

材料研究における大規模言語モデルの応用例と現状： プロンプティングからファインチューニングまで

Yuta Suzuki

2024/07/25

AI4MaterialsやMaterials Informaticsの基盤研究をしています

1. Multimodal contrastive learning for materials

- Y. Suzuki, T. Taniai, K. Saito, et al. *Mach. Learn.: Sci. Technol.* 3 045034 (2022).

2. Automatic powder mixing robot

- Y. Nakajima, M. Hamaya, Y. Suzuki, et al. *In proc. IROS2022* pp. 2320-2326 (2022).

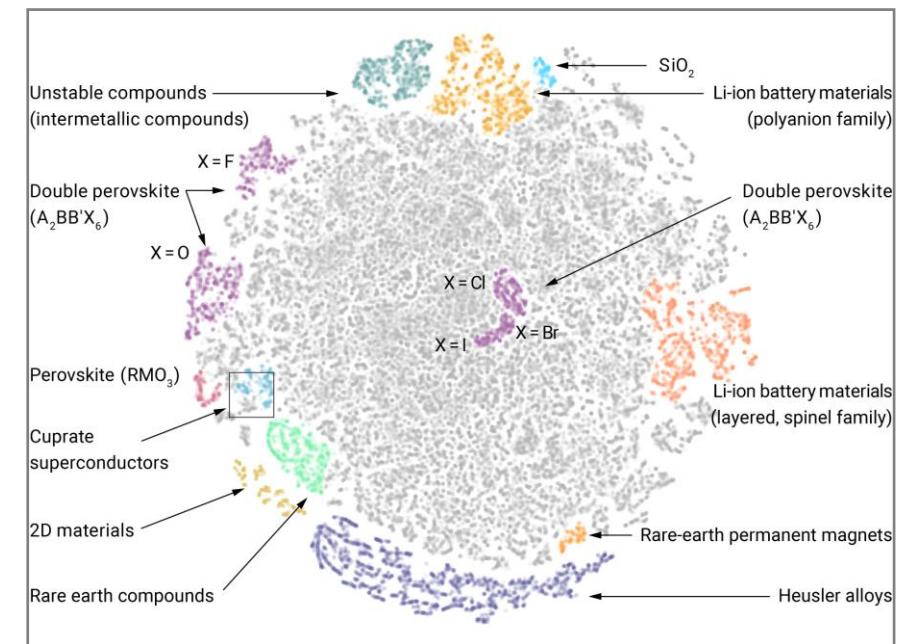
3. NeSF: NeRF-based Crystal Structure Decoder

- N. Chiba, Y. Suzuki, T. Taniai, et al. *Commun Mater* 4, 106 (2023).

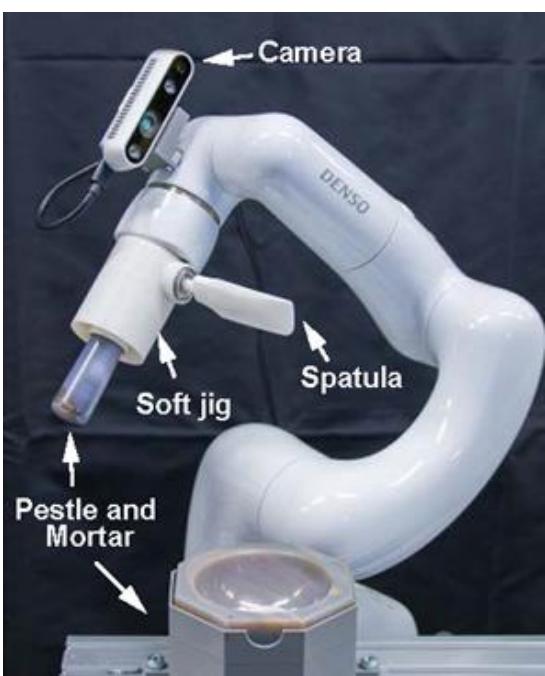
4. Crystalformer: Infinitely Connected Attention for Periodic Structure Encoding

- T. Taniai, R. Igarashi, Y. Suzuki, et al. *ICLR 2024*.

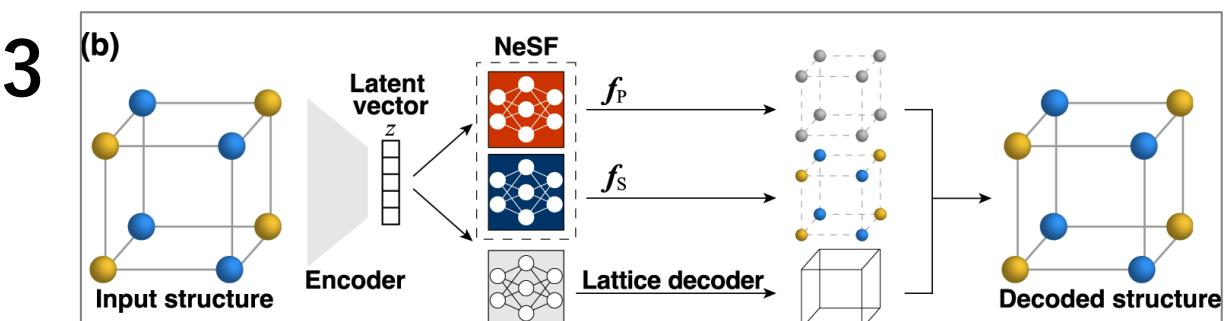
1



2



3



4

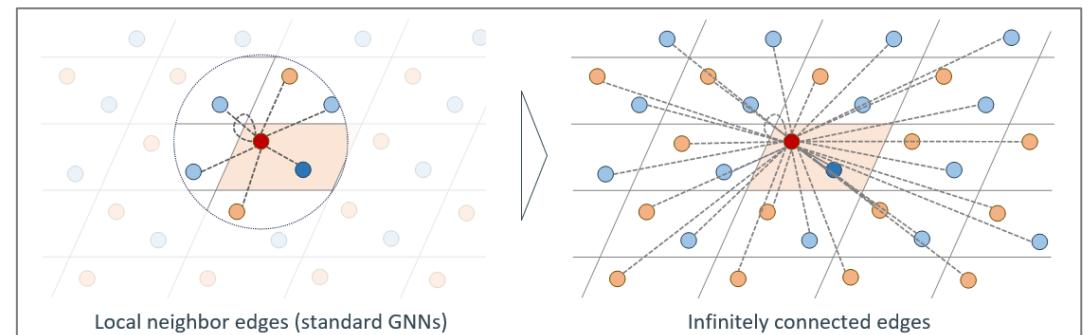


Table of Contents

- **Overview of LLMs**
- Various techniques for LLM
- Practical LLM Tutorial

言語モデル (Language Models)とは



- ある単語の系列 (÷文章) がどれくらい発生しやすいかをモデル化したもの
- 単語の系列を x_1, x_2, \dots, x_L に、その生成確率 $p(x_1, x_2, \dots, x_L)$ を割り当てる確率モデル p のこと

$p(\text{日本}, \text{の}, \text{首都}, \text{は}, \text{東京}) = 0.02$

$p(\text{日本}, \text{の}, \text{首都}, \text{は}, \text{パリ}) = 0.00001$

$p(\text{東京}, \text{の}, \text{首都}, \text{は}, \text{日本}) = 0.0005$

- 様々な言語タスクがこの生成確率の推定問題として扱うことができる
例：翻訳（ある英語文に続くのにふさわしい日本語は？）
例：QA（ある質問に続くのにふさわしい答えは？）
- 生成確率をどう求めるか？が言語モデル技術的な問題の一つ

自己回帰言語モデル (Autoregressive Language Models) M

- $p(x_1, x_2, \dots, x_L)$ を条件分布の積として表現する

$$p(x_1, x_2, \dots, x_L) = p(x_1)p(x_2|x_1) \cdots p(x_L|x_1, x_2, \dots, x_{L-1})$$

- このように確率の連鎖律で分解したモデルを特に自己回帰言語モデルと呼ぶ
- 条件付き確率がわかると、生成することもできる

$$p(\text{東京} | \text{日本}, \text{の}, \text{首都}, \text{は}) = \mathbf{0.2}$$

$$p(\text{パリ} | \text{日本}, \text{の}, \text{首都}, \text{は}) = 0.001$$

⋮

$$p(\text{カイロ} | \text{日本}, \text{の}, \text{首都}, \text{は}) = 0.0005$$

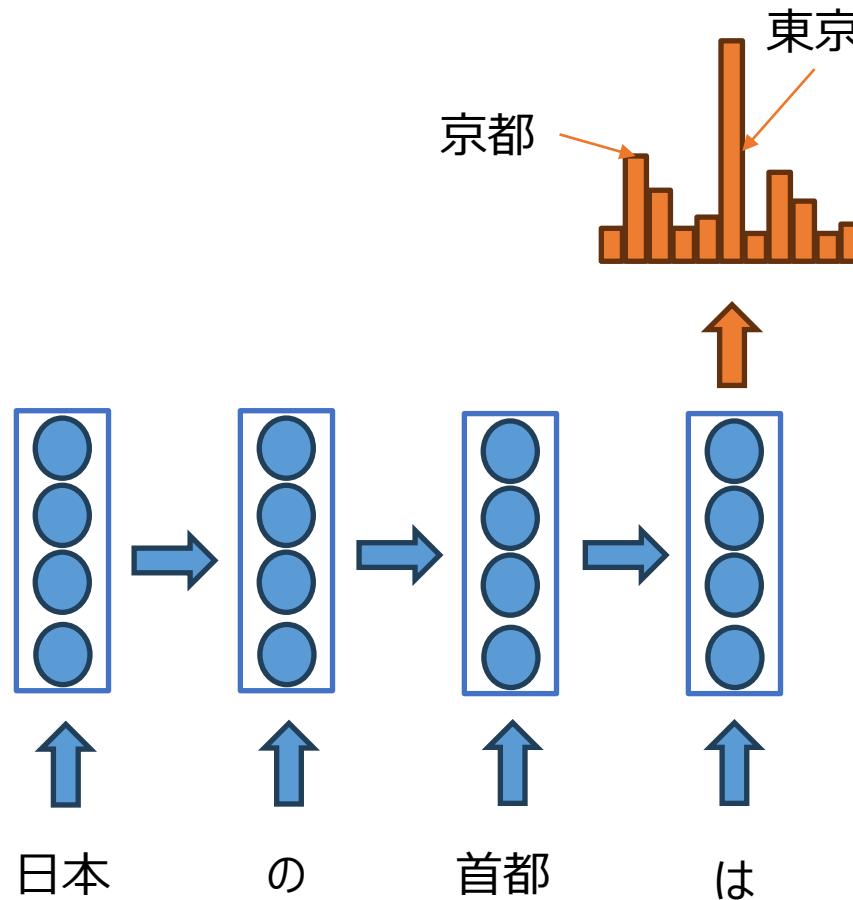
日本の首都は → **東京**

- この条件付き確率をどう求めるか？

ニューラル言語モデル

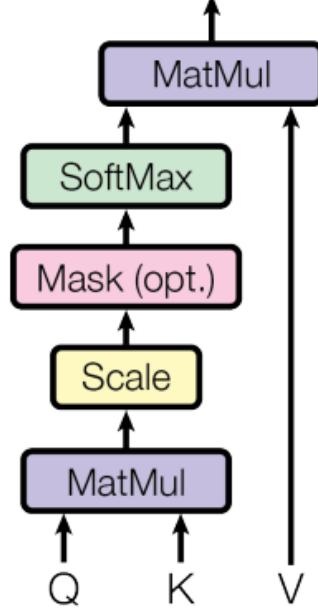


- 条件付き確率を何らかのニューラルネットで推定したモデル
- 他の学習と同様尤度を最大化するように訓練（誤差逆伝播）

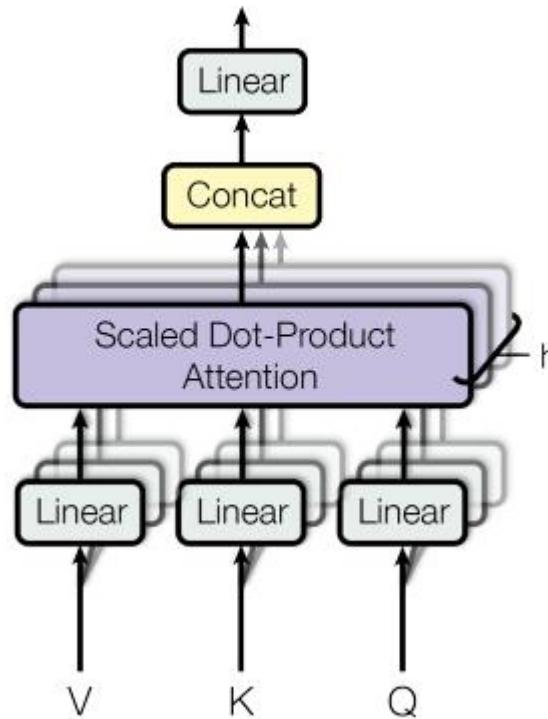


Transformer

Scaled Dot-Product Attention



Multi-Head Attention



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

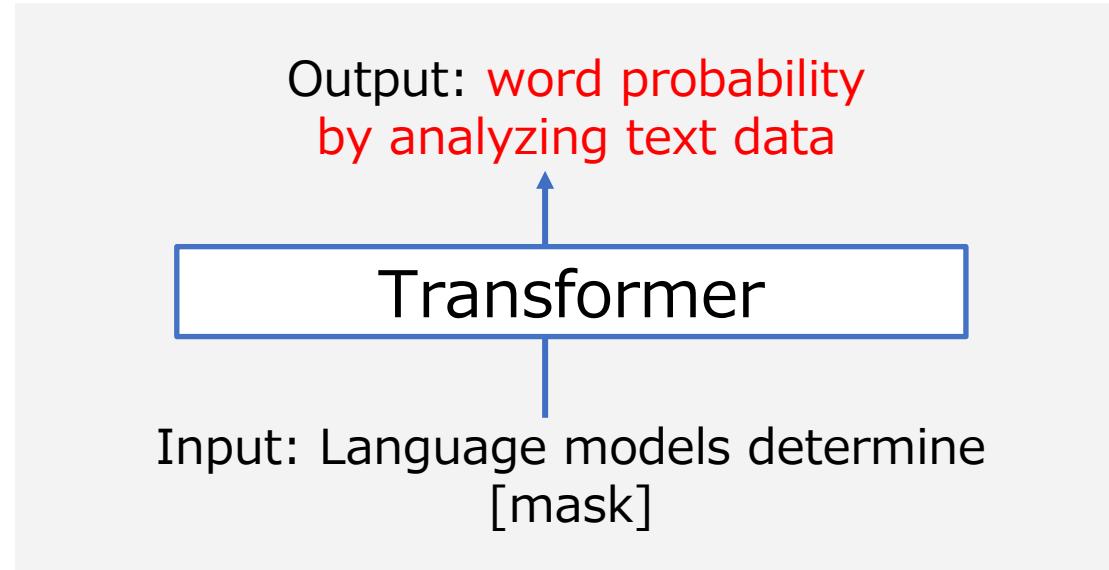
Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [15]	23.75			
Deep-Att + PosUnk [32]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [31]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [8]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [26]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [32]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [31]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [8]	26.36	41.29	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1		$3.3 \cdot 10^{18}$
Transformer (big)	28.4	41.0		$2.3 \cdot 10^{19}$

- Googleを中心とした研究チームが2017年に発表
- Self Attentionを中心としたネットワーク構造（左）
主に翻訳等の教師あり学習で性能検証（右）
例：英語文 → Transformer → ドイツ語文
となるように誤差逆伝播で訓練

[1] Ashish Vaswani et al. (2017) “[Attention Is All You Need](#)” NeurIPS 2017 より引用

Generative Pretraining Transformer (GPT)

Pre-training (事前学習)

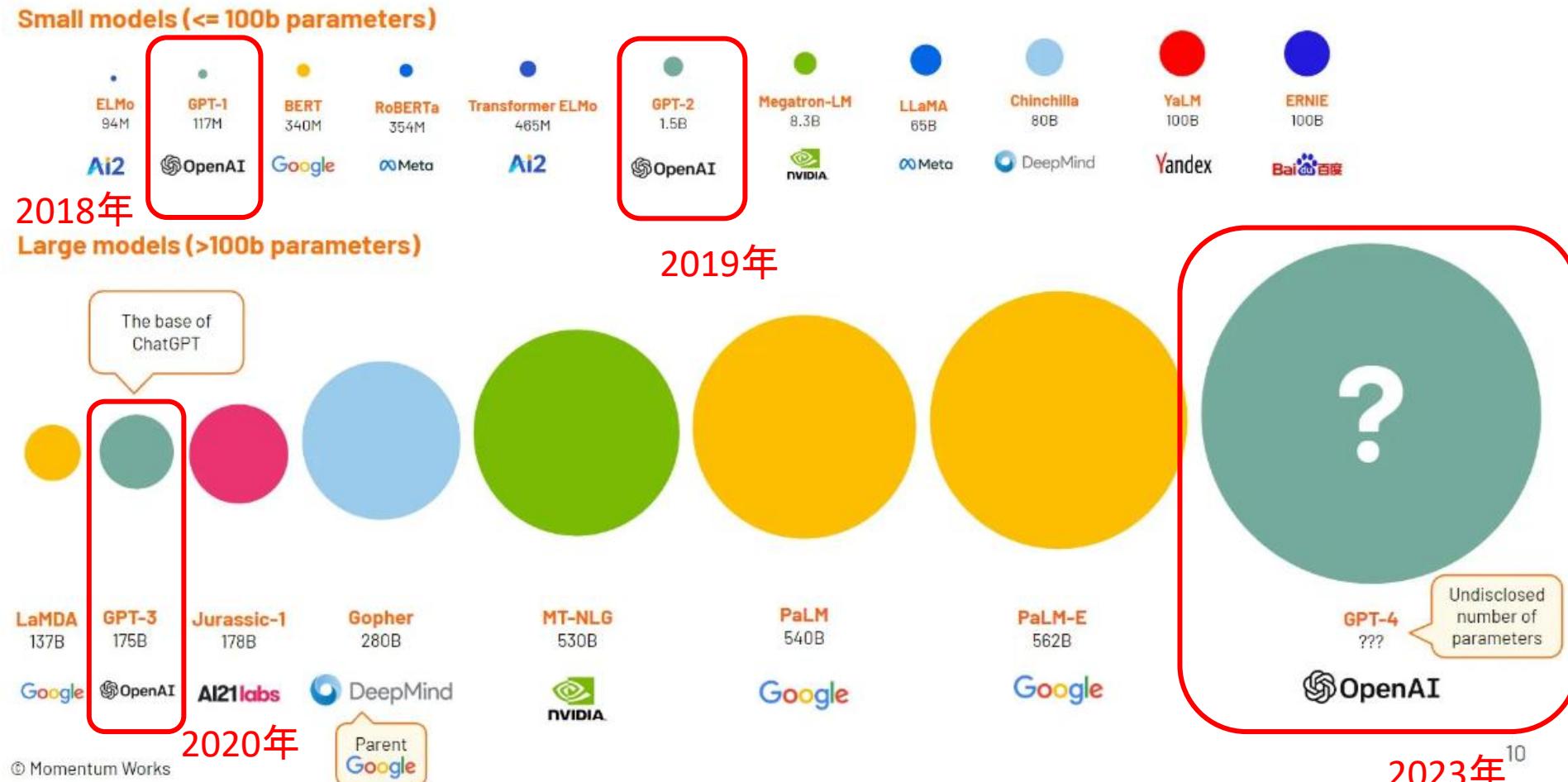


Original: Language models determine word probability by analyzing text data

- OpenAIにより2018年に発表されたモデル
- 事前学習にTransformerを利用
(Transformerを使った言語モデル)
- 具体的には次に来る単語をTransformerで予測するように学習（左図）
Book Corpusという未発表書籍を利用
- GPT, GPT-2, GPT-3とバージョンを経るごとに学習データ数やモデルサイズが増加

