Multilingual Question Answering: Methods

601.764

2/28/23

IBM Watson Wins Jeopardy!

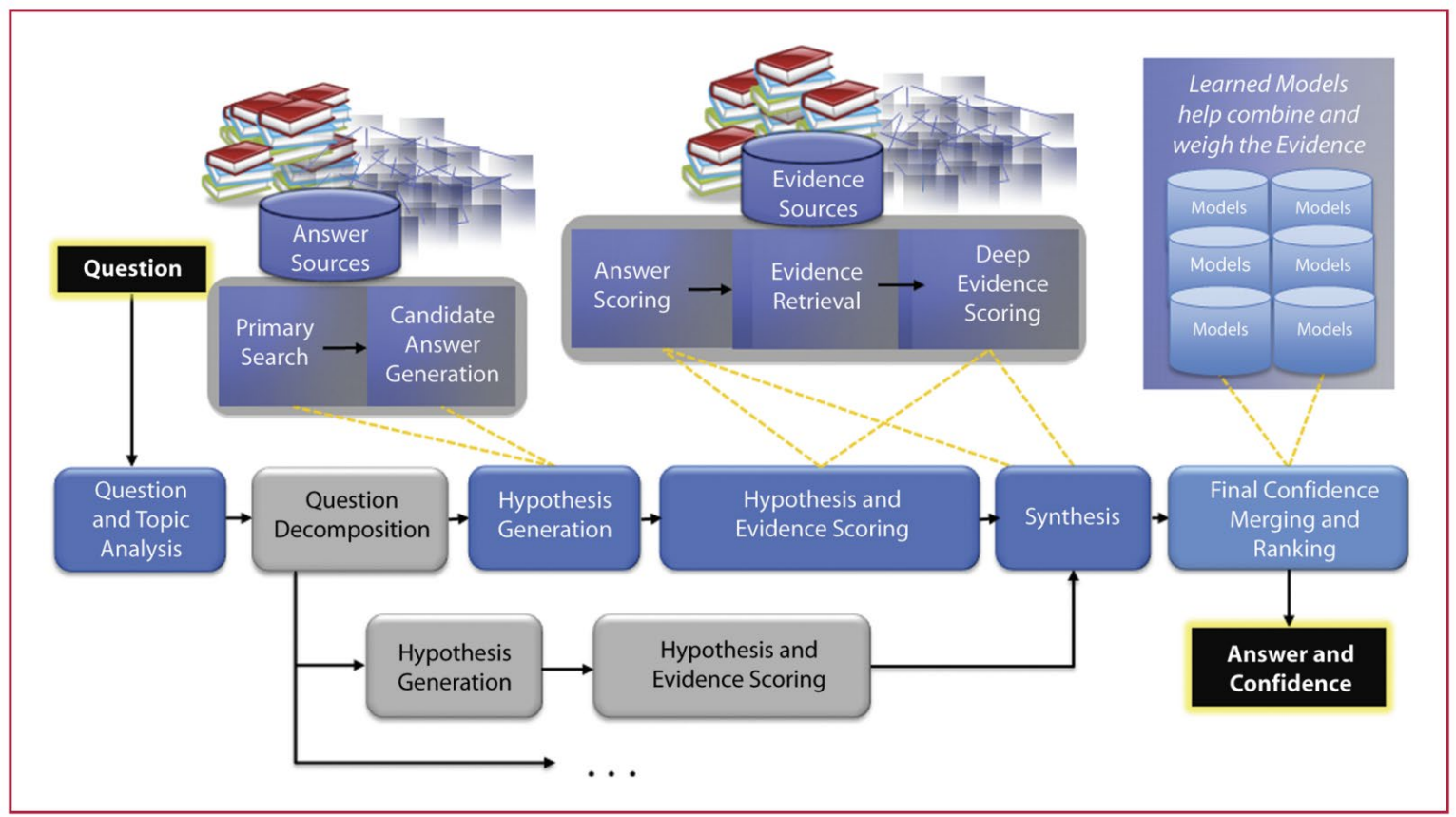


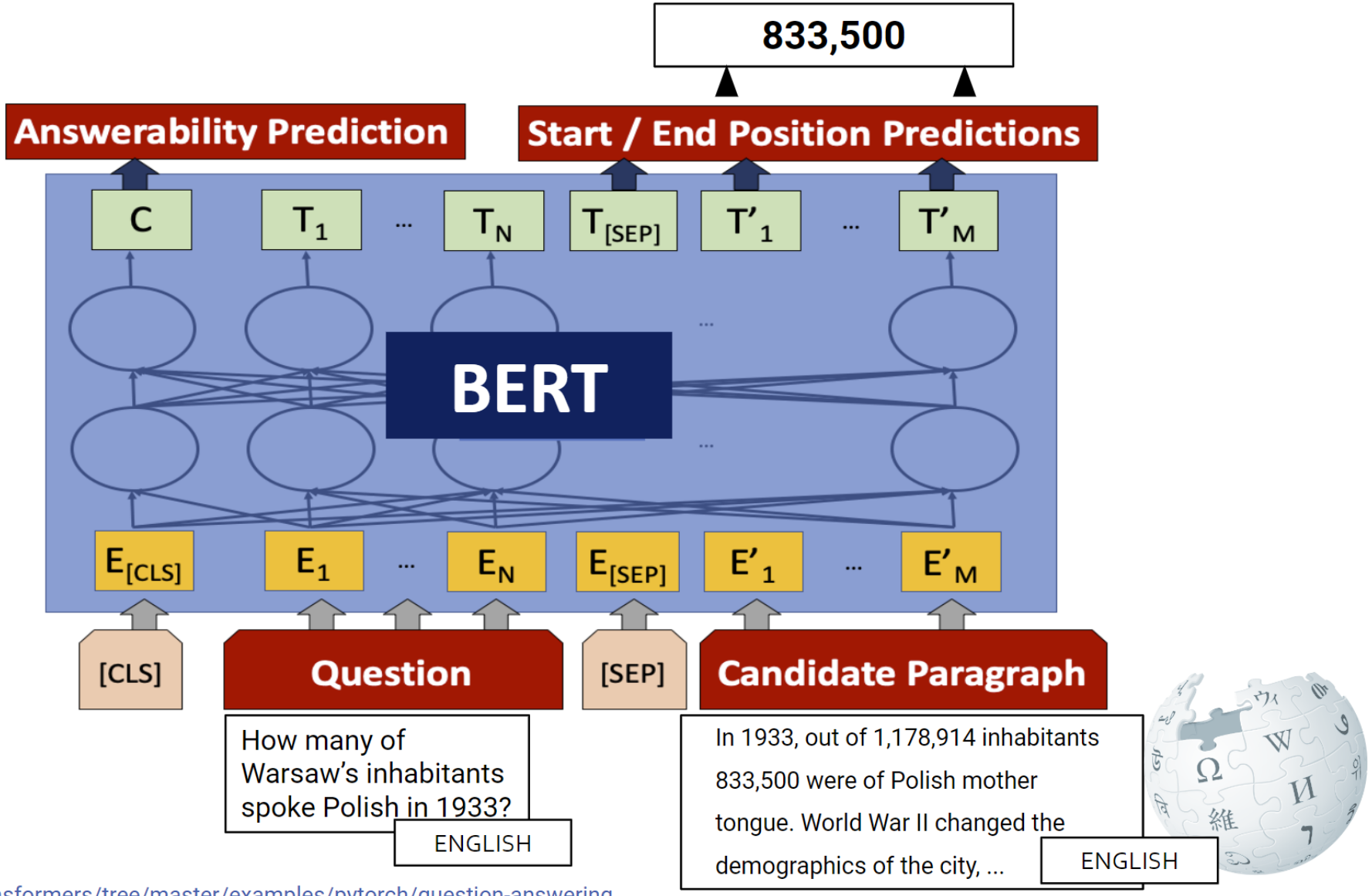
Figure 1

DeepQA architecture.

Ferucci 2012

Machine Reading Comprehension (MRC)

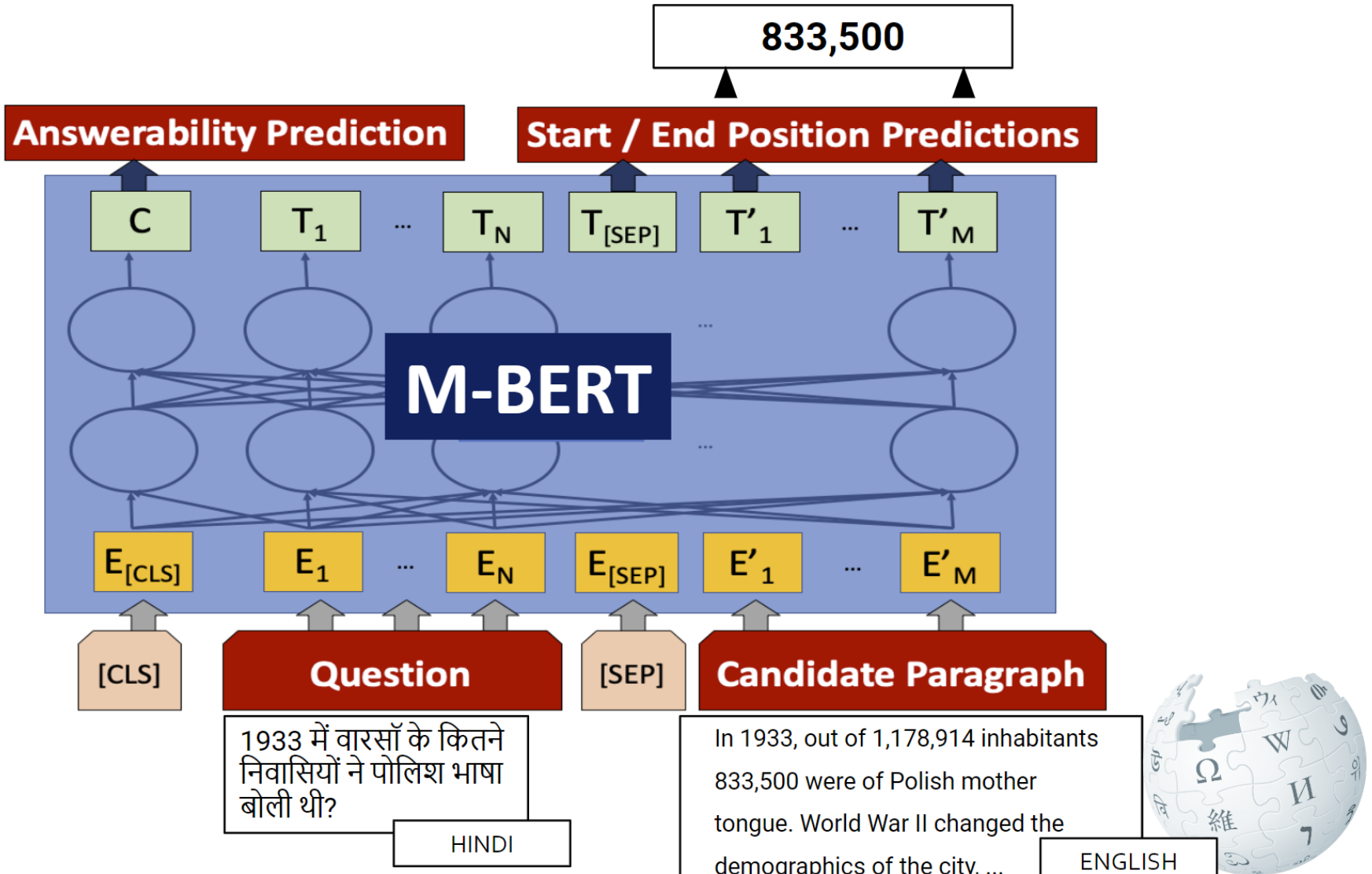
- Popular choice: Add a fine-tuning layer on top of BERT [Devlin et al., 2019]



Code: https://github.com/huggingface/transformers/tree/master/examples/pytorch/question-answering

Multilingual Machine Reading Comprehension (MRC)

- Popular choice: Add a fine-tuning layer on top of M-BERT [Bornea et al., 2021]



Code: https://github.com/huggingface/transformers/tree/master/examples/pytorch/question-answering

TyDiQA-GoldP Exact Match

- 8 Languages

RETHINKING EMBEDDING COUPLING IN PRE-TRAINED LANGUAGE MODELS

Hyung Won Chung[*†]
Google Research
hwchung@google.com

Thibault Févry[*†]
thibaultfevry@gmail.com

Henry Tsai
Google Research
henrytsai@google.com

Melvin Johnson
Google Research
melvinp@google.com

Sebastian Ruder
DeepMind
ruder@google.com

Table 1: Overview of the number of parameters in (coupled) embedding matrices of state-of-the-art multilingual (top) and monolingual (bottom) models with regard to overall parameter budget. $|V|$: vocabulary size. N, N_{emb}: number of parameters in total and in the embedding matrix respectively.

