注意事項: 本資料の再利用(2次利用)について

- 本資料は東京大学 松尾研究室が作成し、東京大学サマースクール2023として2023年9月から11月にかけて開催されたLLM大規模言語モデル講座の講義資料となっております。
- 本資料はクリエイティブ・コモンズのCC BY-NC-SA 4.0 DEED(表示 – 非営利 – 継承 4.0 国際)のライセンスが登録されています。
- ライセンスの表示について
  - 各スライドのページ最下部にライセンスの記載がございます。再利用時にはこちらの要素も含めてご利用ください。ただしこちらはスライドマスターに設定されている為、再利用時に複製が困難な場合は、下記のテキストボックスを利用の上、ハイパーリンクも含めてライセンスの表記をする様にお願いします。

    LLM 大規模言語モデル講座 講義資料 © 2023 by 東京大学松尾研究室 is licensed under CC BY-NC-ND 4.0

  - 再利用するページに参照論文等の引用がある場合は、巻末にあるReferenceより引用箇所を掲載してください。(引用元の著作権者に対しての再利用の正当性が証明できなくなる可能性がございます。)
- 非営利目的での利用に限り、再利用(2次利用)が許諾されております。
- 営利目的での再利用の場合はこちらからお問い合わせください。
- 元の表現が変わらない範囲(フォント、サイズ等)であれば改変可能です。
- それ以外の改変や、その他ライセンスについての詳細は、こちらをご覧の上、適切な取り扱いをして頂くようお願いします。

東京大学 松尾研究室

Overview of Large Language Models

松尾研究室
MATSUO LAB THE UNIVERSITY OF TOKYO

サマースクール 2023
大規模言語モデル講座

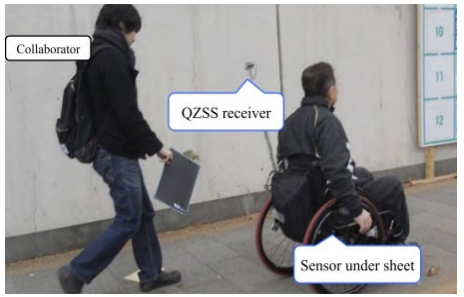
2023/09/04

岩澤 有祐（いわさわ ゆうすけ）

2017年東京大学工学系研究化博士課程修了（松尾研）．卒業後特任研究員，特任助教などを経て2022年より技術経営戦略学専攻で講師.

■ 研究テーマ

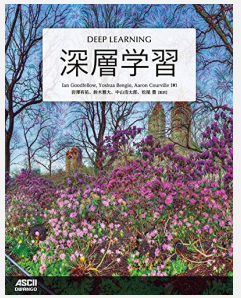
- 修士までは障害者支援への機械学習技術の応用
- 博士から深層学習の主に転移学習技術に関する研究
  “Large-Language Models are Zero-Shot Reasoners”, NeurIPS2022
  など



■ 教育関連

- DL基礎講座 / 深層学習，世界モデルと知能，など
- 今回の大規模言語モデル講座は全体の設計 / Day1 & Day2の講師などの予定

DL輪読会
松尾研メンバ，講義受講生などが参加する勉強会を主催.
2015年 ~ 累計350回以上実施（毎週金曜朝10:00）



DL本（監訳，翻訳）
Goodfellowらが執筆した深層学習の教科書の監訳，翻訳．2018年に出版.

目次

- **LLMの概要（LLMをなぜ学ぶのか？）**
- 各回の概要
- 日本のLLMを取り巻く環境

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$x_1, x_2, \cdots, x_L$に，その生成確率$p(x_1, x_2, \cdots, x_L)$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)