[2] Alec Radford et al. (2018) "[Improving Language Understanding by Generative Pre-training](#)" を参考

Transformerを使った言語モデル

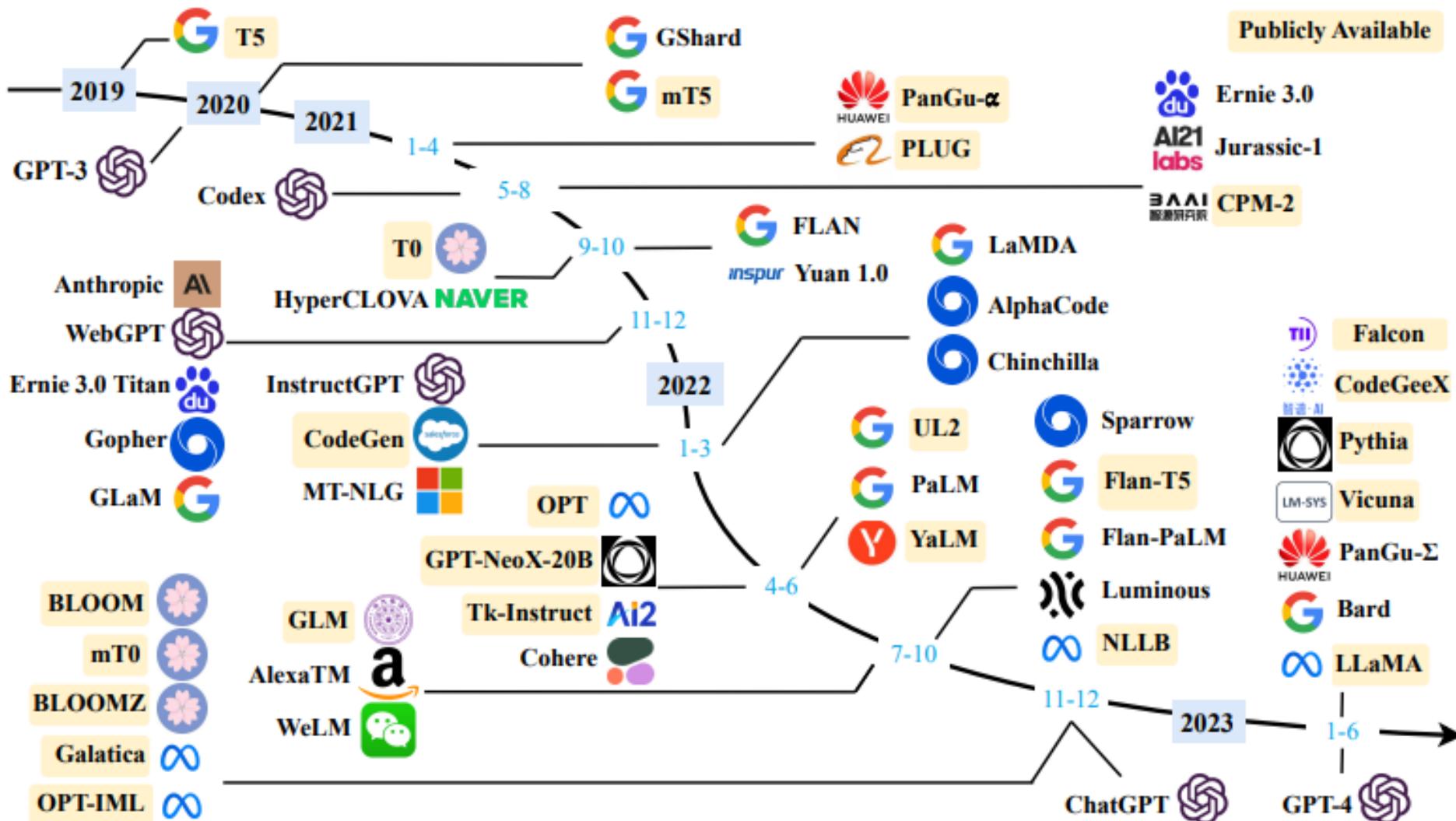


基本的にはいずれも2017年に発明されたTransformerと呼ばれる構造を利用.
GPT-3登場以降、米国企業を中心に複数の研究機関が独自の大規模言語モデルを開発.

[3] Momentum Works 2023 “[The future by ChatGPT](#)” より引用し、一部改変



2020年のGPT-3登場後、2022年後半から加速度的に増加。

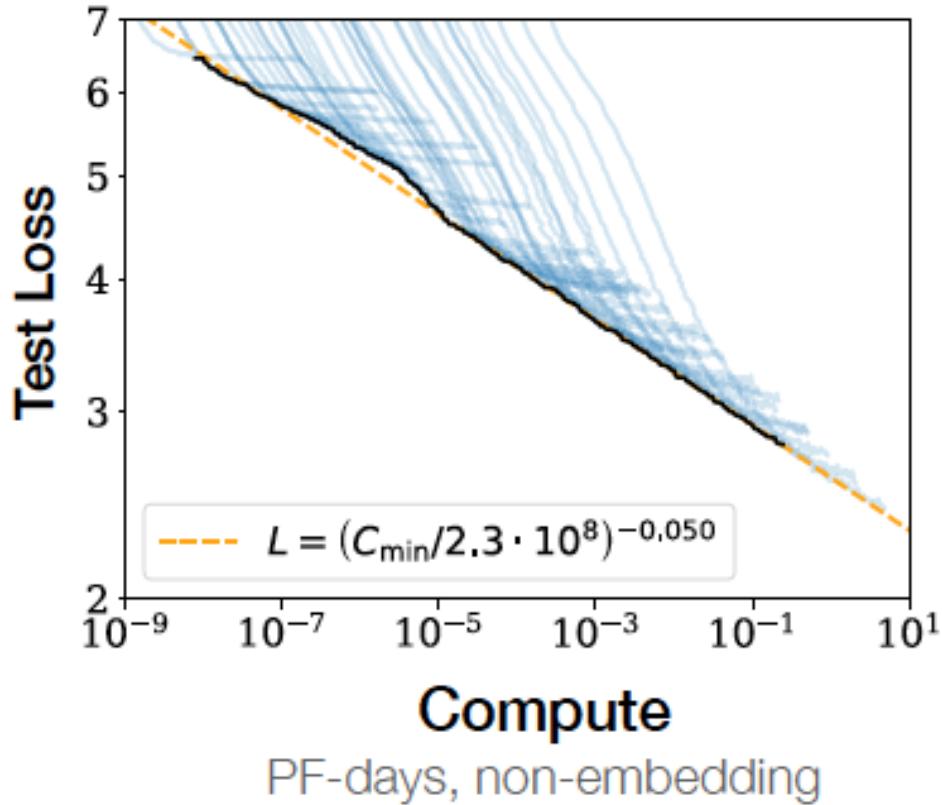


[4] Wayne Xin Zhao et al. (2023), ["A Survey of Large Language Models"](#) より引用

なぜいまLLMを学ぶのか？ 1. Scaling and Emergence



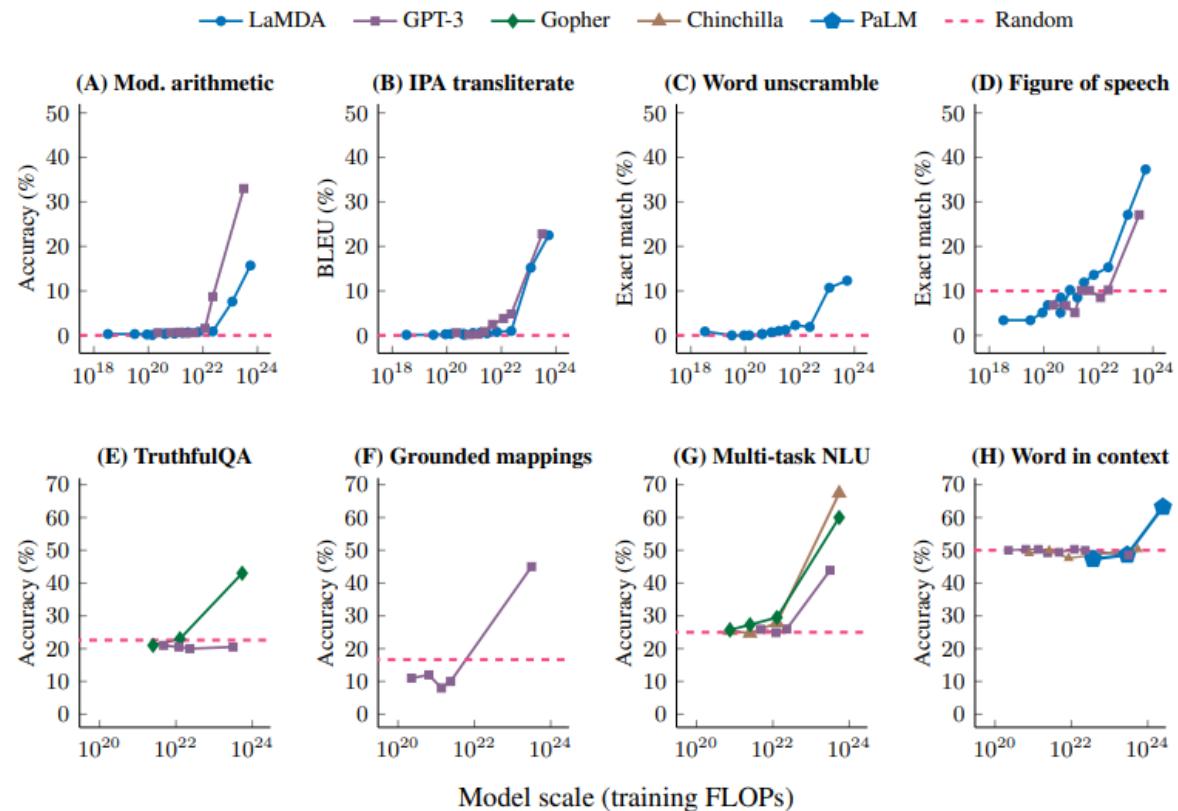
Scaling Law



3つの変数に関するべき乗に従って上がる。

計算資源 C , データセットサイズ D , パラメータ数 N

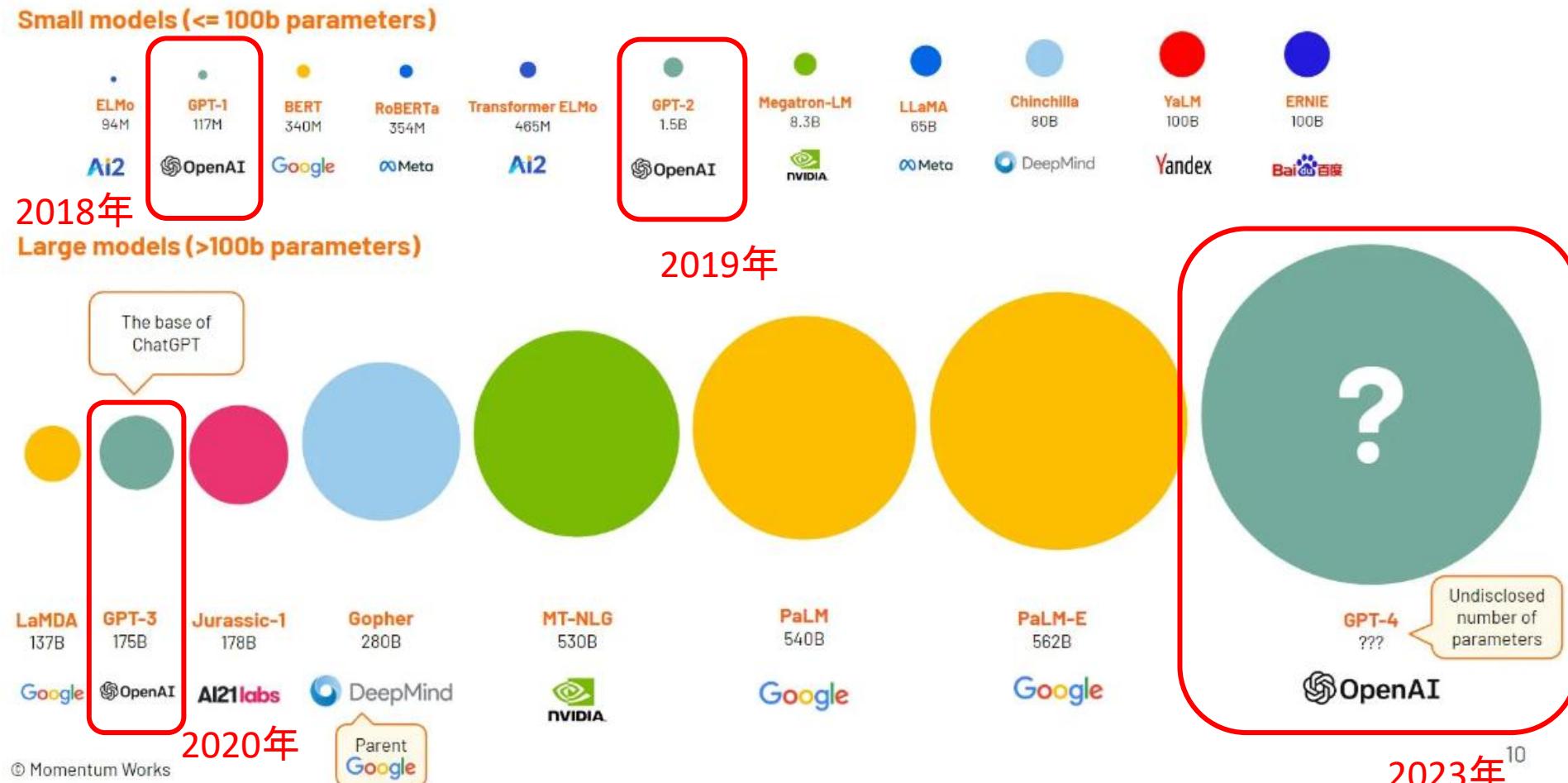
Emergent Ability



モデルサイズが巨大なときのみ解けるタスクが存在

- [5] Jared Kaplan et al. (2020), [“Scaling Laws for Neural Language Models”](#) より引用(左図)
- [6] Jason Wei et al. (2022), [“Emergent Abilities of Large Language Models”](#) より引用(右図)

Transformerを使った言語モデル（再掲）



基本的にはいずれも2017年に発明されたTransformerと呼ばれる構造を利用.
GPT-3登場以降、米国企業を中心に複数の研究機関が独自の大規模言語モデルを開発.