Model	Languages	$\lvert V \rvert$	N	N_{emb}	%Emb.
mBERT (Devlin et al., 2019)	104	120k	178M	92M	52%
XLM-R_{Base} (Conneau et al., 2020a)	100	250k	270M	192M	71%
XLM-R_{Large} (Conneau et al., 2020a)	100	250k	550M	256M	47%
$\text{BERT}_{\text{Base}}$ (Devlin et al., 2019)	1	30k	110M	23M	21%
$\text{BERT}_{\text{Large}}$ (Devlin et al., 2019)	1	30k	335M	31M	9%

Table 2: Effect of decoupling the input and output embedding matrices on performance on multiple tasks in XTREME. PT: Pre-training. FT: Fine-tuning.

	# PT params	# FT params	XNLI Acc	NER F1	PAWS-X Acc	XQuAD EM/F1	MLQA EM/F1	TyDi-GoldP EM/F1	Avg
Coupled	177M	177M	70.7	**69.2**	**85.3**	46.2/63.2	**37.3/53.1**	40.7/56.7	62.3
Decoupled	269M	177M	**71.3**	68.9	85.0	**46.9/63.8**	**37.3/53.1**	**42.8/58.1**	**62.7**

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

PaLM

- Transformer Based
- Decoder Only
- SwiGLU Activation
- Parallel Layers
- Multi-Query Attention
- RoPE Embeddings (better on long sequence)
- Shared Input-Output (Coupled)
- No Biases
- 256k Sentence Piece

PaLM

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

PaLM Chain-of-Though

Standard prompting

Input:

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?
A: The answer is 11.

...

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?
A:

Model output:

The answer is 50. ✗

Chain of thought prompting

Input:

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?
A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

...

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?
A:

Model output:

The cafeteria had 23 apples originally. They used 20 to make lunch. So they had 23 - 20 = 3. They bought 6 more apples, so they have 3 + 6 = 9. The answer is 9. ✓

PaLM

- Few-shot setting, we provide the context, question and answer which are all separated by new line characters.
- “Q:” to denote the question
- “A:” to denote the answer for all the languages
- Finetuning:
  - LR 5 $x10^{-5}$
  - Reset Adafactor Acuumulators
  - Batch Size of 32, we use the same set of hyperparameters as the English SuperGLUE finetuning experiments.

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

Transcending Scaling Laws with 0.1% Extra Compute

Yi Tay **Jason Wei** **Hyung Won Chung** **Vinh Q. Tran** **David R. So** **Siamak Shakeri**
Xavier Garcia **Huaixiu Steven Zheng** **Jinfeng Rao** **Aakanksha Chowdhery**
Denny Zhou **Donald Metzler** **Slav Petrov** **Neil Houlsby**
Quoc V. Le **Mostafa Dehghani**

Google

U-PaLM

- “The key idea is to continue training a state-of-the-art large language model (e.g., PaLM) on a few more steps with UL2’s mixture-of-denoiser objective.”
- ~ Negligible extra computational costs and no new sources of data → Improve Scaling
- Continue training PaLM with UL2R

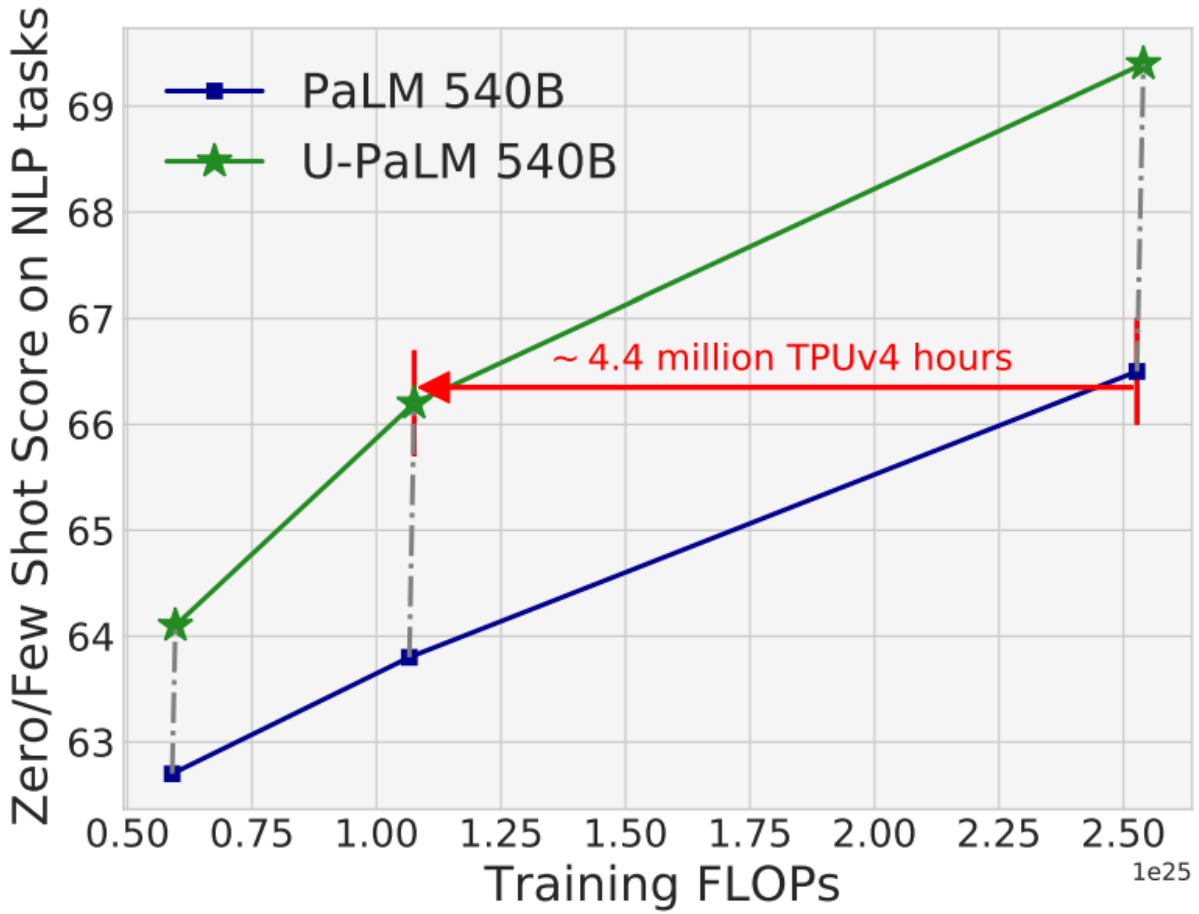


Figure 1: Compute (training flops) versus Quality (average of 20+ NLP zero and few-shot tasks listed in Appendix 7.1). The black dotted line shows the path from initialization from a PaLM checkpoint and training further with UL2R.