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

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

- 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

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

$p($パリ$|$日本, の, 首都, は$) = 0.001$

$\vdots$

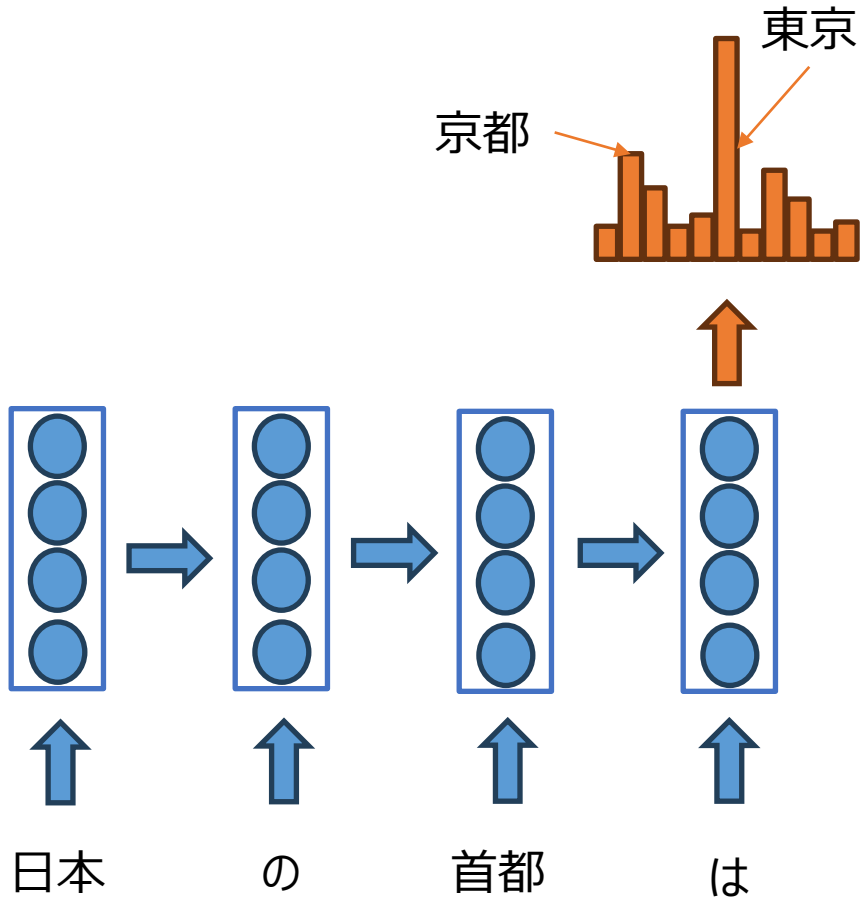
$p($カイロ$|$日本, の, 首都, は$) = 0.0005$

日本の首都は → **東京**

- この条件付き確率をどう求めるか？

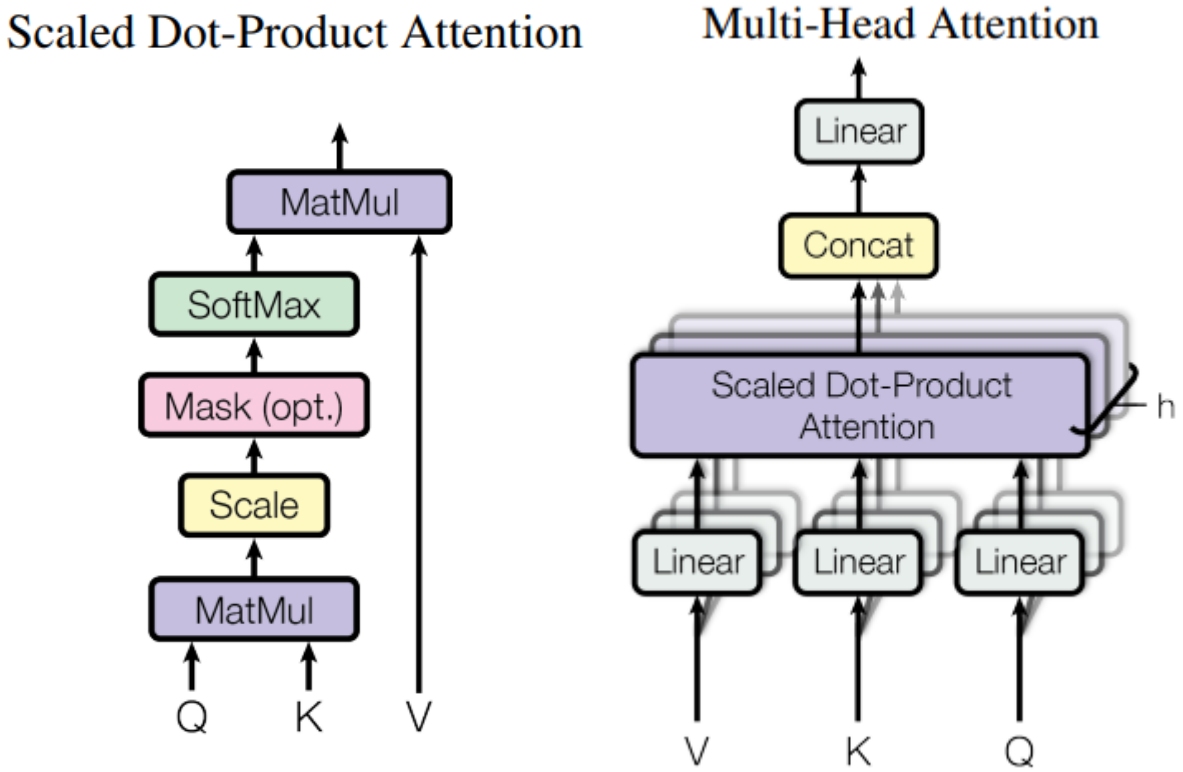
ニューラル言語モデル

- 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）



“Attention is All You Need”, NeurIPS 2017