[3] Momentum Works 2023 “[The future by ChatGPT](#)” より引用し、一部改変



GPT-3の学習データ量

GPT-3の事前学習トークン数

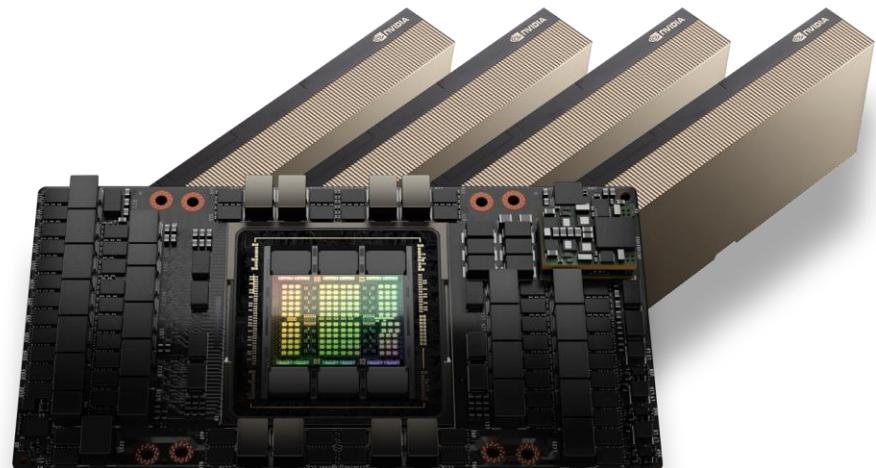
Dataset	Quantity (tokens)
Common Crawl (filtered)	410 billion
WebText2	19 billion
Books1	12 billion
Books2	55 billion
Wikipedia	3 billion

- 約5000億トークン*のテキストを利用
 - *トークンとは、言語AIが処理する単位。
日本語だと大体1文字1トークン
- *書籍でいうとGPT-3は約500万冊に相当
 - 参考：東大図書館が約130万冊、
国会図書館が約4700万冊
- *リーク情報によるとGPT-4は約1.3億冊に相当

[7] Tom Brown et al. (2020), “[Language Models are Few-Shot Learners](#)”, NeurIPS2020 より引用

■ 補足 | 必要な計算能力も大規模化

GPU (H100, A100, V100など)



GPT3相当の場合 : A100 × **1200基 × 30日**

GPT4相当の場合 : A100 × **25000基 × 100日**

よく利用されるGPUクラスタ*

- ABCI (産総研)
960基のA100 GPU
(国内最大規模)



- 海外のIaaS
AWS (Amazon), GCP (Google), Azure (Microsoft)



*GPUを搭載した複数の計算機をまとめて提供するシステムこと

(GPUの画像) <https://www.scsk.jp/sp/nvidia/ai-server/index.htm>^[8]

(ABCIのロゴ) <https://abci.ai/ja/>^[9]

(AWSのロゴ) <https://aws.amazon.com/jp/>^[10]

(Google Cloudのロゴ) <https://dev.classmethod.jp/referencecat/classmethod-google-cloud-advent-calendar-2021/>^[11]

(Azureのロゴ) <https://1000logos.net/microsoft-azure-logo>^[12]

なぜいまLLMを学ぶのか？ 2. 汎用性 (Prompting / In-Context Learning)



Pre-training (事前学習)

Output: word probability
by analyzing text data

LLMs (Transformer)

Input: Language models determine
[mask]

Original: Language models determine word
probability by analyzing text data

[7] Tom Brown et al. (2020), “[Language Models are Few-Shot Learners](#)”より引用

Translation (Few-Shot)

1	Translate English to French:	task description
2	sea otter => loutre de mer	examples
3	peppermint => menthe poivrée	
4	plush girafe => girafe peluche	
5	cheese =>	

Translation (Zero-Shot)

1	Translate English to French:	task description
2	cheese =>	prompt

Summarization (Zero-Shot)

- Starting with “TL;DR” drastically improves the performance
- Many other examples

Pre-train, Prompt, Predict



Paradigm	Engineering	Task Relation
a. Fully Supervised Learning (Non-Neural Network)	Features (e.g. word identity, part-of-speech, sentence length)	
b. Fully Supervised Learning (Neural Network)	Architecture (e.g. convolutional, recurrent, self-attentional)	
c. Pre-train, Fine-tune	Objective (e.g. masked language modeling, next sentence prediction)	
d. Pre-train, Prompt, Predict	Prompt (e.g. cloze, prefix)	

従来

タスクごとにモデルを学習
(NN以外)

タスクごとにモデルを学習
(NN)

モデルを共有して学習
(Fine-Tuning)

モデルを固定して指示を変更
(Prompting)

現代

[9] Pengfei Liu et al. (2021),
[“Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing”](#) より引用



■ 補足 | Foundation Model (基盤モデル)

On the Opportunities and Risks of Foundation Models

Rishi Bommasani* Drew A. Hudson Ehsan Adeli Russ Altman Simran Arora
Sydney von Arx Michael S. Bernstein Jeannette Bohg Antoine Bosselut Emma Brunskill
Erik Brynjolfsson Shyamal Buch Dallas Card Rodrigo Castellon Niladri Chatterji
Annie Chen Kathleen Creel Jared Quincy Davis Dorottya Demszky Chris Donahue
Moussa Doumbouya Esin Durmus Stefano Ermon John Etchemendy Kawin Ethayarajh
Li Fei-Fei Chelsea Finn Trevor Gale Lauren Gillespie Karan Goel Noah Goodman
Shelby Grossman Neel Guha Tatsunori Hashimoto Peter Henderson John Hewitt
Daniel E. Ho Jenny Hong Kyle Hsu Jing Huang Thomas Icard Saahil Jain
Dan Jurafsky Pratyusha Kalluri Siddharth Karamcheti Geoff Keeling Fereshte Khani
Omar Khattab Pang Wei Koh Mark Krass Ranjay Krishna Rohith Kuditipudi
Ananya Kumar Faisal Ladhak Mina Lee Tony Lee Jure Leskovec Isabelle Levent
Xiang Lisa Li Xuechen Li Tengyu Ma Ali Malik Christopher D. Manning
Suvir Mirchandani Eric Mitchell Zanele Munyikwa Suraj Nair Avanika Narayan
Deepak Narayanan Ben Newman Allen Nie Juan Carlos Niebles Hamed Nilforoshan
Julian Nyarko Giray Ogut Laurel Orr Isabel Papadimitriou Joon Sung Park Chris Piech
Eva Portelance Christopher Potts Aditi Raghunathan Rob Reich Hongyu Ren
Frieda Rong Yusuf Roohani Camilo Ruiz Jack Ryan Christopher Ré Dorsa Sadigh
Shiori Sagawa Keshav Santhanam Andy Shih Krishnan Srinivasan Alex Tamkin
Rohan Taori Armin W. Thomas Florian Tramèr Rose E. Wang William Wang Bohan Wu
Jiajun Wu Yuhuai Wu Sang Michael Xie Michihiro Yasunaga Jiaxuan You Matei Zaharia
Michael Zhang Tianyi Zhang Xikun Zhang Yuhui Zhang Lucia Zheng Kaitlyn Zhou
Percy Liang*¹

Center for Research on Foundation Models (CRFM) – Stanford University

- 2021/8/16初出のホワイトペーパーで登場した言葉
- Stanfordの研究機関の名称にもなっている（青枠）
- 多様なタスクに適用可能な巨大モデルによるパラダイムシフト

(Abstractより抜粋)

*"AI is undergoing a **paradigm shift with the rise of models** (e.g., BERT, DALL-E, GPT-3) that are trained on broad data at scale and are adaptable to a wide range of downstream tasks. We call these models **foundation models** to underscore their critically central yet incomplete character"*

[10] Rishi Bommasani et al. (2021) "[On the Opportunities and Risks of Foundation Models](#)" より引用し、一部改変

GPT-4の専門知識 (“GPT-4 Technical Report”, 2023)



Exam	GPT-4	GPT-4 (no vision)	GPT-3.5
Uniform Bar Exam (MBE+MEE+MPT)	298 / 400 (~90th)	298 / 400 (~90th)	213 / 400 (~10th)
LSAT	163 (~88th)	161 (~83rd)	149 (~40th)
SAT Evidence-Based Reading & Writing	710 / 800 (~93rd)	710 / 800 (~93rd)	670 / 800 (~87th)
SAT Math	700 / 800 (~89th)	690 / 800 (~89th)	590 / 800 (~70th)
Graduate Record Examination (GRE) Quantitative	163 / 170 (~80th)	157 / 170 (~62nd)	147 / 170 (~25th)
Graduate Record Examination (GRE) Verbal	169 / 170 (~99th)	165 / 170 (~96th)	154 / 170 (~63rd)
Graduate Record Examination (GRE) Writing	4 / 6 (~54th)	4 / 6 (~54th)	4 / 6 (~54th)
USABO Semifinal Exam 2020	87 / 150 (99th - 100th)	87 / 150 (99th - 100th)	43 / 150 (31st - 33rd)
USNCO Local Section Exam 2022	36 / 60	38 / 60	24 / 60
Medical Knowledge Self-Assessment Program	75 %	75 %	53 %
Codeforces Rating	392 (below 5th)	392 (below 5th)	260 (below 5th)
AP Art History	5 (86th - 100th)	5 (86th - 100th)	5 (86th - 100th)
AP Biology	5 (85th - 100th)	5 (85th - 100th)	4 (62nd - 85th)
AP Calculus BC	4 (43rd - 59th)	4 (43rd - 59th)	1 (0th - 7th)
AP Chemistry	4 (71st - 88th)	4 (71st - 88th)	2 (22nd - 46th)
AP English Language and Composition	2 (14th - 44th)	2 (14th - 44th)	2 (14th - 44th)
AP English Literature and Composition	2 (8th - 22nd)	2 (8th - 22nd)	2 (8th - 22nd)
AP Environmental Science	5 (91st - 100th)	5 (91st - 100th)	5 (91st - 100th)
AP Macroeconomics	5 (84th - 100th)	5 (84th - 100th)	2 (33rd - 48th)
AP Microeconomics	5 (82nd - 100th)	4 (60th - 82nd)	4 (60th - 82nd)
AP Physics 2	4 (66th - 84th)	4 (66th - 84th)	3 (30th - 66th)
AP Psychology	5 (83rd - 100th)	5 (83rd - 100th)	5 (83rd - 100th)
AP Statistics	5 (85th - 100th)	5 (85th - 100th)	3 (40th - 63rd)
AP US Government	5 (88th - 100th)	5 (88th - 100th)	4 (77th - 88th)
AP US History	5 (89th - 100th)	4 (74th - 89th)	4 (74th - 89th)
AP World History	4 (65th - 87th)	4 (65th - 87th)	4 (65th - 87th)
AMC 10 ³	30 / 150 (6th - 12th)	36 / 150 (10th - 19th)	36 / 150 (10th - 19th)
AMC 12 ³	60 / 150 (45th - 66th)	48 / 150 (19th - 40th)	30 / 150 (4th - 8th)
Introductory Sommelier (theory knowledge)	92 %	92 %	80 %
Certified Sommelier (theory knowledge)	86 %	86 %	58 %
Advanced Sommelier (theory knowledge)	77 %	77 %	46 %
Leetcode (easy)	31 / 41	31 / 41	12 / 41
Leetcode (medium)	21 / 80	21 / 80	8 / 80
Leetcode (hard)	3 / 45	3 / 45	0 / 45

- OpenAIにより2023年に発表されたモデル
(詳細は未公開、リーク情報はあり)

- 司法試験やSAT/GREなどの多様な試験で
好成績

例: Uniform Bar Examでは298/400
(~90th)

例 : GRE (Quantitative)が163/179
(~80th)

- 一方コーディング能力などではまだ低い
スコア

[11] OpenAI 2023 “[GPT-4 Technical Report](#)” より引用し、一部改変

- 言語モデルとは単語列の生成確率をモデル化したもの
 - 自己回帰言語モデル / ニューラル言語モデル / GPT
 - 単に確率的に次の単語を選ぶことの繰り返しで、これだけ知的な文章生成ができるのが驚くべきこと
- なぜいま言語モデルなのか ?
 - 1. モデル, データ, 計算量のスケールによりできることが急速に広がっている
 - 2. Promptingにより, 単一モデルで様々なことができるよう (言語モデルの汎用性)
 - 3. 言語モデルの発展が他の領域にも影響を与えている

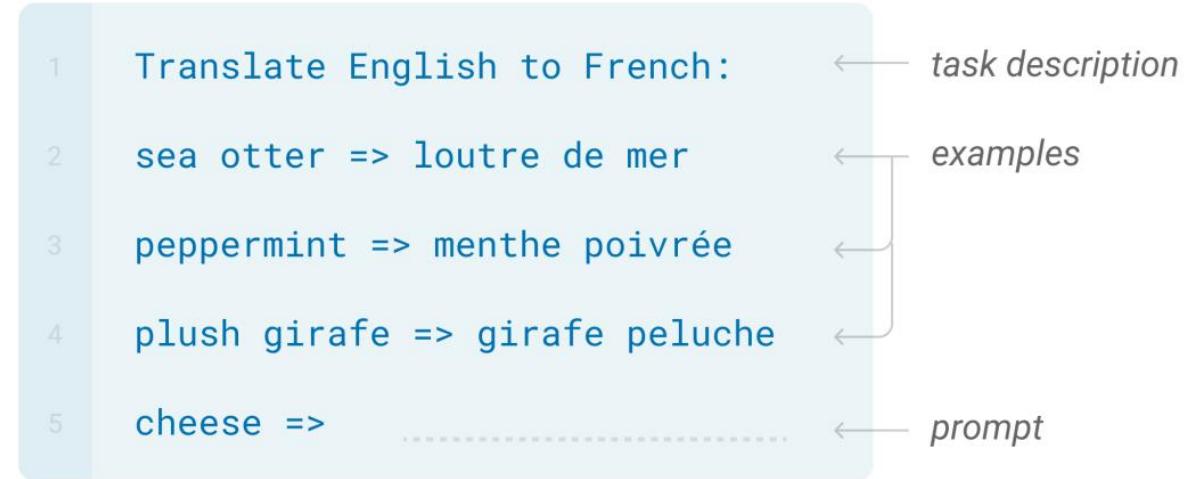
目次

- LLMの概要
- 様々なテクニック
 - プロンプティング
 - フайнチューニング
- フайнチューニングのチュートリアル

プロンプティング (Prompting)

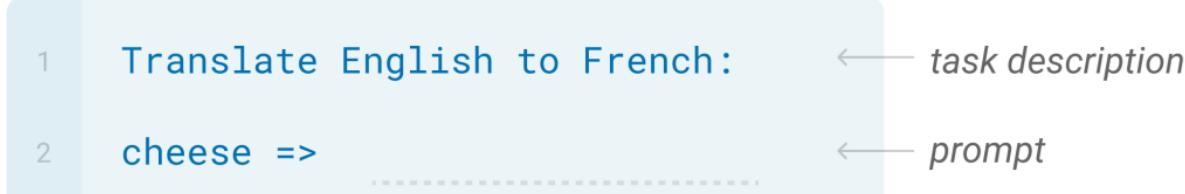
特定の機能の発生を促進 (prompt) するような言語モデルに入力するコンテキスト文

Demonstration (Few-Shot)



与える事例を変えれば異なる
ことができる
(例：ポジネガ判定)

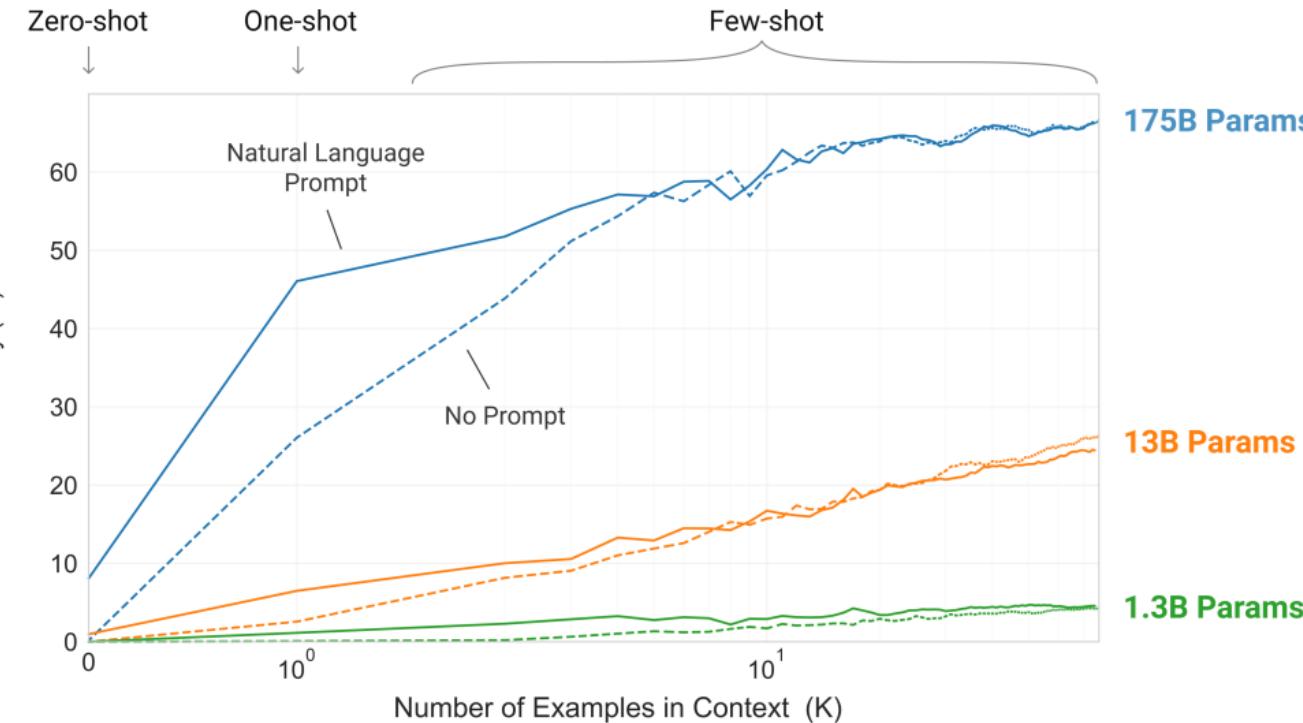
Instruction (Zero-Shot)



[7] Tom Brown et al. (2020), “[Language Models are Few-Shot Learners](#)” より引用

加えるとある機能が強化される文字列
例 : tl;drをつけると要約性能が上がる [1]
例 : According toをつけると知識を参照してくれるようになる [2]
中間指示 (例 必要な変数を保持してください)
プロンプトエンジニアリング

文脈内学習 (In-Context Learning)によるFew-Shot学習

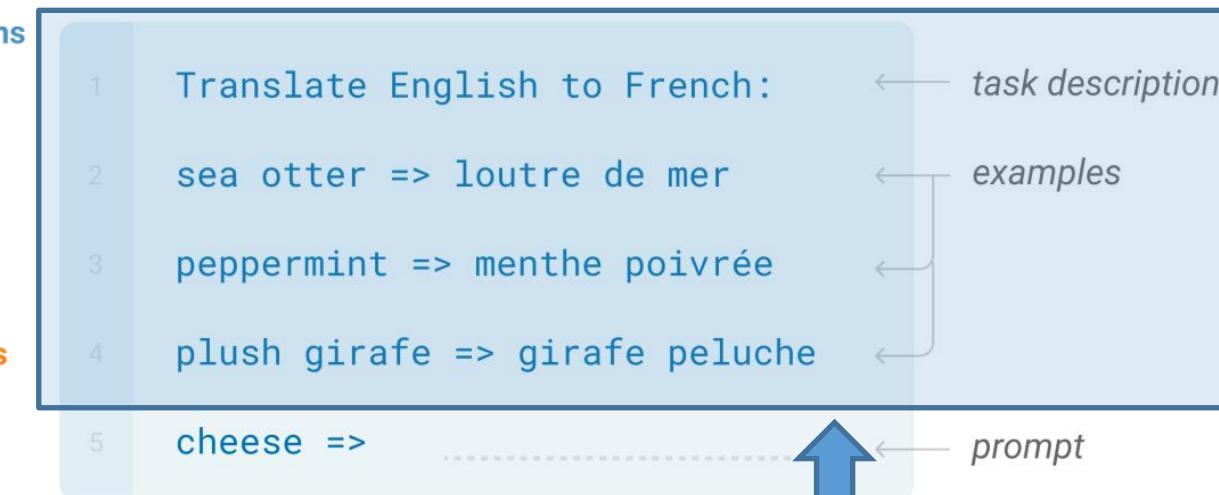


175B Params

13B Params

1.3B Params

Demonstration (Few-Shot)