U-PaLM

- Still Chain-of-Thought

Task / Model	PaLM 540B	U-PaLM 540B
TydiQA	52.9	**54.6** (+3.2%)

mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer

Linting Xue* **Noah Constant*** **Adam Roberts***
Mihir Kale **Rami Al-Rfou** **Aditya Siddhant** **Aditya Barua** **Colin Raffel**
Google Research

Model	Sentence pair		Structured	Question answering		
	XNLI	PAWS-X	WikiAnn NER	XQuAD	MLQA	TyDiQA-GoldP
Metrics	Acc.	Acc.	F1	F1 / EM	F1 / EM	F1 / EM
Cross-lingual zero-shot transfer (models fine-tuned on English data only)						
mBERT	65.4	81.9	62.2	64.5 / 49.4	61.4 / 44.2	59.7 / 43.9
XLM	69.1	80.9	61.2	59.8 / 44.3	48.5 / 32.6	43.6 / 29.1
InfoXLM	81.4	-	-	- / -	73.6 / 55.2	- / -
X-STILTs	80.4	87.7	64.7	77.2 / 61.3	72.3 / 53.5	76.0 / 59.5
XLM-R	79.2	86.4	65.4	76.6 / 60.8	71.6 / 53.2	65.1 / 45.0
VECO	79.9	88.7	65.7	77.3 / 61.8	71.7 / 53.2	67.6 / 49.1
RemBERT	80.8	87.5	**70.1**	79.6 / 64.0	73.1 / 55.0	77.0 / 63.0
mT5-Small	67.5	82.4	50.5	58.1 / 42.5	54.6 / 37.1	35.2 / 23.2
mT5-Base	75.4	86.4	55.7	67.0 / 49.0	64.6 / 45.0	57.2 / 41.2
mT5-Large	81.1	88.9	58.5	77.8 / 61.5	71.2 / 51.7	69.9 / 52.2
mT5-XL	82.9	89.6	65.5	79.5 / 63.6	73.5 / 54.5	75.9 / 59.4
mT5-XXL	**85.0**	**90.0**	69.2	**82.5 / 66.8**	**76.0 / 57.4**	**80.8 / 65.9**
Translate-train (models fine-tuned on English data plus translations in all target languages)						
XLM-R	82.6	90.4	-	80.2 / 65.9	72.8 / 54.3	66.5 / 47.7
FILTER + Self-Teaching	83.9	91.4	-	82.4 / 68.0	76.2 / 57.7	68.3 / 50.9
VECO	83.0	91.1	-	79.9 / 66.3	73.1 / 54.9	75.0 / 58.9
mT5-Small	64.7	79.9	-	64.3 / 49.5	56.6 / 38.8	48.2 / 34.0
mT5-Base	75.9	89.3	-	75.3 / 59.7	67.6 / 48.5	64.0 / 47.7
mT5-Large	81.8	91.2	-	81.2 / 65.9	73.9 / 55.2	71.1 / 54.9
mT5-XL	84.8	91.0	-	82.7 / 68.1	75.1 / 56.6	79.9 / 65.3
mT5-XXL	**87.8**	**91.5**	-	**85.2 / 71.3**	**76.9 / 58.3**	**82.8 / 68.8**
In-language multitask (models fine-tuned on gold data in all target languages)						
mBERT	-	-	89.1	-	-	77.6 / 68.0
mT5-Small	-	-	83.4	-	-	73.0 / 62.0
mT5-Base	-	-	85.4	-	-	80.8 / 70.0
mT5-Large	-	-	88.4	-	-	85.5 / 75.3
mT5-XL	-	-	90.9	-	-	87.5 / 78.1
mT5-XXL	-	-	**91.2**	-	-	**88.5 / 79.1**

	T5	mT5
Small	87.2 / 79.1	84.7 / 76.4
Base	92.1 / 85.4	89.6 / 83.8
Large	93.8 / 86.7	93.0 / 87.0
XL	95.0 / 88.5	94.5 / 88.9
XXL	96.2 / 91.3	95.6 / 90.4

Table 3: Comparison of T5 vs. mT5 on SQuAD question answering (F1/EM).

ByT5: Towards a Token-Free Future with Pre-trained Byte-to-Byte Models

Linting Xue*, **Aditya Barua***, **Noah Constant***, **Rami Al-Rfou***,
Sharan Narang, Mihir Kale, Adam Roberts, Colin Raffel
Google Research
{lintingx, adityabarua, nconstant, rmyeid, sharannarang, mihirkale, adarob}
@google.com, craffel@gmail.com

