	80.0
PaLM 540B (*few-shot*)	56.4 (5)	54.0 (1)	65.5 (10)	66.4 (5)	69.2 (5)	63.8 (5)	46.8 (5)	75.6 (10)	46.9 (1)	60.5

Table 16: Comparison against SOTA on TyDiQA-GoldP validation set (exact match metric).

Evaluation – Model Analysis



Memorization

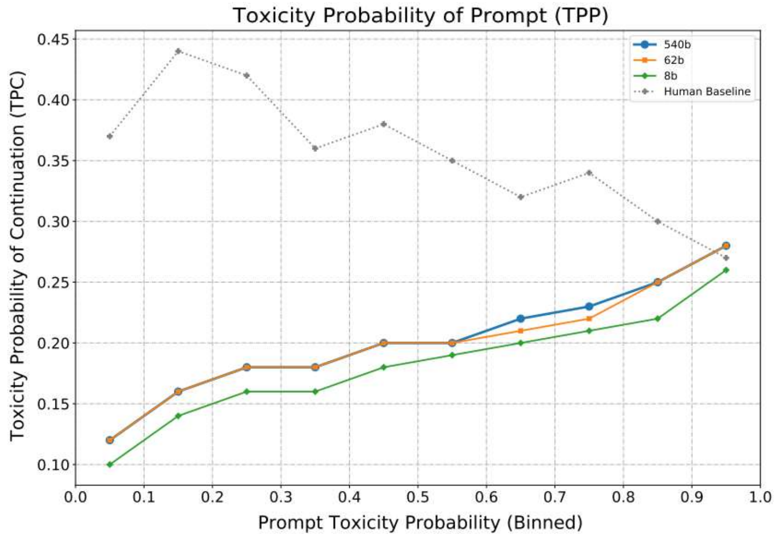


Dataset Contamination

Dataset	Clean Proportion	PaLM 8B 1-Shot Full Set Accuracy	PaLM 8B 1-Shot Clean Subset Delta	PaLM 540B 1-Shot Full Set Accuracy	PaLM 540B 1-Shot Clean Subset Delta
TriviaQA (Wiki)	80.1%	48.5	+0.5	81.4	+0.1
WebQuestions	73.3%	12.6	+1.1	22.6	+0.3
Lambada	70.7%	57.8	+0.6	81.8	+0.0
Winograd	61.5%	82.4	-4.4	87.5	-1.8
SQuADv2 (F1)	14.8%	50.1	-2.5	82.9	+1.1
ARC-e	69.6%	71.3	-0.3	85.0	-0.4
ARC-c	75.3%	42.3	+0.4	60.1	-1.1
WSC	63.2%	81.4	-1.4	86.3	-3.5
ReCoRD	56.6%	87.8	-2.0	92.8	-1.6
CB	51.8%	41.1	-3.1	83.9	+5.8

Challenges and Limitations

- Gender and occupation bias
- Toxicity and bias
- Toxicity in open-ended generation
- Training data has the English language in the majority
- Ethical considerations



Conclusion

- Efficient scaling – Used Pathways to create a big ML system
- Continued improvements from scaling
- Breakthrough capabilities – especially in reasoning tasks
- Discontinuous improvements
- Multilingual understanding
- Bias and toxicity