文脈 (Context)

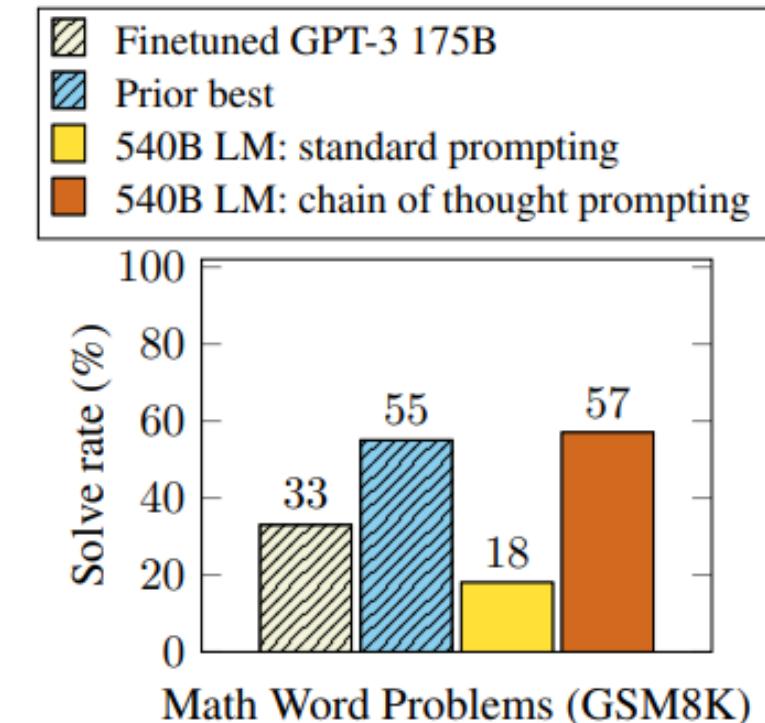
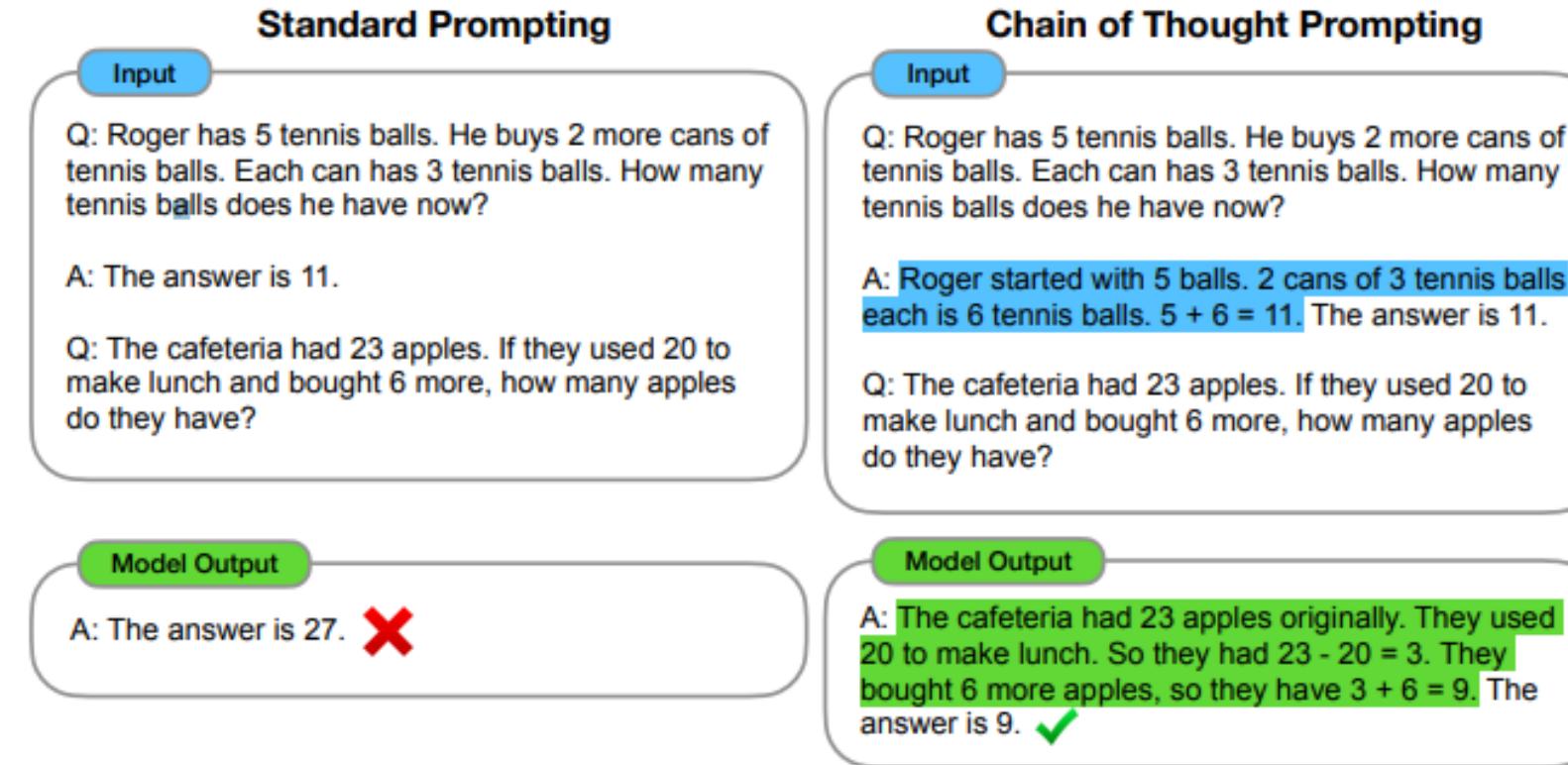
[7] Tom Brown et al. (2020), “[Language Models are Few-Shot Learners](#)”より引用し、一部改変

特にモデルが大規模な場合Few-Shotのデモンストレーションの追加で性能が大幅に上ることが多い。

文脈から学習するため、文脈内学習 (In-Context Learning)と呼ぶ。

Chain-of-Thought (CoT) Prompting

※ GSM8kは9-12歳の正解率が60%。



[20] Jason Wei et al. (2022), “[Chain of Thought Prompting Elicits Reasoning in Large Language Models](#)” NeurIPS2022 より引用

- Few-Shotの事例の際に思考過程を入れる (Chain of thought prompting)と、新しい質問についても思考過程を明示してくれる。
- 算数の文章題など、従来難しいとされていた推論タスクでも大幅に性能が向上。



Augmented Language Models : 外部ツールを利用する事例

The New England Journal of Medicine is a registered trademark of [QA("Who is the publisher of The New England Journal of Medicine?") → Massachusetts Medical Society] the MMS.

Out of 1400 participants, 400 (or [Calculator(400 / 1400) → 0.29] 29%) passed the test.

The name derives from "la tortuga", the Spanish word for [MT("tortuga") → turtle] turtle.

The Brown Act is California's law [WikiSearch("Brown Act") → The Ralph M. Brown Act is an act of the California State Legislature that guarantees the public's right to attend and participate in meetings of local legislative bodies.] that requires legislative bodies, like city councils, to hold their meetings open to the public.

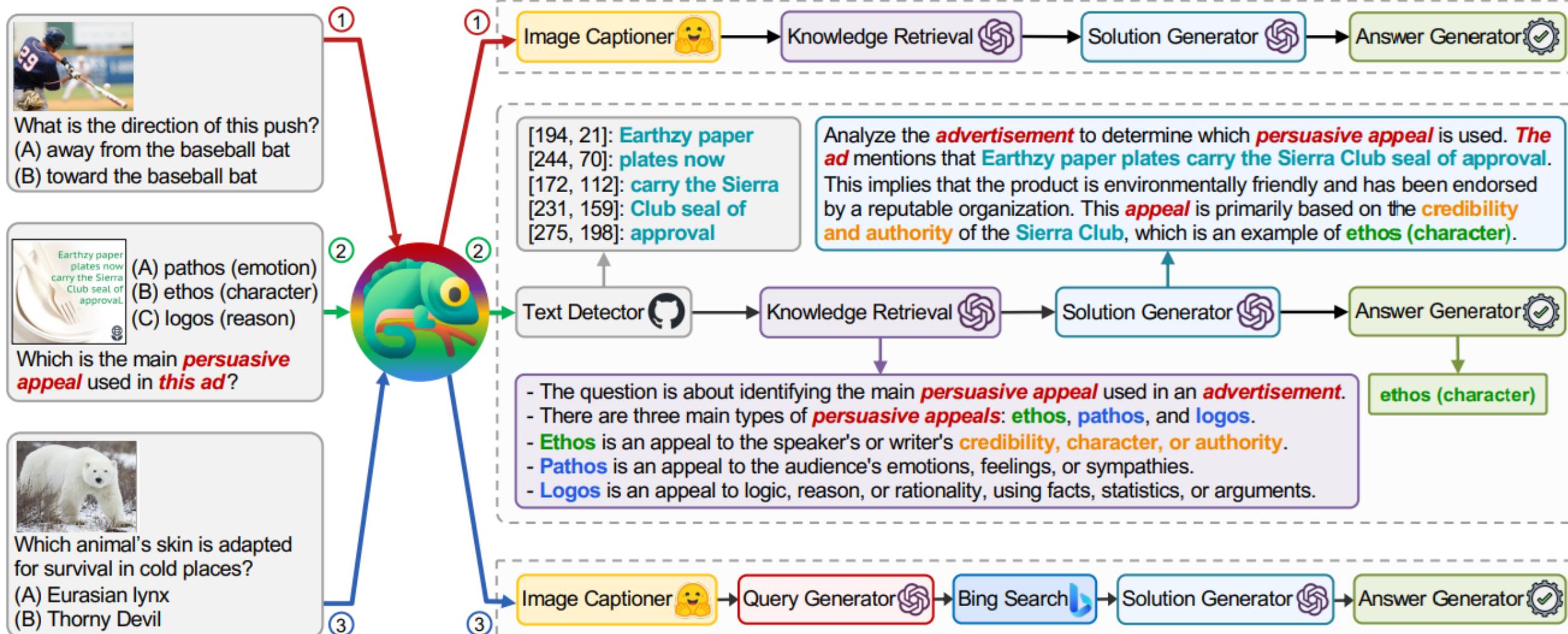
[21] Timo Schick et al. (2023),
["Toolformer: Language Models Can Teach Themselves to Use Tools"](#) より引用

LLMが検索、計算、翻訳など外部のツールを利用する。

- The New England Journal of Medicine の登録商標者は、[QA("The New England Journal of Medicineの発行元は ? ") → Massachusetts Medical Society] MMSです。
- 1400人の参加者のうち、400人つまり [計算機(400/1400)→0.29] 29%が試験に合格した。
- その名前は"la tortuga"に由来しており、それはスペイン語で [MT("tortuga") → 龜] 龜です。

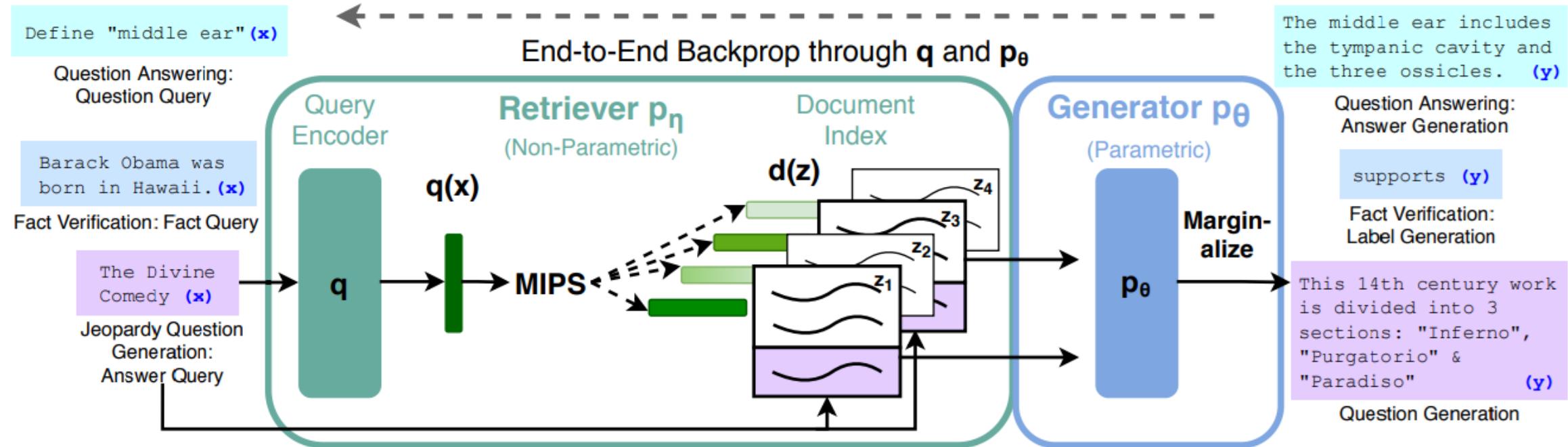


Augmented Language Models : 外部ツールを利用する事例



[22] Pan Lu et al. (2023), “[Chameleon: Plug-and-Play Compositional Reasoning with Large Language Models](#)” より引用

Augmented Language Models : 文書検索をする事例



[23] Patrick Lewis et al. (2020), “[Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks](#)”, NeurIPS2020 より引用

- いわゆるRAG(Retrieval-Augmented Generation)
- 事前にIndex化して蓄積した文章データベースから、問い合わせに類似した文章を取り出し (Retrieveし) 、それをLLMの入力として用いる。
- パラメータの更新をせずとも情報の正確性を上げることが可能。ただしRetrievalの精度に依存する。
- LlamaIndexはこのアイデアを活用している。

Augmented Language Modelsの利点

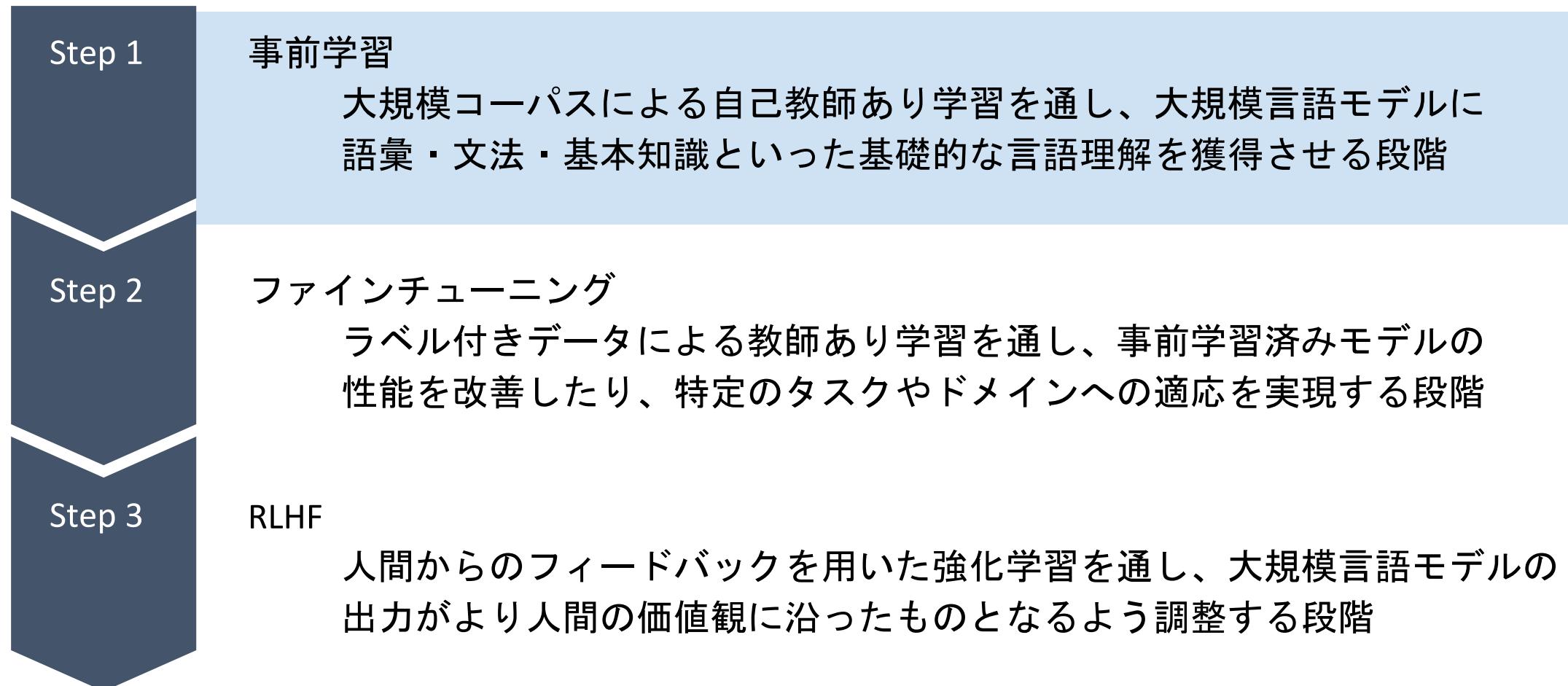


- Truthfulness (真実性)
 - 計算機の利用や情報ソースへのアクセスを可能にすることで Hallucination (幻覚) を軽減
- Estimating and reducing uncertainty (不確実性の推定と低減)
 - LLMだけで生成するか、いつtoolに頼るべきか組み合わせられる
- Interpretability (解釈性)
 - 途中過程を確認できたり回答根拠を出させることで人間にとて解釈可能性が上がる
- Enhanced capabilities (性能改善)
 - 通常のLMに比べ、toolの利用でより人間に役立つ