ByT5

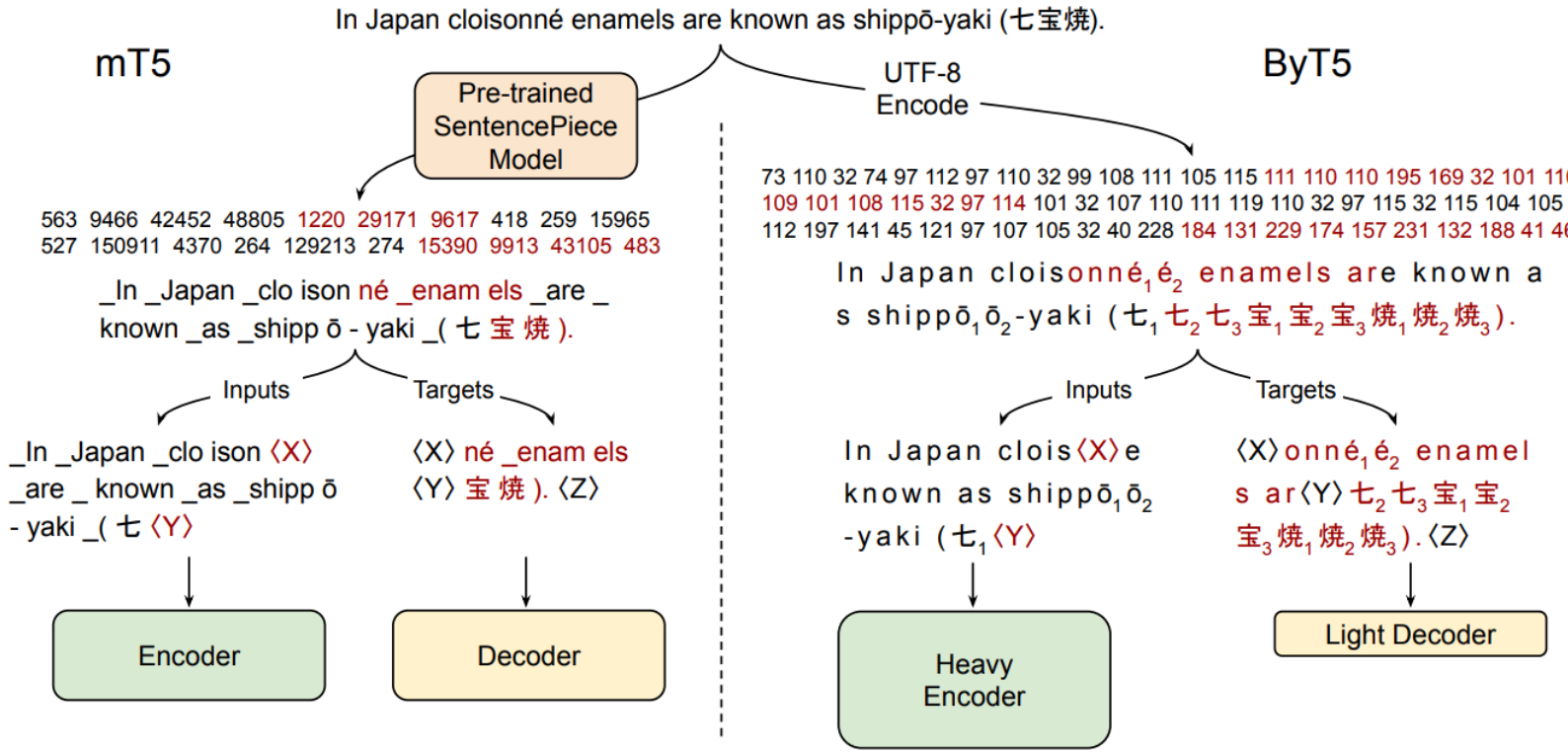


Figure 1: Pre-training example creation and network architecture of mT5 (Xue et al., 2021) vs. ByT5 (this work). **mT5**: Text is split into SentencePiece tokens, spans of ~3 tokens are masked (red), and the encoder/decoder transformer stacks have equal depth. **ByT5**: Text is processed as UTF-8 bytes, spans of ~20 bytes are masked, and the encoder is 3× deeper than the decoder. ⟨X⟩, ⟨Y⟩, and ⟨Z⟩ represent sentinel tokens.

ByT5

	mT5 XXL	ByT5XXL
In-Language TyDiQA	79.5	81.4
Translate-Train	69.4	69.6

Scaling Instruction-Finetuned Language Models

Hyung Won Chung* Le Hou* Shayne Longpre* Barret Zoph† Yi Tay†
William Fedus† Yunxuan Li Xuezhi Wang Mostafa Dehghani Siddhartha Brahma
Albert Webson Shixiang Shane Gu Zhuyun Dai Mirac Suzgun Xinyun Chen
Aakanksha Chowdhery Alex Castro-Ros Marie Pellat Kevin Robinson
Dasha Valter Sharan Narang Gaurav Mishra Adams Yu Vincent Zhao
Yanping Huang Andrew Dai Hongkun Yu Slav Petrov Ed H. Chi
Jeff Dean Jacob Devlin Adam Roberts Denny Zhou Quoc V. Le
Jason Wei*

Google

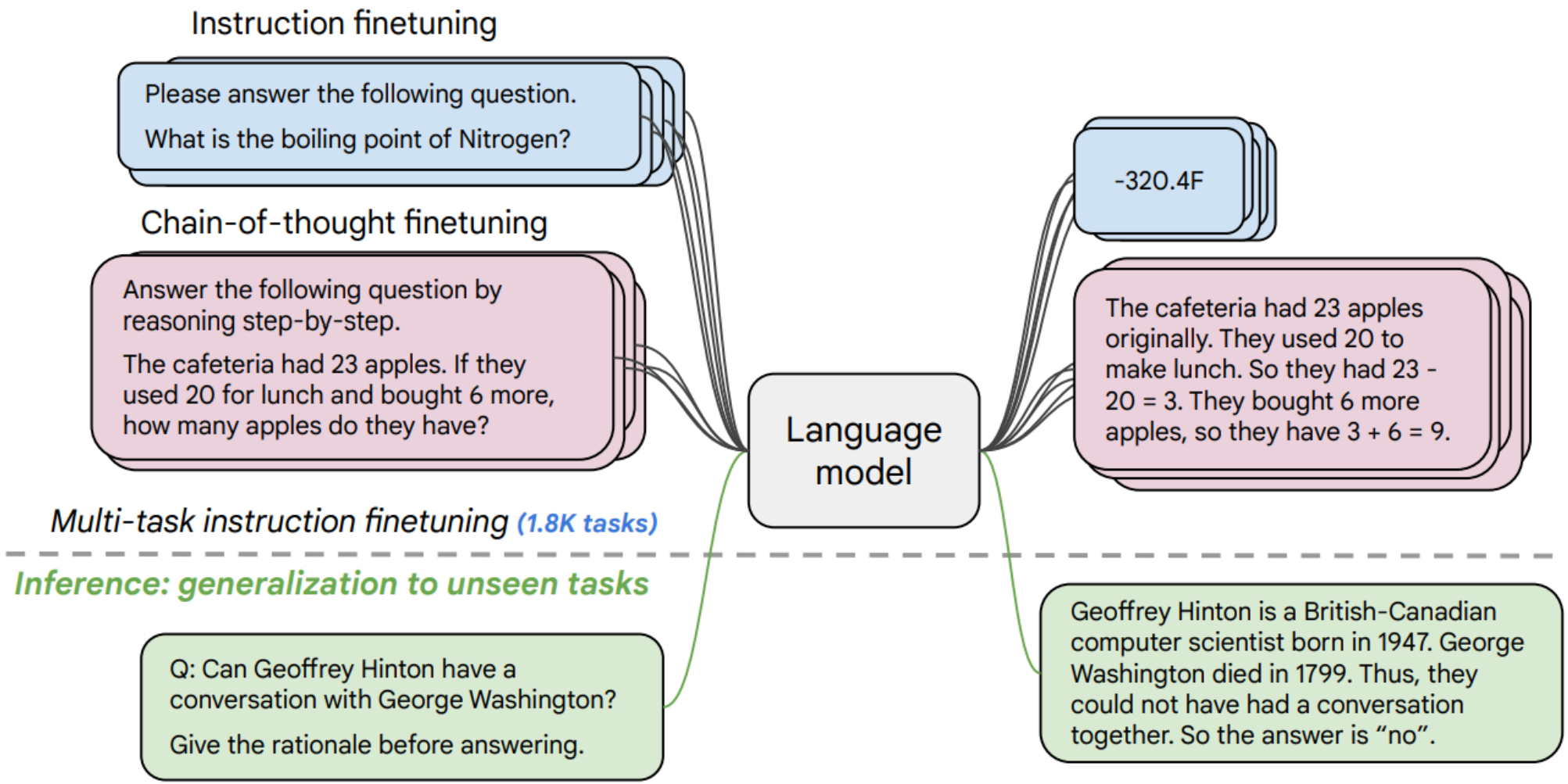


Figure 1: We finetune various language models on 1.8K tasks phrased as instructions, and evaluate them on unseen tasks. We finetune both with and without exemplars (i.e., zero-shot and few-shot) and with and without chain-of-thought, enabling generalization across a range of evaluation scenarios.

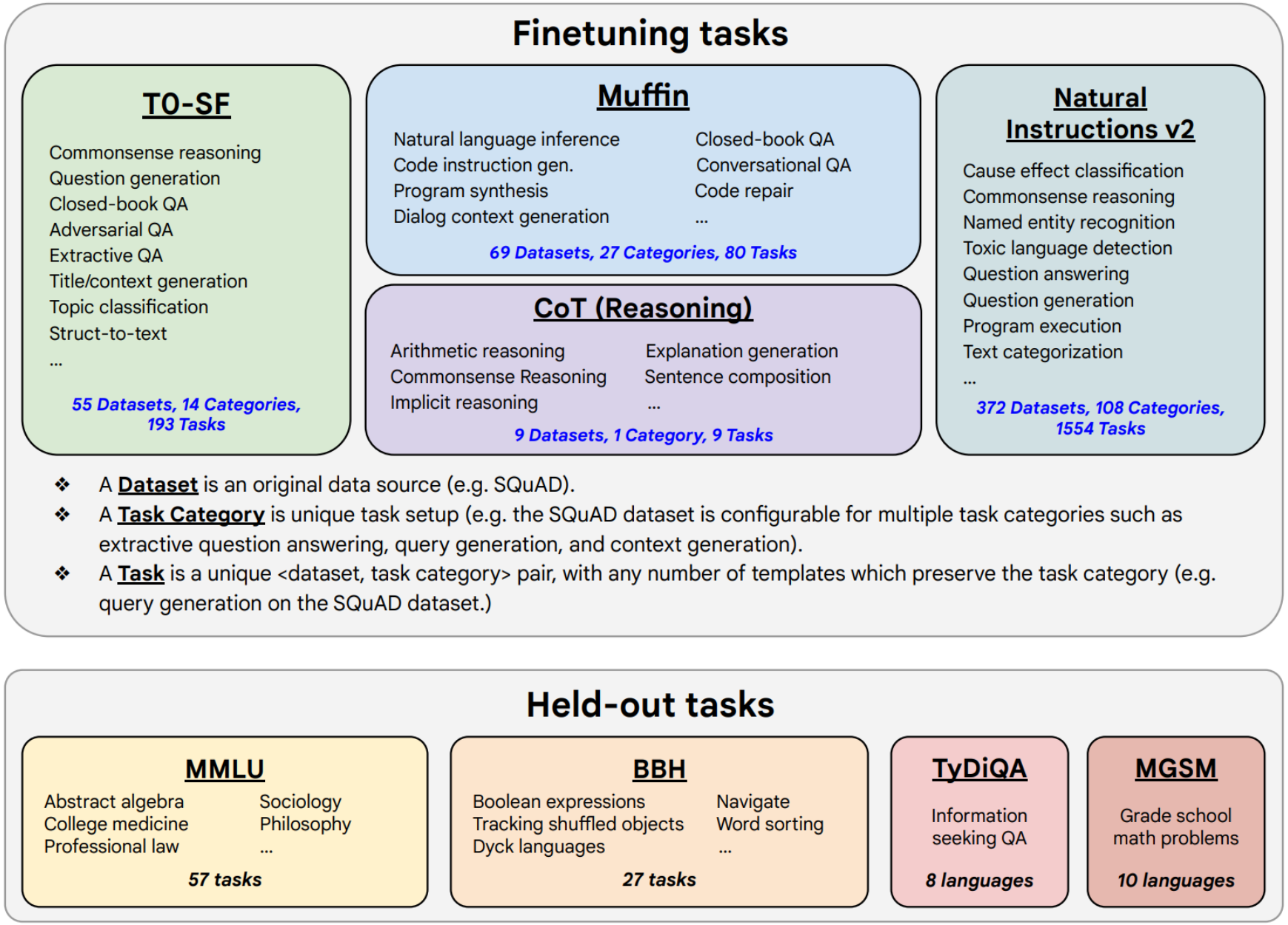


Figure 2: Our finetuning data comprises 473 datasets, 146 task categories, and 1,836 total tasks. Details for the tasks used in this paper is given in Appendix F.