目次

- LLMの概要
- 様々なテクニック
 - プロンプティング
 - フайнチューニング
- フайнチューニングのチュートリアル

LLM学習フロー

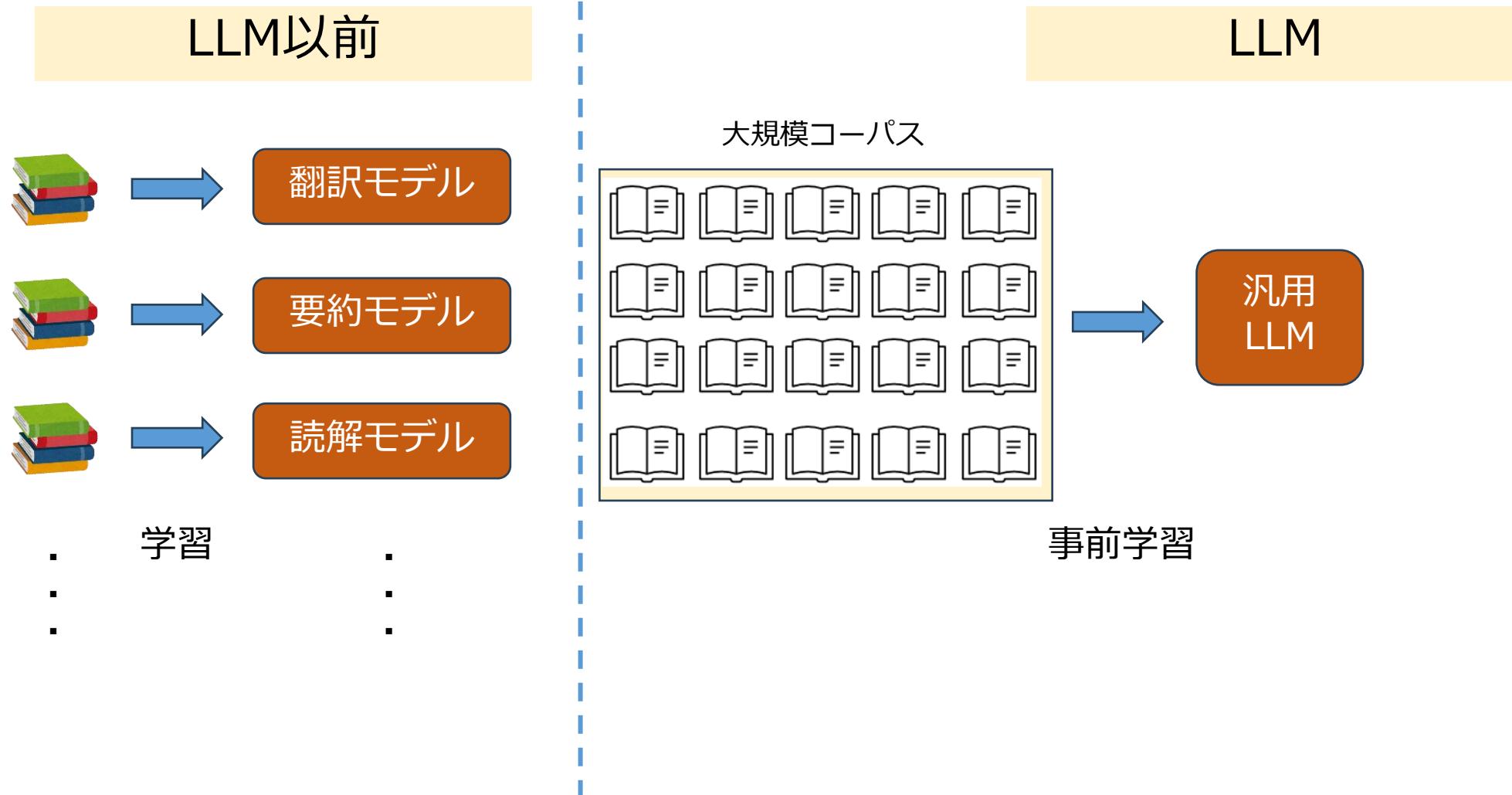


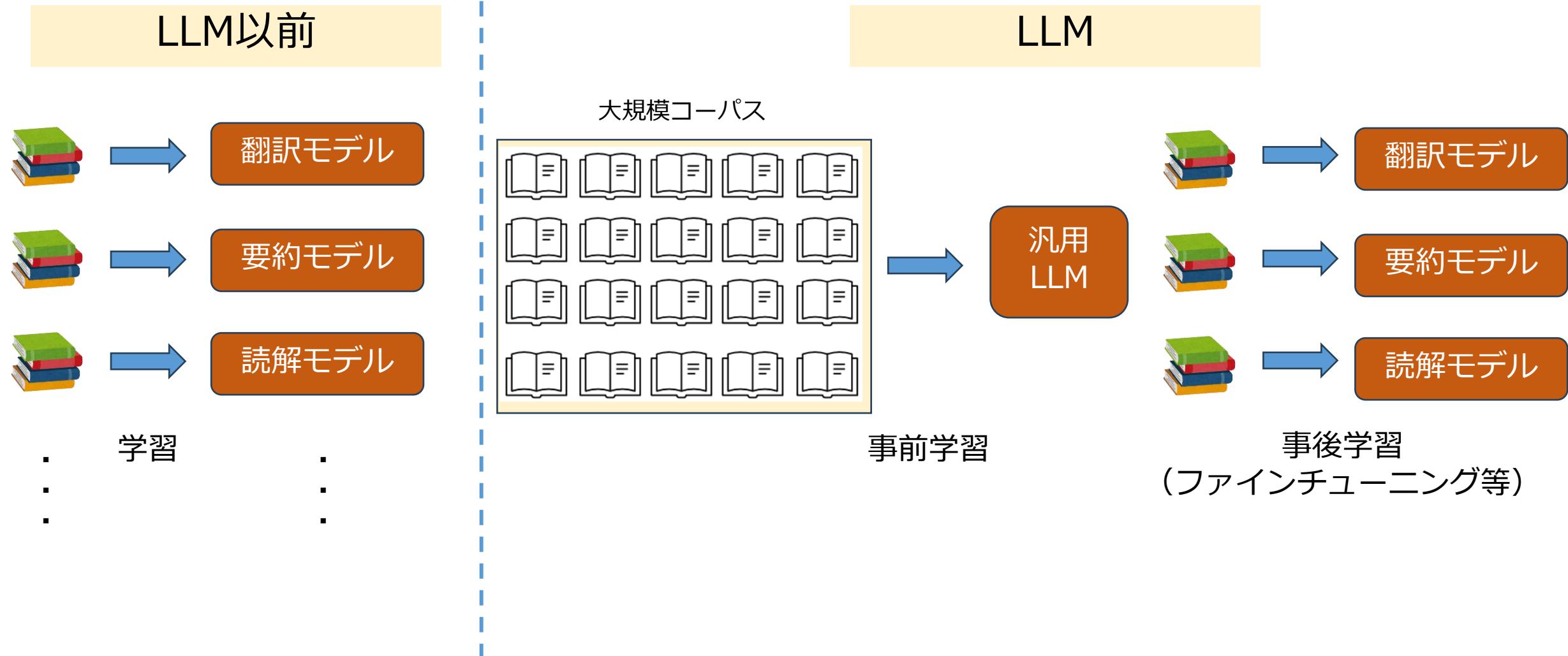
LLM以前

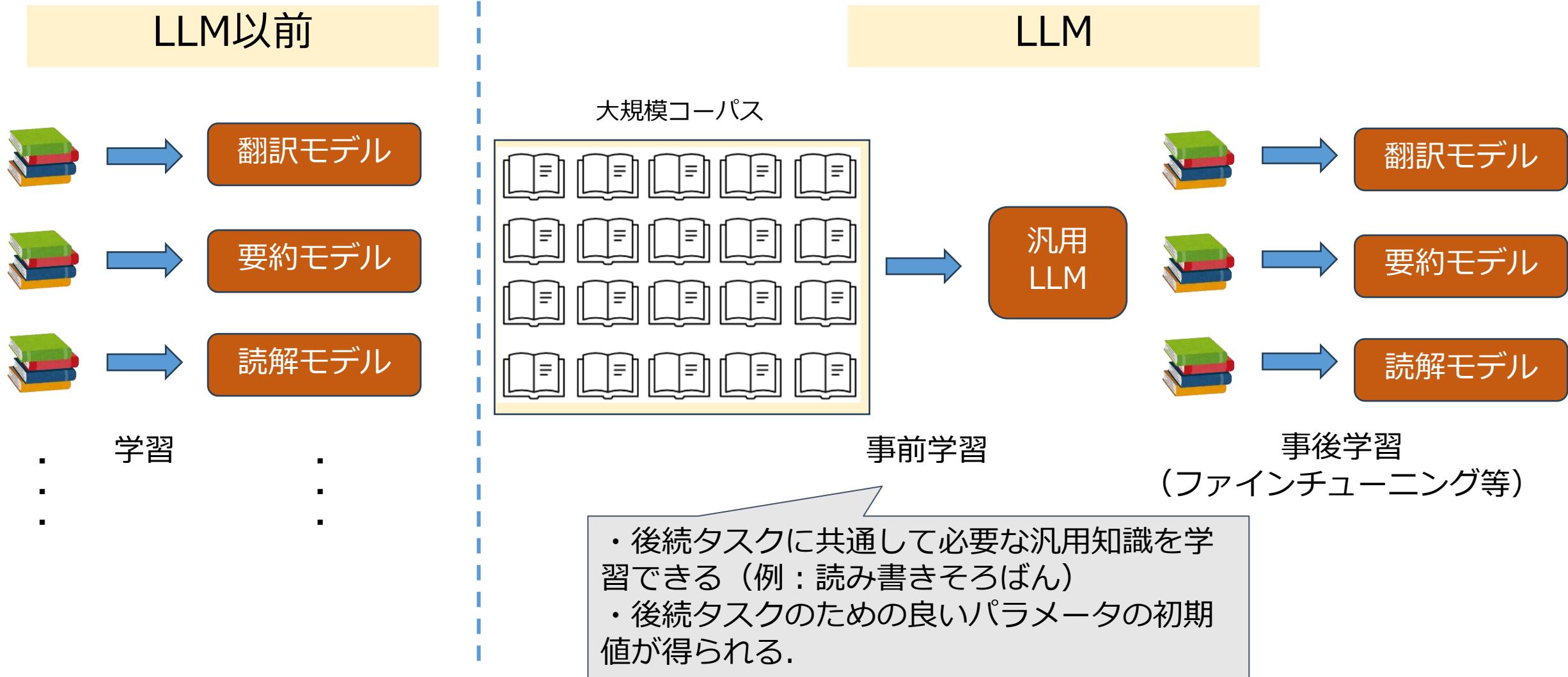


学習

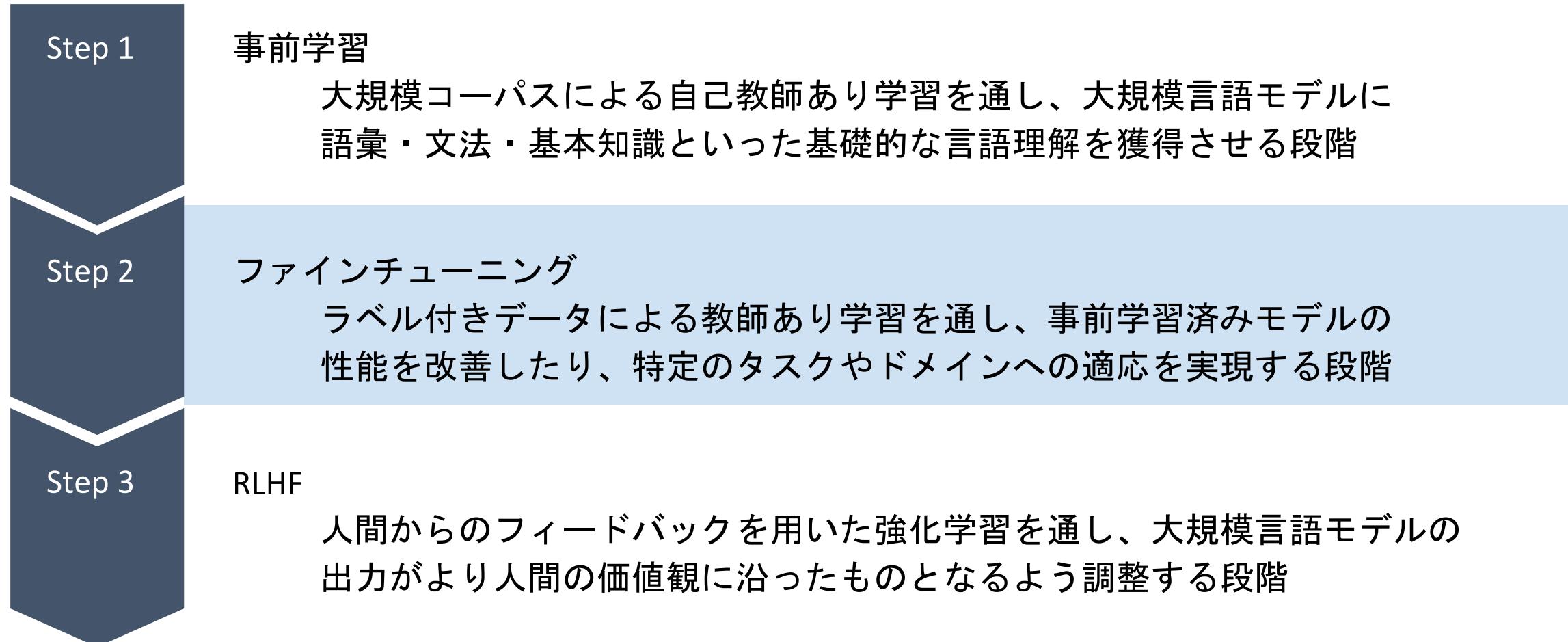
-
-
-







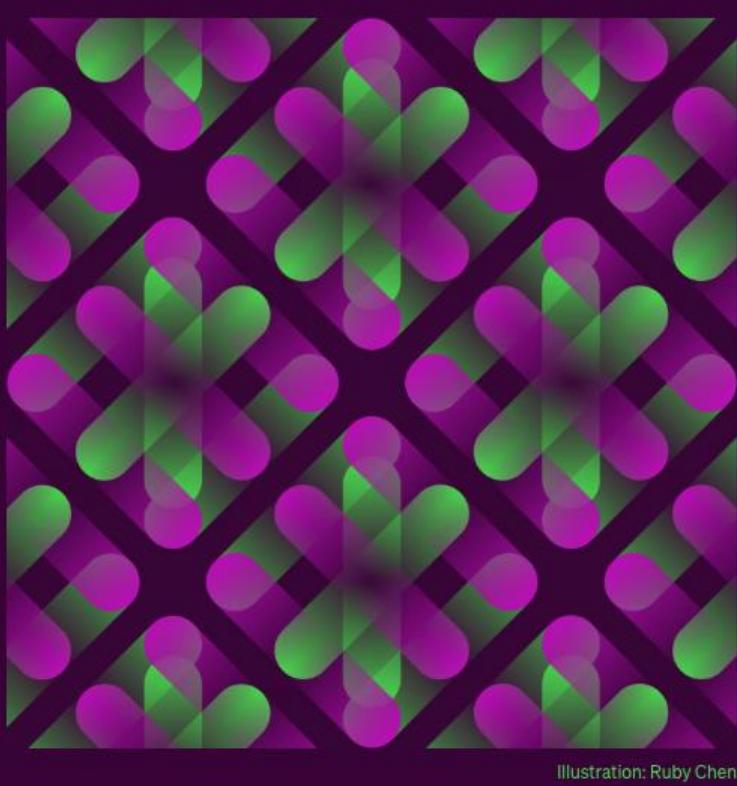
LLM学習フロー



大規模言語モデル Fine-Tuning 事例 | GPT-3.5 Fine-Tuning

GPT-3.5 Turbo fine-tuning and API updates

Developers can now bring their own data to customize GPT-3.5 Turbo for their use cases.



[34] OpenAI “[GPT-3.5 Turbo fine-tuning and API updates](#)” より引用

- 2023年8月22日に OpenAI API で GPT-3.5-Turbo Fine-Tuning 機能がリリース
- 自前の質疑応答データセットで Fine-Tuning を実施可能に
- 以下のユースケースが例示
 - Improved steerability
 - Reliable output formatting
 - Custom tone

大規模言語モデル Fine-Tuning 事例 | Med-PaLM

You are a helpful medical knowledge assistant. Provide useful, complete, and scientifically-grounded answers to common consumer search queries about health.

Question: How do you treat skin redness?

Complete Answer: It depends on the cause of the skin redness. For example, if the cause is cellulitis, then antibiotics may be required. However, this might be inappropriate for other causes of redness such as eczema. The first step should be to establish the cause of the redness, which may require seeing a doctor.

You are a helpful medical knowledge assistant. Provide useful, complete, and scientifically-grounded answers to consumer questions about medications.

Question: Why take simvastatin?

Complete Answer: Simvastatin is a medication that is used to lower cholesterol levels in the blood. Simvastatin is typically used to treat high cholesterol and to prevent heart disease, and to reduce the risk of heart attacks and strokes in those with the greatest risk of these happening (or re-occurring). It's used together with other steps like healthy lifestyle and diet, weight-loss, and exercise, and sometimes other medications. Simvastatin may also be used to decrease the amount of...