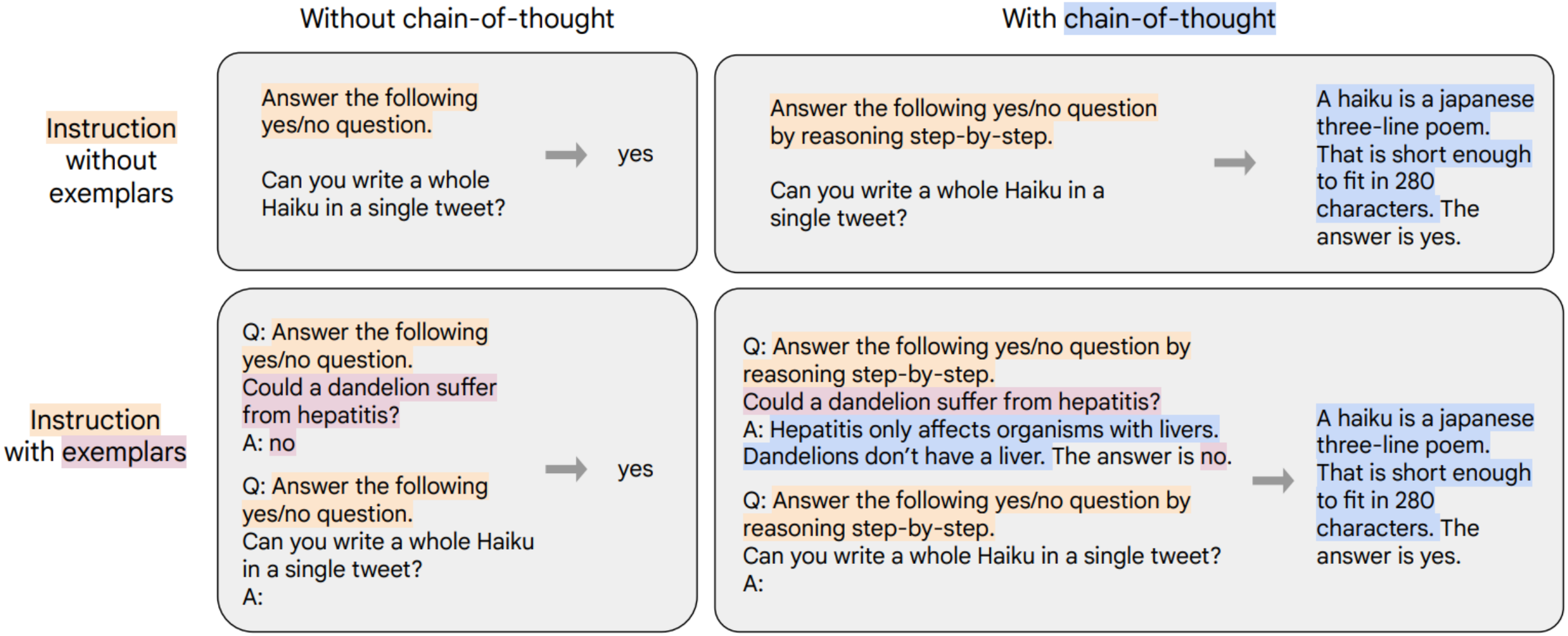


Figure 3: Combinations of finetuning data formats in this work. We finetune with and without exemplars, and also with and without chain-of-thought. In addition, we have some data formats without instructions but with few-shot exemplars only, like in Min et al. (2022) (not shown in the figure). Note that only nine chain-of-thought (CoT) datasets use the CoT formats.

Params	Model	Architecture	Pre-training Objective	Pre-train FLOPs	Finetune FLOPs	% Finetune Compute
80M	Flan-T5-Small	encoder-decoder	span corruption	1.8E+20	2.9E+18	1.6%
250M	Flan-T5-Base	encoder-decoder	span corruption	6.6E+20	9.1E+18	1.4%
780M	Flan-T5-Large	encoder-decoder	span corruption	2.3E+21	2.4E+19	1.1%
3B	Flan-T5-XL	encoder-decoder	span corruption	9.0E+21	5.6E+19	0.6%
11B	Flan-T5-XXL	encoder-decoder	span corruption	3.3E+22	7.6E+19	0.2%
8B	Flan-PaLM	decoder-only	causal LM	3.7E+22	1.6E+20	0.4%
62B	Flan-PaLM	decoder-only	causal LM	2.9E+23	1.2E+21	0.4%
540B	Flan-PaLM	decoder-only	causal LM	2.5E+24	5.6E+21	0.2%
62B	Flan-cont-PaLM	decoder-only	causal LM	4.8E+23	1.8E+21	0.4%
540B	Flan-U-PaLM	decoder-only	prefix LM + span corruption	2.5E+23	5.6E+21	0.2%

Table 2: Across several models, instruction finetuning only costs a small amount of compute relative to pre-training. T5: Raffel et al. (2020). PaLM and cont-PaLM (also known as PaLM 62B at 1.3T tokens): Chowdhery et al. (2022). U-PaLM: Tay et al. (2022b).

	MMLU	BBH-nlp	BBH-alg	TyDiQA	MGSM
Prior best	69.3[a]	73.5[b]	**73.9**[b]	**81.9**[c]	55.0[d]
PaLM 540B					
- direct prompting	69.3	62.7	38.3	52.9	18.3
- CoT prompting	64.5	71.2	57.6	-	45.9
- CoT + self-consistency	69.5	78.2	62.2	-	57.9
Flan-PaLM 540B					
- direct prompting	72.2	70.0	48.2	67.8	21.2
- CoT prompting	70.2	72.4	61.3	-	57.0
- CoT + self-consistency	**75.2**	**78.4**	66.5	-	**72.0**

Rank	Model	EM ↑	F1	Paper	Code	Result	Year	Tags
8	**Decoupled**	42.8	58.1	Rethinking embedding coupling in pre-trained language models			2020	

Rank	Model	EM ↑	F1	Paper	Code	Result	Year	Tags
3	**Flan-U-PaLM 540B** (direct-prompting)	68.3		Scaling Instruction-Finetuned Language Models			2022	
4	**Flan-PaLM 540B** (direct-prompting)	67.8		Scaling Instruction-Finetuned Language Models			2022	
8	**Decoupled**	42.8	58.1	Rethinking embedding coupling in pre-trained language models			2020	

Rank	Model	EM ↑	F1	Paper	Code	Result	Year	Tags
3	**Flan-U-PaLM 540B** (direct-prompting)	68.3		Scaling Instruction-Finetuned Language Models			2022	
4	**Flan-PaLM 540B** (direct-prompting)	67.8		Scaling Instruction-Finetuned Language Models			2022	
5	**ByT5 XXL**	60.0	75.3	ByT5: Towards a token-free future with pre-trained byte-to-byte models			2021	
8	**Decoupled**	42.8	58.1	Rethinking embedding coupling in pre-trained language models			2020	