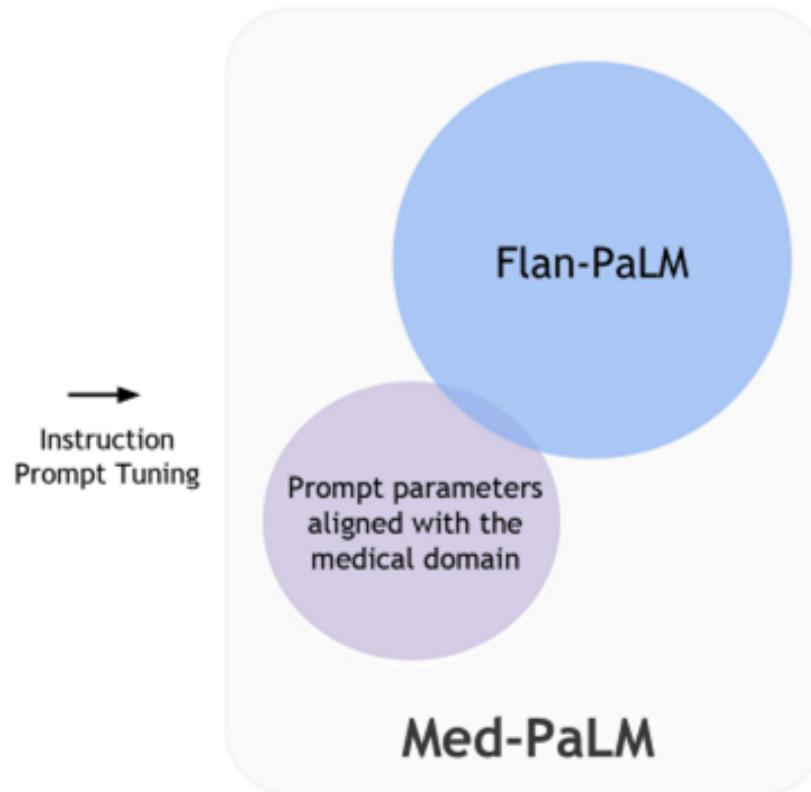
You are a helpful medical knowledge assistant. Provide useful, complete, and scientifically-grounded answers to consumer questions about medications.

Question: How can I split a 3mg or 4mg warfarin pill so I can get a 2.5mg pill?

Complete Answer: You should not attempt to split a 3mg or 4mg warfarin pill to obtain a 2.5mg dose. If you are prescribed warfarin in a 2.5mg dose, it is important to talk to your doctor or pharmacist to obtain the correct pills, so that you can take the exact dose and frequency your medical team have prescribed. Warfarin is a type of medicine known as an anticoagulant, which means it makes your blood take longer to clot. The exact amount required to achieve the desired effect can vary.

Extended Data Fig. 1 | Instruction prompt tuning for Med-PaLM. We use instructions and exemplars from a panel of qualified clinicians for each of the consumer medical question answering datasets and use them to instruction

[35] Karan Singhal et al. (2023), “[Large language models encode clinical knowledge](#)” より引用

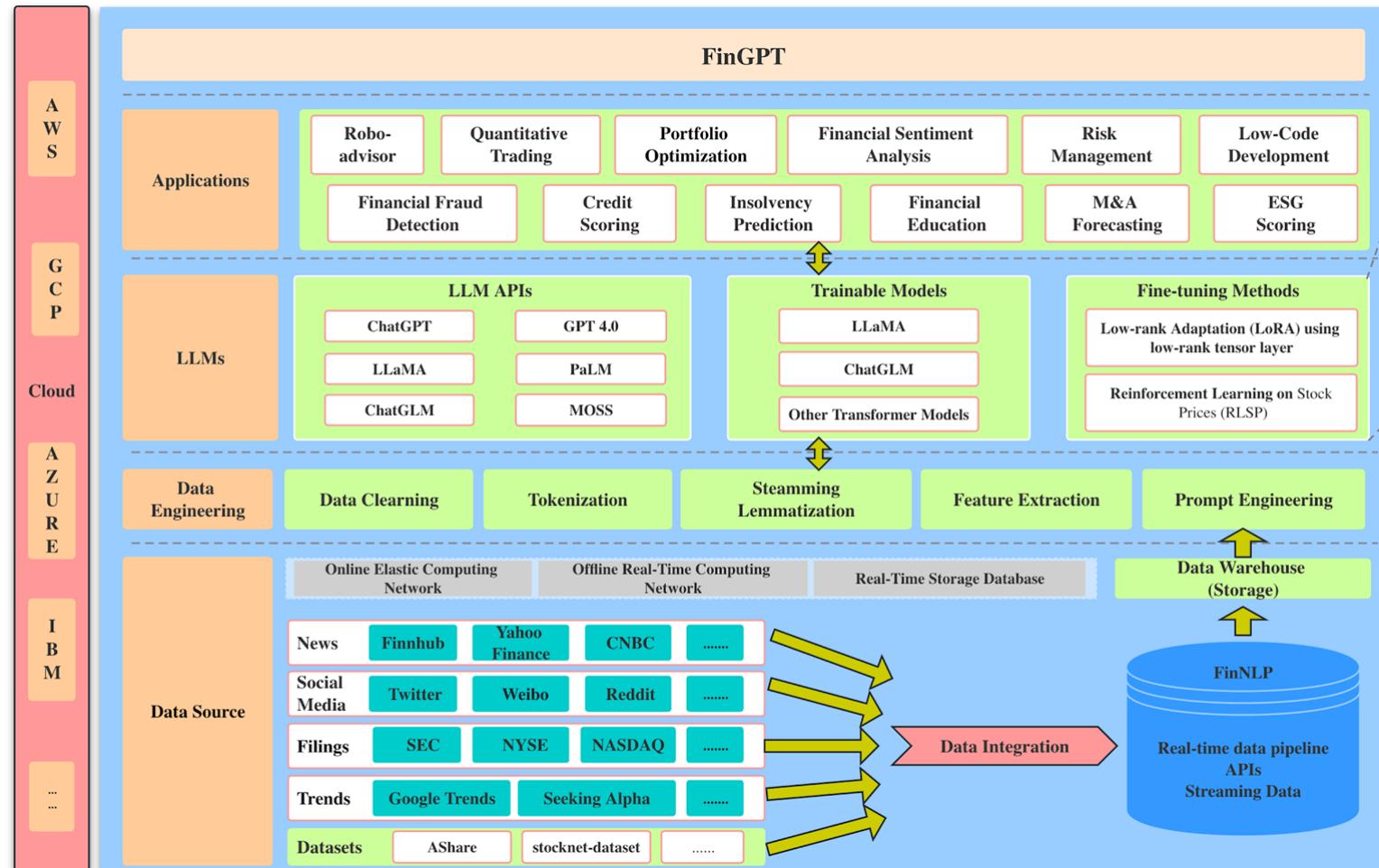


prompt tune Flan-PaLM. Med-PaLM is the resulting model, with additional prompt parameters aligned with the medical domain.

- **Med-PaLM**^[35] : Google が開発したLLM PaLM^[36]を医療向けに Fine-Tuning したモデル
- 医療質疑応答タスクでSOTA
- 複数の Fine-Tuning 手法を組み合わせた、 Instruction Prompt Tuning を適用

[36] Aakanksha Chowdhery et al. (2022), “[PaLM: Scaling Language Modeling with Pathways](#)” を参考

大規模言語モデル Fine-Tuning 事例 | FinGPT



Fine-tuning Methods

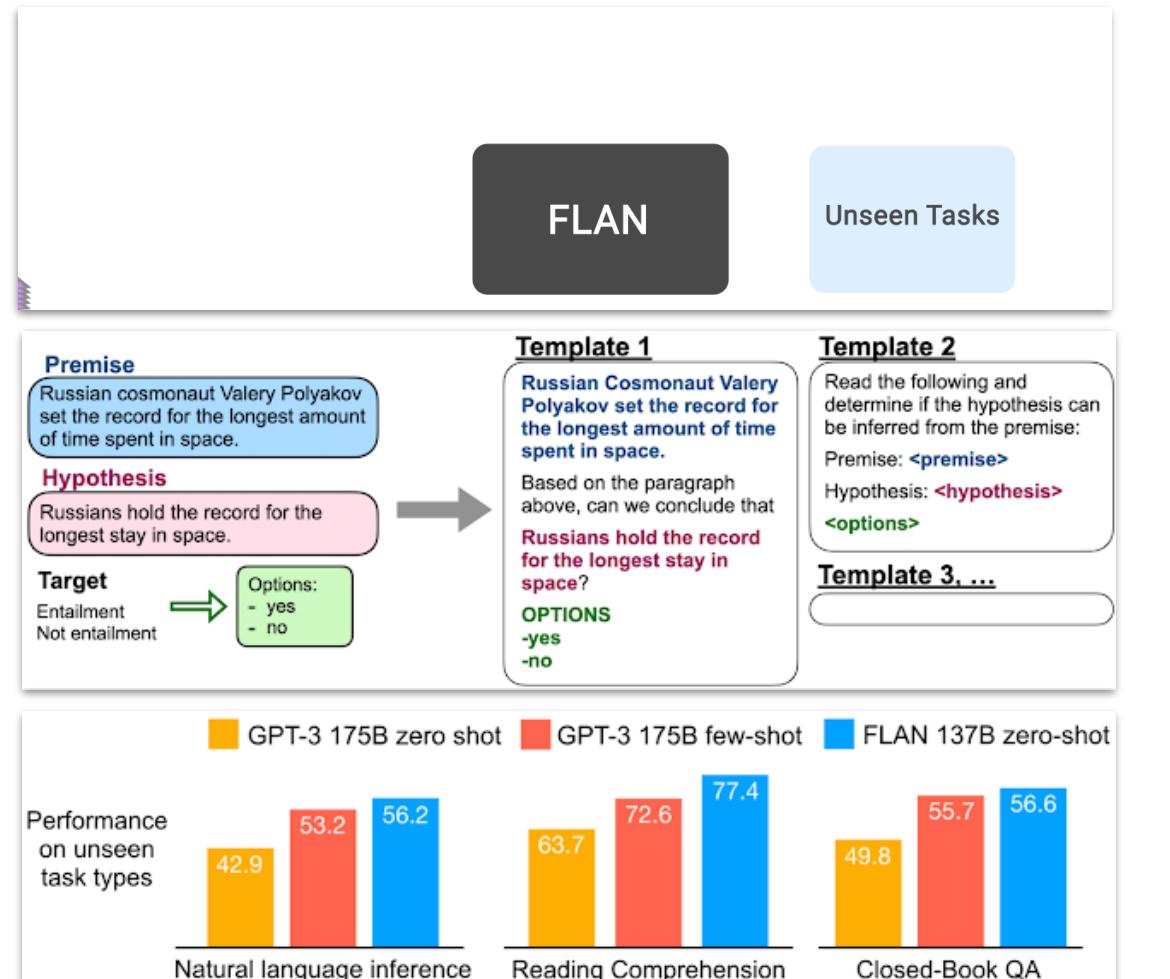
Low-rank Adaptation (LoRA) using low-rank tensor layer

Reinforcement Learning on Stock Prices (RLSP)

- **FinGPT[37] :**
 - 金融分野に特化したLLMを開発するためのオープンソース
 - ・フレームワーク
- 事前学習済みLLM をFine-Tuning する手法を推進

[37] Hongyang Yang et al. (2023), “[FinGPT: Open-Source Financial Large Language Models](#)” より引用

Instruction Tuning 概要 | FLAN論文による提案

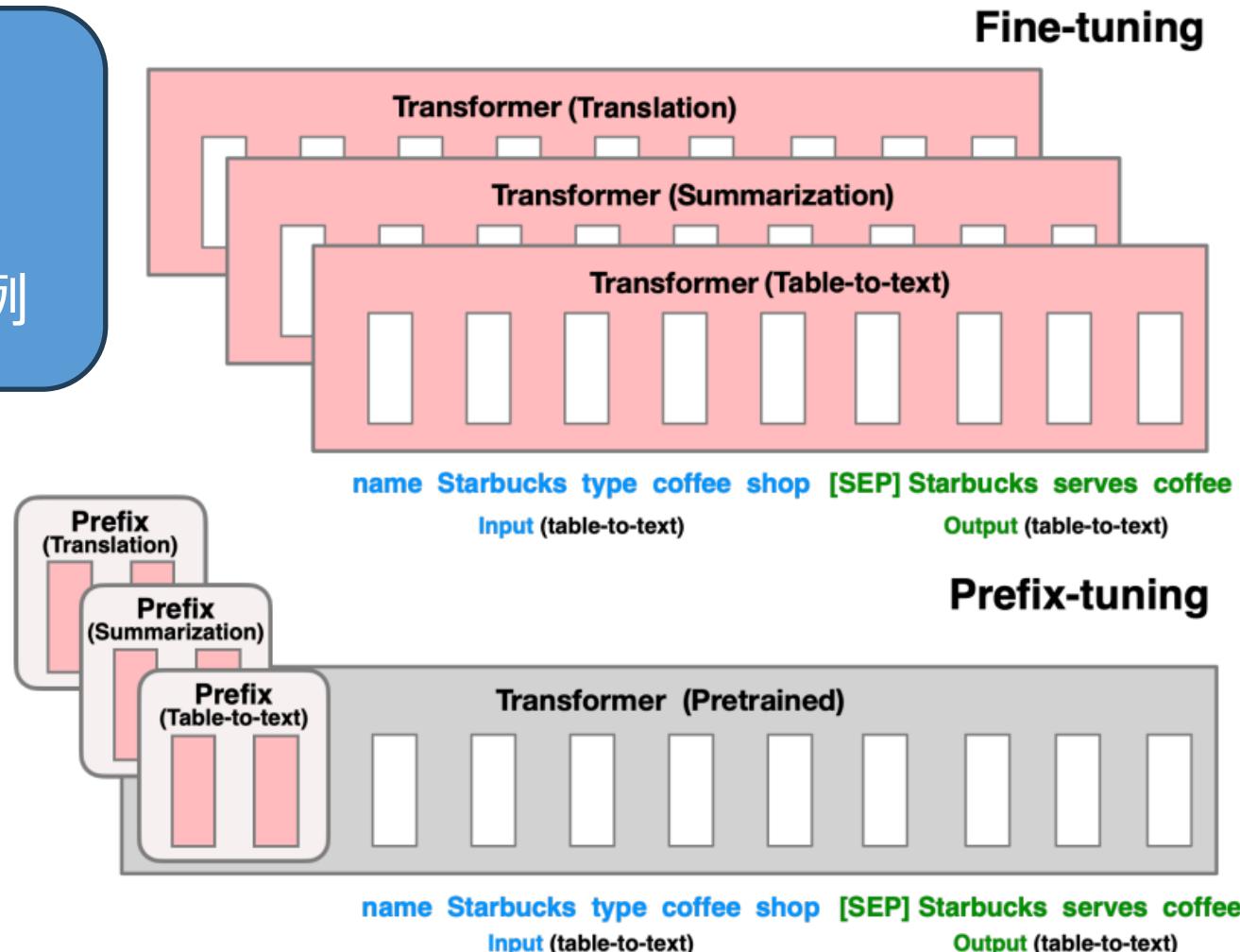


- Wei, Jason, et al. "**Finetuned language models are zero-shot learners.**" arXiv preprint arXiv:2109.01652 (2021).
- 様々なタスクを指示・回答という形式に統一したデータセットで、言語モデルを Fine-Tuning する手法を提案 (**Instruction Tuning**)
- このように Fine-Tuning されたモデルは、評価に用いられた25のタスクについて:
 - 21タスクで、Zero-shot性能が向上
 - 20タスクで、よりパラメータ数の多い GPT-3と比べて、高いZero-shot性能

[38] Google Research “[Introducing FLAN: More generalizable Language Models with Instruction Fine-Tuning](#)”より引用

■効率的なファインチューニング事例：Prefix-Tuning

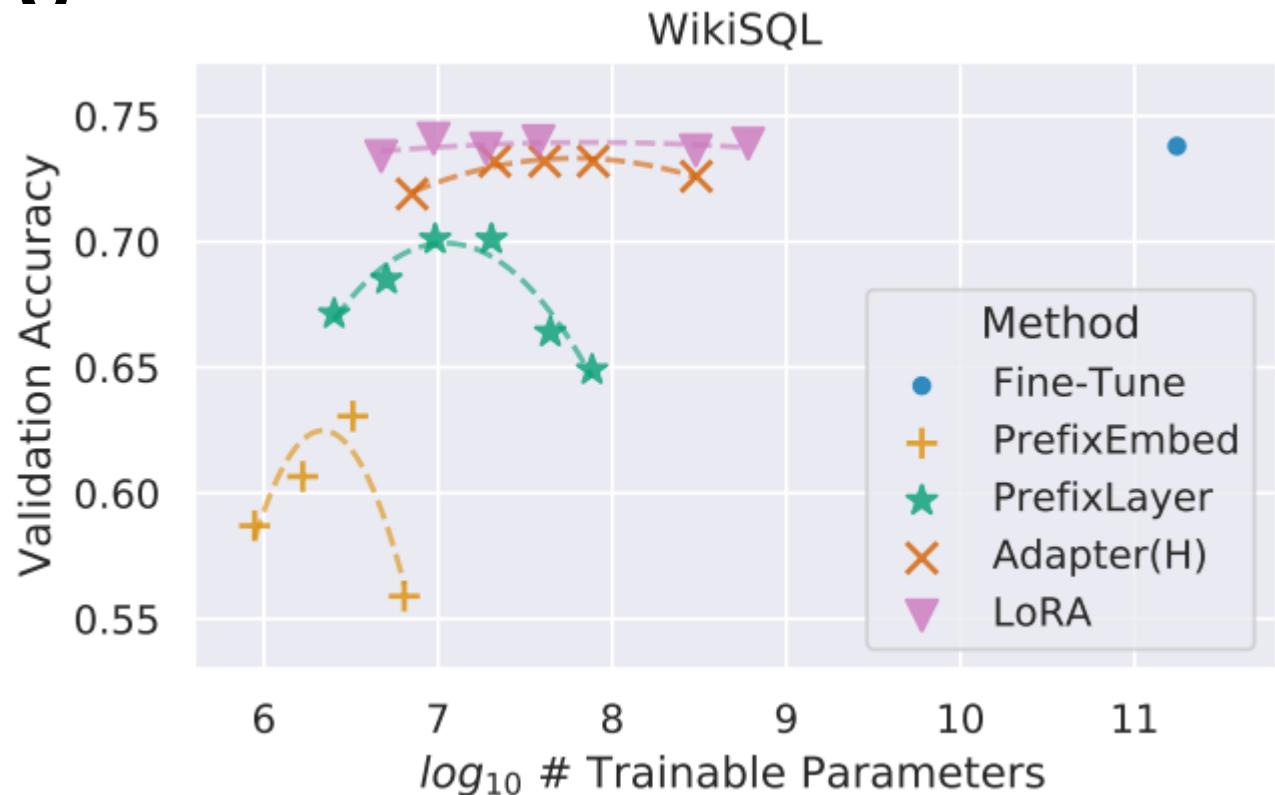
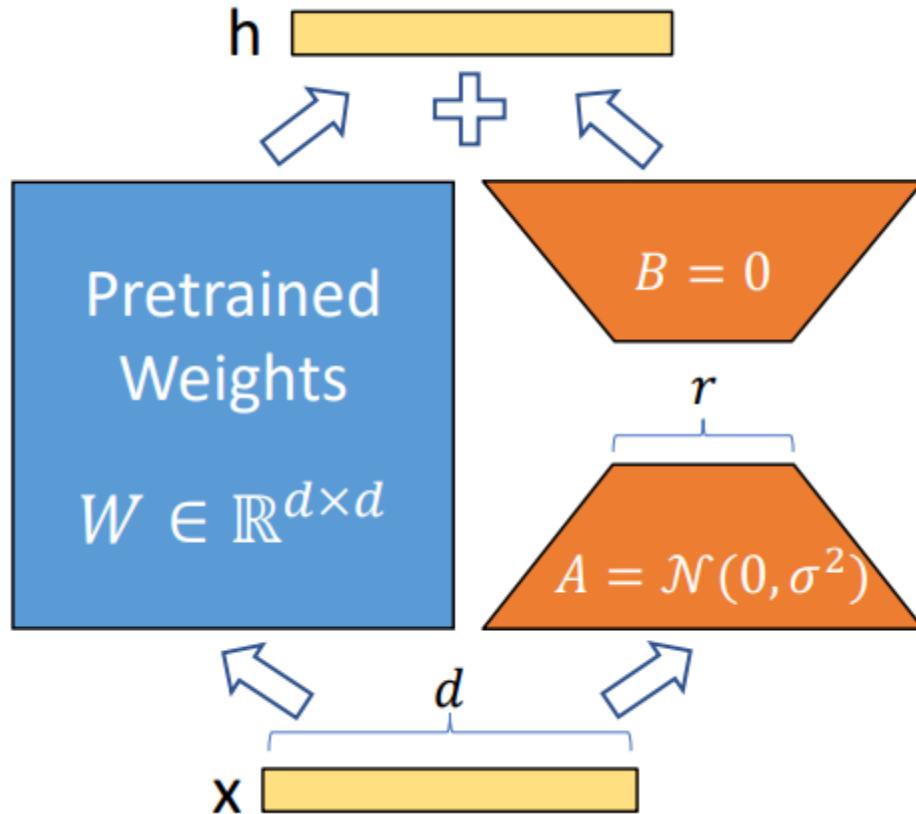
Parameter
Efficient
Fine-Tuning
(PEFT) の事例



Prefixとしてタスクごとに学習可能な埋め込みを挿入

[39] Xiang Lisa Li & Percy Liang, 2021 “[Prefix-Tuning: Optimizing Continuous Prompts for Generation](#)”, ACL2021より引用

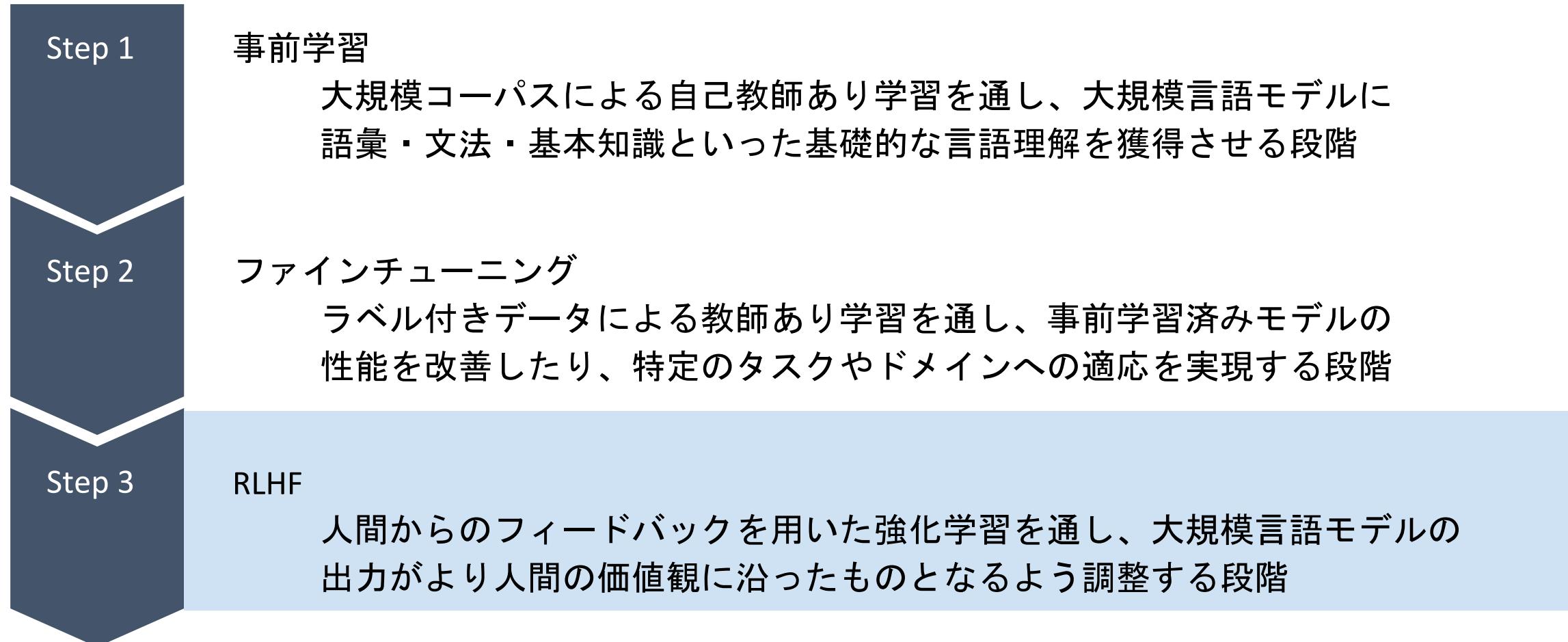
■効率的なファインチューニング事例： Low Rank Adaptation (LoRA)



[40] Edward J. Hu et al. (2021), “[LoRA: Low-Rank Adaptation of Large Language Models](#)” より引用

事前学習されたパラメータのパスとは別に
重みを低ランク近似した計算パスを用意し、足し合わせる。
安定してチューニングできる。

LLM学習フロー





- 意図には明示的な意図と暗黙的な意図が存在する

- 明示的な意図: 言語化して伝えている意図
 - Ex. この指示に従ってください, アシスタントとして振る舞ってください
- 暗黙的な意図: 言語化はしていないが, 対話において当たり前とされている意図
 - Ex. 捏造しない, 有害なことは言わない

Explicit intent

- Follow instructions
- Be an assistant

Implicit intent

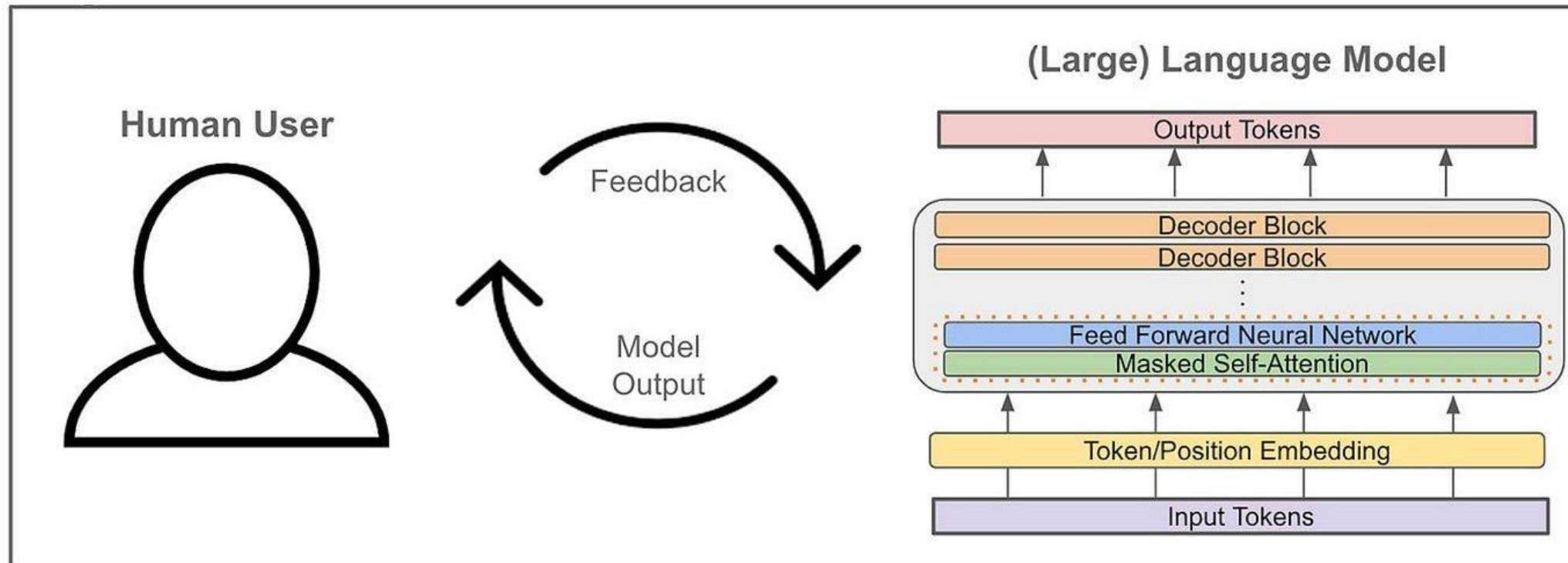
- Do what I mean
- Don't make stuff up
- Don't be mean
- Ask follow-up questions
- Refuse harmful tasks
- Avoid stereotyping
- ...

[45] [Cs25 Stanford Seminar - Transformers United 2023, Language and Human Alignment](#) より引用

基本的なAlignmentのアプローチ



- 人間が言語モデルの出力に対してフィードバックを行い、人間の意図通りに調整していく
- HITL(Human in the loop)型のアプローチ

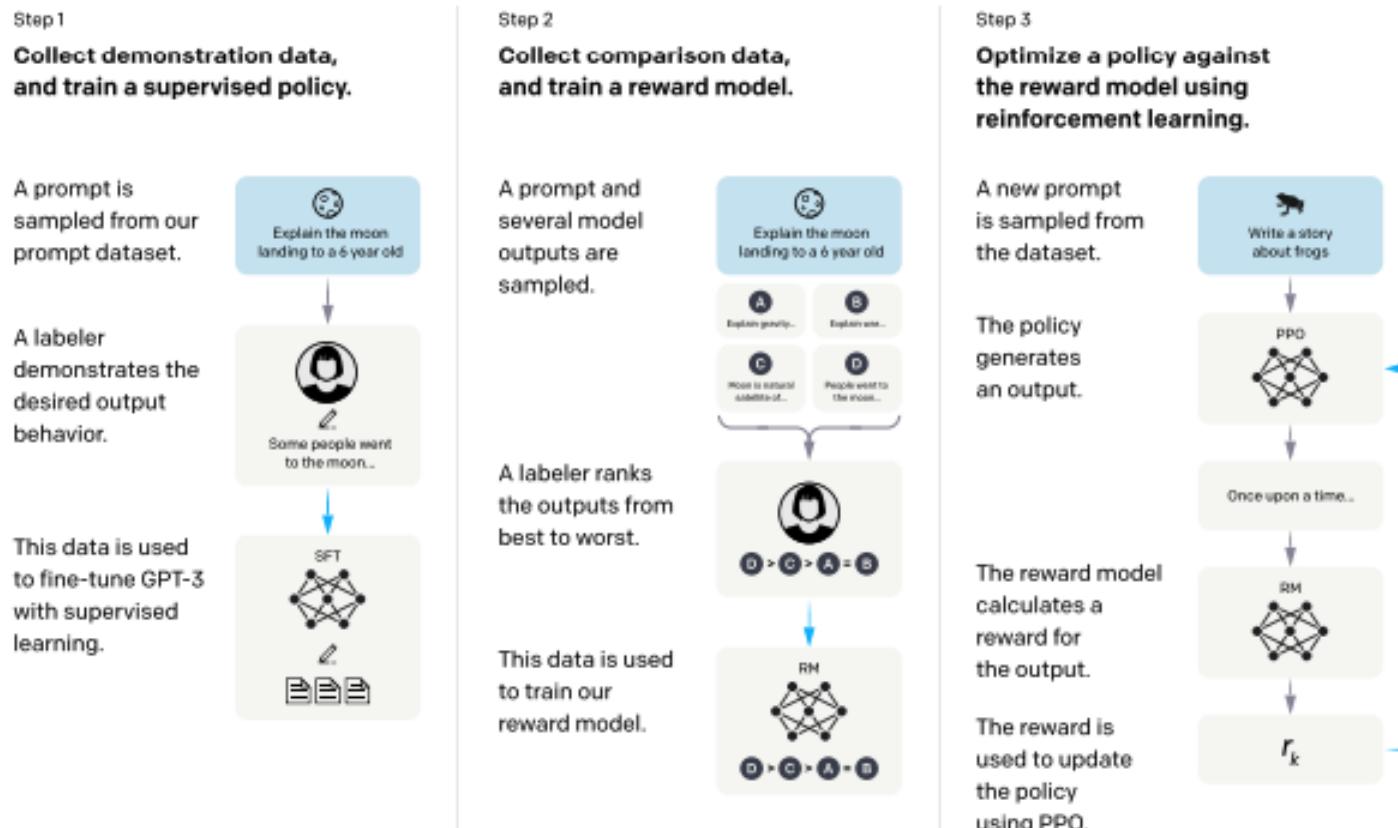


[46] CAMERON R. WOLFE, PH.D (2023), [Specialized LLMs: ChatGPT, LaMDA, Galactica, Codex, Sparrow, and More](#) より引用

Training Language Models to Follow Instructions with Human Feedback (2022)



- ChatGPTの前身であるInstructGPTで用いられている手法
- 既存のGPT-3をアライメントすることが目的
- 一般にRLHFと言うとこの手法を指すことが多い



途中から人間のフィードバックを自動化（モデル化）することで効率的な改善を実現している。

[41] Long Ouyang et al. (2022), “[Training Language Models to Follow Instructions with Human Feedback](#)” NeurIPS2022 より引用



- Helpful (有用かどうか)

- ユーザーの質問にたいして、できるだけ簡潔で効率的な回答を行う
- 不足情報がある場合、適切な質問を投げかけて情報を引き出す
- 相手のレベルに合わせた質問応答を行う

- Honest (誠実かどうか)

- 情報の虚偽がなく、正確な文章を出力する
- モデル自身がどの程度の不確実性のある情報かを提示することが重要
- (モデル自身がモデルの知っていることを理解している必要がある)

- Harmless (無害かどうか)

- 攻撃的、差別的な発言をしない
- 悪意のある質問を検知し、拒否をする

* 他にも、(Taxonomy, behavior, incentive, inner aspectsなど)

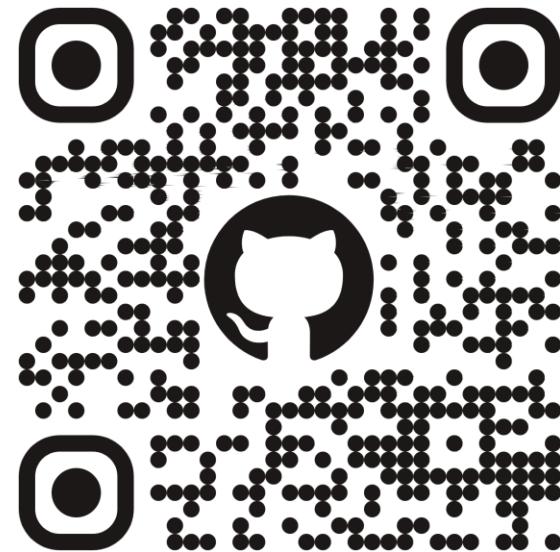
この3つを合わせてalignedされたAIと定義している論文もある(HHH)

目次

- LLMの概要
- 様々なテクニック
 - プロンプティング
 - フайнチューニング
- 実践的なLLMのチュートリアル

実践的なLLMのチュートリアル

- 商用LLMのAPIを使う
- ローカルでLLMを推論する
- ローカルでLLMを追加学習する



チュートリアルに使用するコード：
<https://github.com/resnant/llm4mat-tutorial>
exampleはnotebook/以下

LLMの提供形態は2つに大別

1. 商用LLM

- e.g. GPT-4, Gemini, Claude
 - WebシステムやAPIとして利用
-
- **Pros**
 - 性能は良い
 - 簡単に使える
 - **Cons**
 - 使っただけお金がかかる
 - (大量に使うと数十万円~)
 - モデルはブラックボックス
 - 再現性の確保は不可能
 - 研究で使う場合は注意

2. オープンなLLM

- e.g. Llama3, Mistral
 - モデル（DNN本体）が配布されており、それを手元の計算機で動かす
-
- **Pros**
 - データを完全に管理下における
 - モデルの改造や追加学習も自由
 - いくら使ってもコストは計算機代のみ
 - **Cons**
 - 性能は玉石混交
 - 高価な計算機が必要

まずは商用LLMを使って、
やりたいことができるか試してみると良い

商用LLMのAPIを使う例（Gemini 1.5 Pro）

例：論文のテキストから、論文を説明するキーワードを生成する

- 入力：論文のタイトルと本文
- 出力：キーワードのリスト
- プロンプトの例：

```
Instruction: 以下の論文を説明するキーワードを最大10個提案してください  
Title: {論文タイトル}  
Main text: {論文本文}
```

- Note:
 - 最近の商用LLMは非常に長い文章の入出力が可能
 - Gemini 1.5 Proは200M tokens（論文100本入れてまとめさせることも余裕）