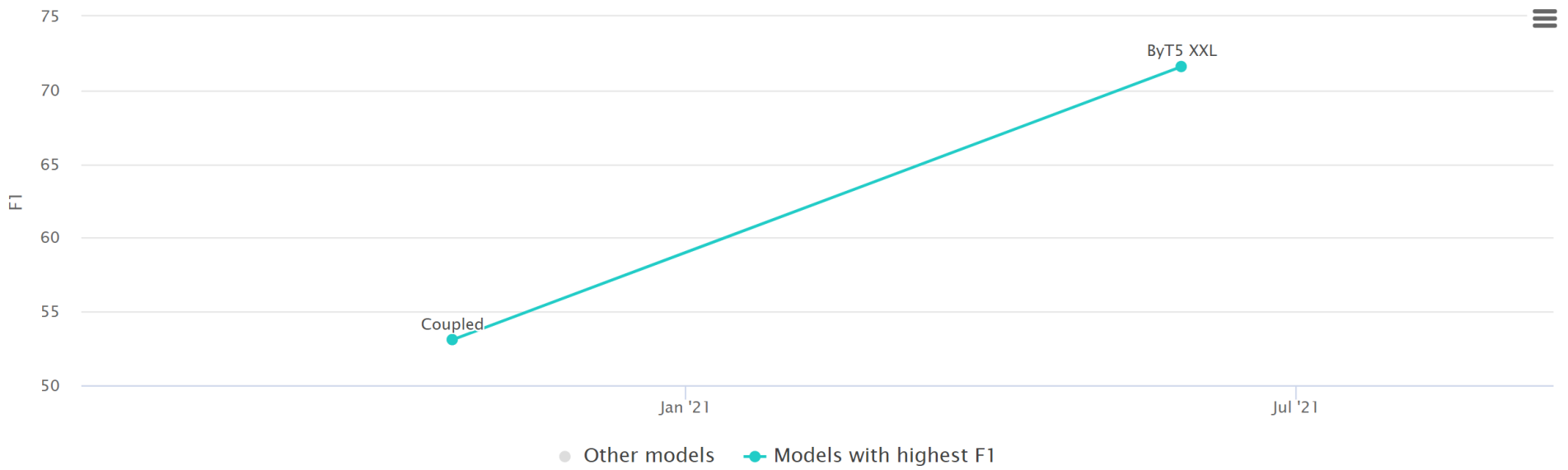
Rank	Model	EM ↑	F1	Paper	Code	Result	Year	Tags
3	**Flan-U-PaLM 540B** (direct-prompting)	68.3		Scaling Instruction-Finetuned Language Models			2022	
4	**Flan-PaLM 540B** (direct-prompting)	67.8		Scaling Instruction-Finetuned Language Models			2022	
5	**ByT5 XXL**	60.0	75.3	ByT5: Towards a token-free future with pre-trained byte-to-byte models			2021	
6	**U-PaLM-540B** (CoT)	54.6		Transcending Scaling Laws with 0.1% Extra Compute			2022	chain-of-thought
7	**PaLM-540B** (CoT)	52.9		PaLM: Scaling Language Modeling with Pathways			2022	chain-of-thought
8	**Decoupled**	42.8	58.1	Rethinking embedding coupling in pre-trained language models			2020	

Rank	Model	EM ↑	F1	Paper	Code	Result	Year	Tags
1	**ByT5** (fine-tuned)	81.9		ByT5: Towards a token-free future with pre-trained byte-to-byte models			2021	fine-tuned
2	**U-PaLM 62B** (fine-tuned)	78.4	88.5	Transcending Scaling Laws with 0.1% Extra Compute			2022	fine-tuned
3	**Flan-U-PaLM 540B** (direct-prompting)	68.3		Scaling Instruction-Finetuned Language Models			2022	
4	**Flan-PaLM 540B** (direct-prompting)	67.8		Scaling Instruction-Finetuned Language Models			2022	
5	**ByT5 XXL**	60.0	75.3	ByT5: Towards a token-free future with pre-trained byte-to-byte models			2021	
6	**U-PaLM-540B** (CoT)	54.6		Transcending Scaling Laws with 0.1% Extra Compute			2022	chain-of-thought
7	**PaLM-540B** (CoT)	52.9		PaLM: Scaling Language Modeling with Pathways			2022	chain-of-thought
8	**Decoupled**	42.8	58.1	Rethinking embedding coupling in pre-trained language models			2020	

Cross-Lingual Question Answering on MLQA

Leaderboard | Dataset

View F1 by Date



Filter: untagged

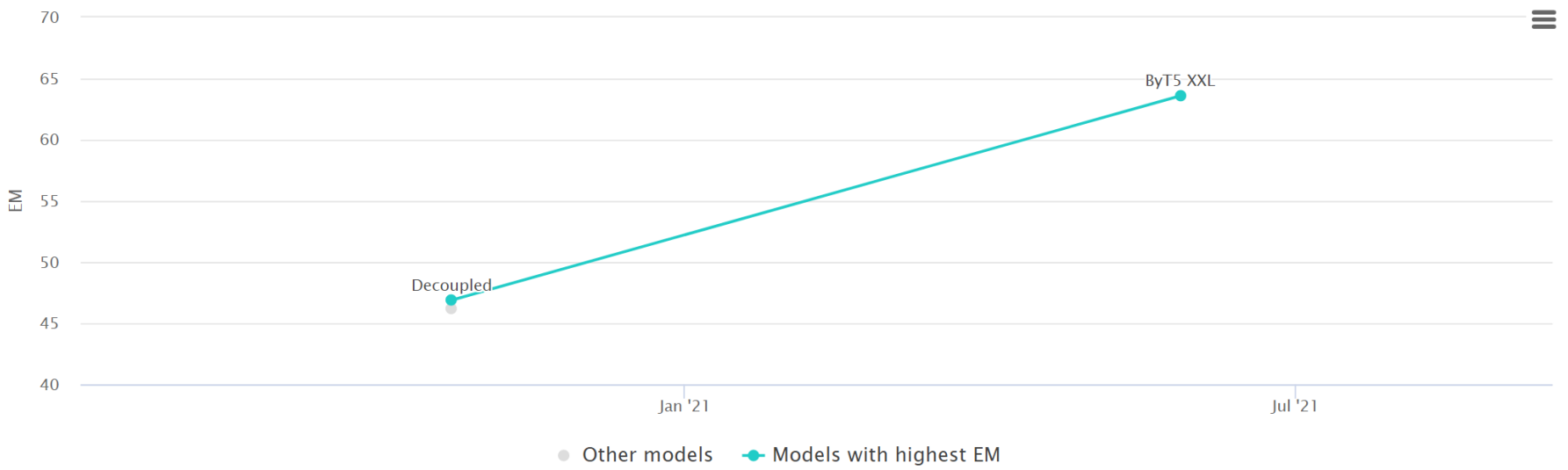
Edit Leaderboard

Rank	Model	F1 ↑	EM	Paper	Code	Result	Year	Tags
1	**ByT5 XXL**	71.6	54.9	ByT5: Towards a token-free future with pre-trained byte-to-byte models			2021	
2	**Coupled**	53.1	37.3	Rethinking embedding coupling in pre-trained language models			2020	
3	**Decoupled**	53.1		Rethinking embedding coupling in pre-trained language models			2020	

Cross-Lingual Question Answering on XQuAD

Leaderboard | Dataset

View EM by Date for All models



Filter: untagged

Edit Leaderboard

Rank	Model	EM	F1	Average F1	Extra Training Data	Paper	Code	Result	Year	Tags
1	**ByT5 XXL**	63.6	79.7		×	ByT5: Towards a token-free future with pre-trained byte-to-byte models			2021	
2	**Decoupled**	46.9	63.8		×	Rethinking embedding coupling in pre-trained language models			2020	
3	**Coupled**	46.2	63.2		×	Rethinking embedding coupling in pre-trained language models			2020	