ローカルLLMをファインチューニングする例

結晶構造から物性値を予測するタスクを解いてみる

- 入力：結晶構造のテキスト表現
- 出力：バンドギャップ（数値）

- 学習コード：
`train_structure2property.py`

- このフォーマットに成形した
入出力ペア11万件を用いて
LLMを学習した
 - モデル：[mistralai/Mistral-Nemo-Base-2407](#)
 - データ：[Materials Project](#)

```
Instruction: What is the bandgap value of following material?:  
Reduced Formula: TlCd(NO2)3  
abc : 5.494943 5.494943 5.494943  
angles: 90.612789 90.612789 90.612788  
pbc : True True True  
space group: ('R3', 146)  
Sites (11)  
# SP a b c magmom  
0 Tl 0.384554 0.384554 0.384554 0  
1 Cd 0.898306 0.898306 0.898306 -0  
2 N 0.442446 0.886628 0.838588 0  
3 N 0.886628 0.838588 0.442446 0  
4 N 0.838588 0.442446 0.886628 0  
5 O 0.688162 0.837134 0.325966 0  
6 O 0.325966 0.688162 0.837134 0  
7 O 0.849424 0.315744 0.077715 -0  
8 O 0.077715 0.849424 0.315744 -0  
9 O 0.315744 0.077715 0.849424 -0  
10 O 0.837134 0.325966 0.688162 0  
  
Output:  
2.1399
```

ローカルLLMをファインチューニングする例

結晶構造から物性値を予測するタスクを解く

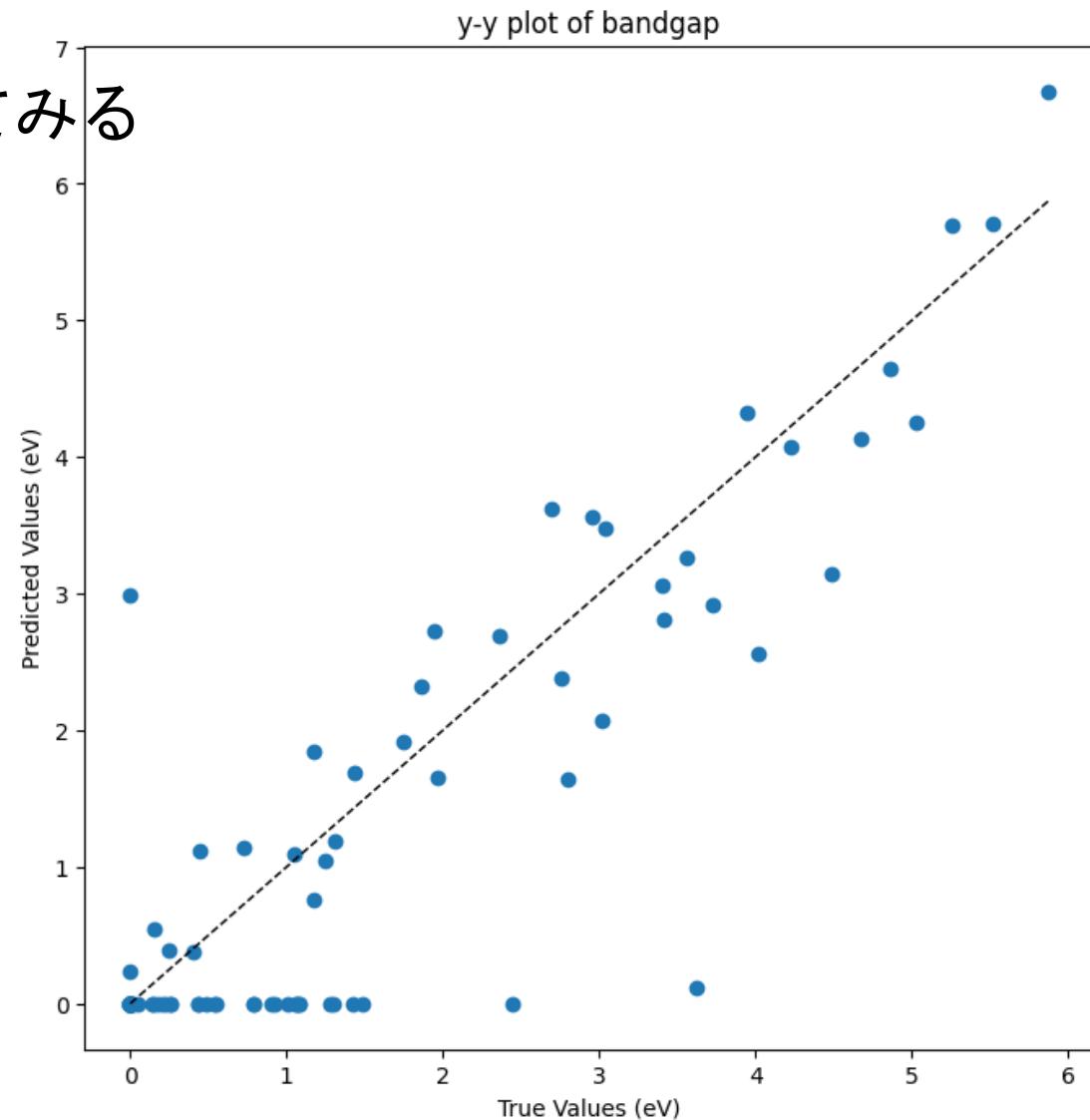
- 入力：結晶構造のテキスト表現
- 出力：バンドギャップ（数値）
- 学習コード：
`train_structure2property.py`
- このフォーマットに成形した
入出力ペア11万件を用いて
LLMを学習した
 - モデル：mistralai/Mistral-Nemo-Base-2407
 - データ：Materials Project

```
1 # dataset loading
2 mp_data = load_mp_pickles("./mp_download")
3 dataset_split = prepare_datasets(mp_data, "band_gap")
4
5 # model setup
6 model_id = "mistralai/Mistral-Nemo-Base-2407"
7 tokenizer, model = setup_tokenizer_and_model(model_id = model_id)
8
9 # define training parameters
10 < training_args = transformers.TrainingArguments(
11     num_train_epochs=1,
12     learning_rate=1.0e-5,
13     per_device_train_batch_size=1,
14     bf16=True
15 )
16 # trainer instance setup
17 < trainer = SFTTrainer(
18     args=training_args,
19     model=model,
20     tokenizer=tokenizer,
21     train_dataset=dataset_split["train"],
22     max_seq_length=2048,
23     formatting_func=formatting_prompts_func
24 )
25
26 trainer.train()
```

ローカルLLMをファインチューニングする例

結晶構造から物性値を予測するタスクを解いてみる

- 入力：結晶構造のテキスト表現
- 出力：バンドギャップ（数値）
- 学習コード：
`train_structure2property.py`
- $R^2 = 0.79$, $MAE = 0.36 \text{ eV}$ (on test set)
 - さほど悪くはないが、良くもない...
 - 最新のDNNなら $MAE = 0.2 \text{ eV}$ ぐらいの精度
 - c.f. <https://omron-sinicx.github.io/crystalformer/>



- [1] Ashish Vaswani et al. (2017) "[Attention Is All You Need](#)" NeurIPS 2017
- [2] Alec Radford et al. (2018) "[Improving Language Understanding by Generative Pre-training](#)"
- [3] Momentum Works 2023 "The future by ChatGPT" <https://momentum.asia/product/the-future-by-chatgpt/> アクセス日:2023/11/19
- [4] Wayne Xin Zhao et al. (2023), "[A Survey of Large Language Models](#)" arXiv:2303.18223
- [5] Jared Kaplan et al. (2020), "[Scaling Laws for Neural Language Models](#)", arXiv:2001.08361
- [6] Jason Wei et al. (2022), "[Emergent Abilities of Large Language Models](#)" arXiv:2206.07682
- [7] Tom Brown et al. (2020), "[Language Models are Few-Shot Learners](#)", NeurIPS2020
- [8] NVIDIA AI対応GPU搭載サーバー | NVIDIA GPU Solution | SCSK株式会社, <https://www.scsk.jp/sp/nvidia/ai-server/index.html>, アクセス日:2023/12/1
- [9] 総産研 ABCI <https://abci.ai/ja/> アクセス日:2023/11/19
- [10] アマゾン ウェブ サービス (AWS クラウド) - ホーム, <https://aws.amazon.com/jp/>, アクセス日:2023/12/1
- [11] クラスマソッド Google Cloud Advent Calendar 2021 の記事一覧 | DevelopersIO, <https://dev.classmethod.jp/referencecat/classmethod-google-cloud-advent-calendar-2021/>, アクセス日:2023/12/1
- [12] Microsoft Azure Logo and symbol, meaning, history, PNG, brand, <https://1000logos.net/microsoft-azure-logo/>, アクセス日:2023/12/1
- [13] Pengfei Liu et al. (2021), "[Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing](#)", arXiv:2107.13586
- [14] Rishi Bommasani et al. (2021) "[On the Opportunities and Risks of Foundation Models](#)", arXiv:2108.07258
- [15] OpenAI 2023 "[GPT-4 Technical Report](#)"
- [16] Jungo Kasai et al. (2023), "[Evaluating gpt-4 and ChatGPTt on Japanese medical licensing examinations](#)" arXiv:2303.18027
- [17] Michael Ahn et al. (2022), "[Do As I Can, Not As I Say: Grounding Language in Robotic Affordances](#)" arXiv:2204.01691
- [18] Guanzhi Wang et al. (2023), "[Voyager: An Open-Ended Embodied Agent with Large Language Models](#)" arXiv: 2305.16291
- [19] Lukasz Kaiser et al. (2017), "[One Model to Learn Them All](#)" arXiv:1706.05137
- [20] Anthony Brohan et al. (2022), "[RT-1: Robotics Transformer for Real-World Control at Scale](#)", arXiv:2212.06817

- [21] Jean-Baptiste Alayrac et al. (2022), “[Flamingo : a Visual Language Model for Few-Shot Learning](#)”, NeurIPS2022
- [22] Jean-Baptiste Alayrac et al.(2022) Tackling multiple tasks with a single visual language model - Google DeepMind
<https://deepmind.google/discover/blog/tackling-multiple-tasks-with-a-single-visual-language-model/> アクセス日: 2023/11/18
- [23] 人類の進化のイラスト | 商用可・フリーイラスト素材 | ソコスト, <https://soco-st.com/13472> アクセス日:2023/11/19
- [24] Jason Wei et al. (2022), “[Chain of Thought Prompting Elicits Reasoning in Large Language Models](#)” NeurIPS2022
- [25] Timo Schick et al. (2023), “[Toolformer: Language Models Can Teach Themselves to Use Tools](#)”, arXiv:2302.04761
- [26] Pan Lu et al. (2023), “[Chameleon: Plug-and-Play Compositional Reasoning with Large Language Models](#)”, arXiv:2304.09842
- [27] Patrick Lewis et al. (2020), “[Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks](#)”, NeurIPS2020
- [28] Raimi Karim (2019) Illustrated: Self-Attention. A step-by-step guide to self-attention… | by Raimi Karim | Towards Data Scienc
<https://towardsdatascience.com/illustrated-self-attention-2d627e33b20a> アクセス日:2023/11/19
- [29] Jay Alammar (2018) The Illustrated Transformer – Jay Alammar – Visualizing machine learning one concept at a time.
<https://jalammar.github.io/illustrated-transformer/> アクセス日:2023/11/19
- [30] Hugo Touvron et al. (2023), “[LLaMA: Open and Efficient Foundation Language Models](#)”, arXiv:2302.13971
- [31] Dzmitry Bahdanau (2022), The FLOPs Calculus of Language Model Training | by Dzmitry Bahdanau | Medium,
<https://medium.com/@dzmitrybahdanau/the-flops-calculus-of-language-model-training-3b19c1f025e4> アクセス日:2023/11/19
- [32] Manzil Zaheer et al. (2020), “[Big Bird: Transformers for Longer Sequences](#)”
- [33] Iz Beltagy et al. (2020), “[Longformer: The Long-Document Transformer](#)”, arXiv:2004.05150
- [34] William Fedus et al. (2022), “[Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity](#)”, Journal of Machine Learning Research 23 (2022) 1-39
- [35] Microsoft Deep Speed Team (2023), DeepSpeed: 深層学習の訓練と推論を劇的に 高速化するフレームワーク,
https://www.deepspeed.ai/assets/files/DeepSpeed_Overview_Japanese_2023Jun7th.pdf アクセス日: 2023/11/19
- [36] Guilherme Penedo et al. (2023), “[The RefinedWeb Dataset for Falcon LLM](#)”, arXiv: 2306.01116
- [37] Ben Sorscher et al. (2022), “[Beyond neural scaling laws:beating power law scaling via data pruning](#)”, NeurIPS2022

- [38] OpenAI “GPT-3.5 Turbo fine-tuning and API updates” <https://openai.com/blog/gpt-3-5-turbo-fine-tuning-and-api-updates> アクセス日:2023/11/19
- [39] Karan Singhal et al. (2023), “[Large language models encode clinical knowledge](#)” Nature vol620 page 172-180
- [40] Aakanksha Chowdhery et al. (2022), “[PaLM: Scaling Language Modeling with Pathways](#)” arXiv:2204.02311
- [41] Hongyang Yang et al. (2023), “[FinGPT: Open-Source Financial Large Language Models](#)” arXiv2306.06031
- [42] Google Research “Introducing FLAN: More generalizable Language Models with Instruction Fine-Tuning”, <https://blog.research.google/2021/10/introducing-flan-more-generalizable.html> アクセス日: 2023/11/19
- [43] Xiang Lisa Li & Percy Liang, 2021 “[Prefix-Tuning: Optimizing Continuous Prompts for Generation](#)”, Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 4582–4597
- [44] Edward J. Hu et al. (2021), “[LoRA: Low-Rank Adaptation of Large Language Models](#)” arXiv:2106.09685
- [45] Cs25 Stanford Seminar - Transformers United 2023,Language and Human Alignment, https://www.youtube.com/watch?v=DJ1Yy6Aquug&list=PLoROMvodv4rNjRchCzutFw5ItR_Z27CM&index=14, アクセス日2023/11/28
- [46] CAMERON R. WOLFE, PH.D, Specialized(2023), LLMs: ChatGPT, LaMDA, Galactica, Codex, Sparrow, and More, <https://cameronrwolfe.substack.com/p/specialized-llms-chatgpt-lamda-galactica>, アクセス日2023/11/28
- [47] Long Ouyang et al. (2022), “[Training Language Models to Follow Instructions with Human Feedback](#)” NeurIPS2022
- [48] Shibani Santurkar et al. (2023), “[Whose Opinions Do Language Models Reflect?](#)” arXiv:2303.17548
- [49] Cyber Agent (2023), サイバーエージェント、最大68億パラメータの日本語LLM（大規模言語モデル）を一般公開 —オープンなデータで学習した商用利用可能なモデルを提供— | 株式会社サイバーエージェント <https://www.cyberagent.co.jp/news/detail/id=28817> アクセス日:2023/11/19
- [50] rinna (2023), rinna、日本語に特化した36億パラメータのGPT言語モデルを公開 | rinna株式会社 <https://rinna.co.jp/news/2023/05/20230507.html> アクセス日:2023/11/19
- [51] stability.ai (2023), 日本語言語モデル「Japanese StableLM Alpha」をリリースしました — Stability AI Japan, <https://ja.stability.ai/blog/japanese-stablelm-alpha> アクセス日:2023/11/19
- [52] Ledge.ai編集部 (2023), LINE 36億パラメータの日本語LLMを公開 商用利用も可 | Ledge.ai, https://ledge.ai/articles/line_japanese_large_lm アクセス日:2023/11/19
- [53] 日本電気株式会社 (2023), NEC、130億パラメータで世界トップクラスの日本語性能を有する軽量なLLMを開発 (2023年7月6日): プレスリリース | NEC, https://jpn.nec.com/press/202307/20230706_02.html アクセス日:2023/11/19

- [54] OpenAI (2023), GPT-4, <https://openai.com/research/gpt-4> アクセス日:2023/11/19
- [55] ELYZA (2023), 70億パラメータの商用利用可能な日本語LLM「ELYZA-japanese-Llama-2-7b」を一般公開しました | ELYZA, <https://elyza.ai/news/2023/08/29/70%E5%84%84%E3%83%91%E3%83%A9%E3%83%A1%E3%83%BC%E3%82%BF%E3%81%AE%E5%95%86%E7%94%A8%E5%88%A9%E7%94%A8%E5%8F%AF%E8%83%BD%E3%81%AA%E6%97%A5%E6%9C%AC%E8%AA%9Elmelyza-ja> アクセス日:2023/11/19
- [56] Linting Xue et al. (2021), “[mT5: A massively multilingual pre-trained text-to-text transformer](#)” Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 483–498
- [57] 櫻井 章雄 (2022), 世界で開発が進む大規模言語モデルとは（後編） | NTTデータ先端技術株式会社, <https://www.intellilink.co.jp/column/ai/2022/072800.aspx> アクセス日:2023/11/19
- [58] 経済産業省 (2023), クラウドプログラム (METI/経済産業省) , https://www.meti.go.jp/policy/economy/economic_security/cloud/index.html アクセス日:2023/11/19
- [59] polm-stability (2023), GitHub - Stability-AI/lm-evaluation-harness: A framework for few-shot evaluation of autoregressive language models., <https://github.com/Stability-AI/lm-evaluation-harness> アクセス日:2023/11/19
- [60] Aarohi Srivastava et al. (2022), “[Beyond the Imitation Game: Quantifying and extrapolating the capabilities of language models](#)” arXiv:2206.04615
- [61] Shayne Longpre et al. (2023), “[The Flan Collection: Designing Data and Methods for Effective Instruction Tuning](#)” arXiv:2301.13688
- [62]. Yizhong Wang et al. (2022), “[Super-NaturalInstructions: Generalization via Declarative Instructions on 1600 et al. NLP Tasks](#)” arXiv:2204.07705
- [63] Dan Hendryck et al. (2020), “[Measuring Massive Multitask Language Understanding](#)” arXiv:2009.03300
- [64] cerebras (2023) Cerebras-GPT: A Family of Open, Compute-efficient, Large Language Models – Cerebras, <https://www.cerebras.net/blog/cerebras-gpt-a-family-of-open-compute-efficient-large-language-models/> アクセス日:2023/11/19