Transformer



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

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**	$2.3 \cdot 10^{19}$	

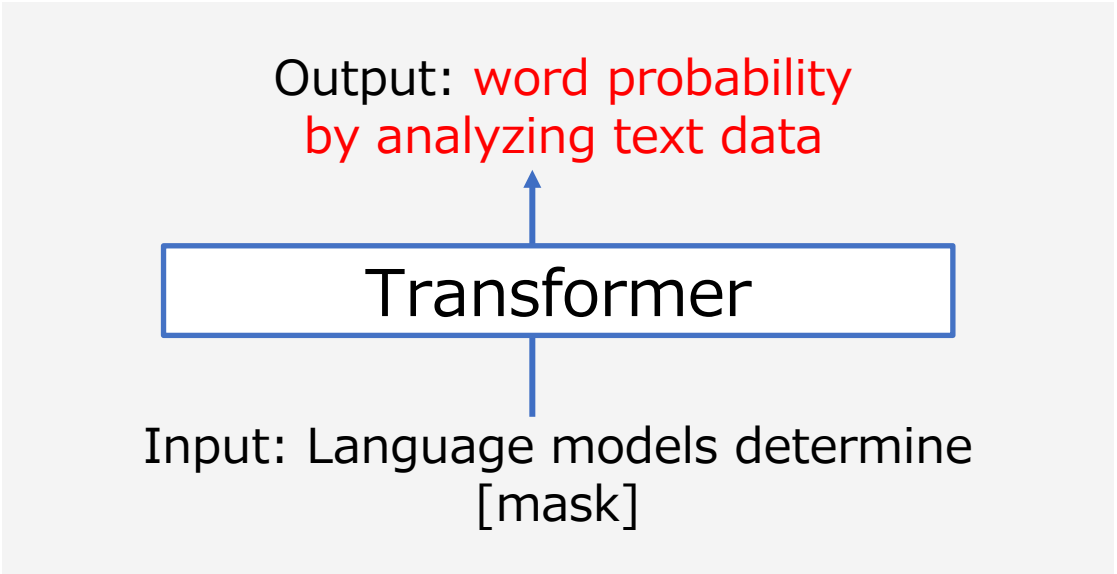
- Googleを中心にした研究チームが2017年に発表
- Self Attentionを中心にしたネットワーク構造（左）
  ※構造の詳細はDay3で話します
- 主に翻訳等の教師あり学習で性能検証（右）
  例：英語文 → Transformer → ドイツ語文
  となるように誤差逆伝播で訓練

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

“Improving Language Understanding by Generative Pre-training”, 2018

Generative Pretraining Transformer (GPT)

Pre-training （事前学習）

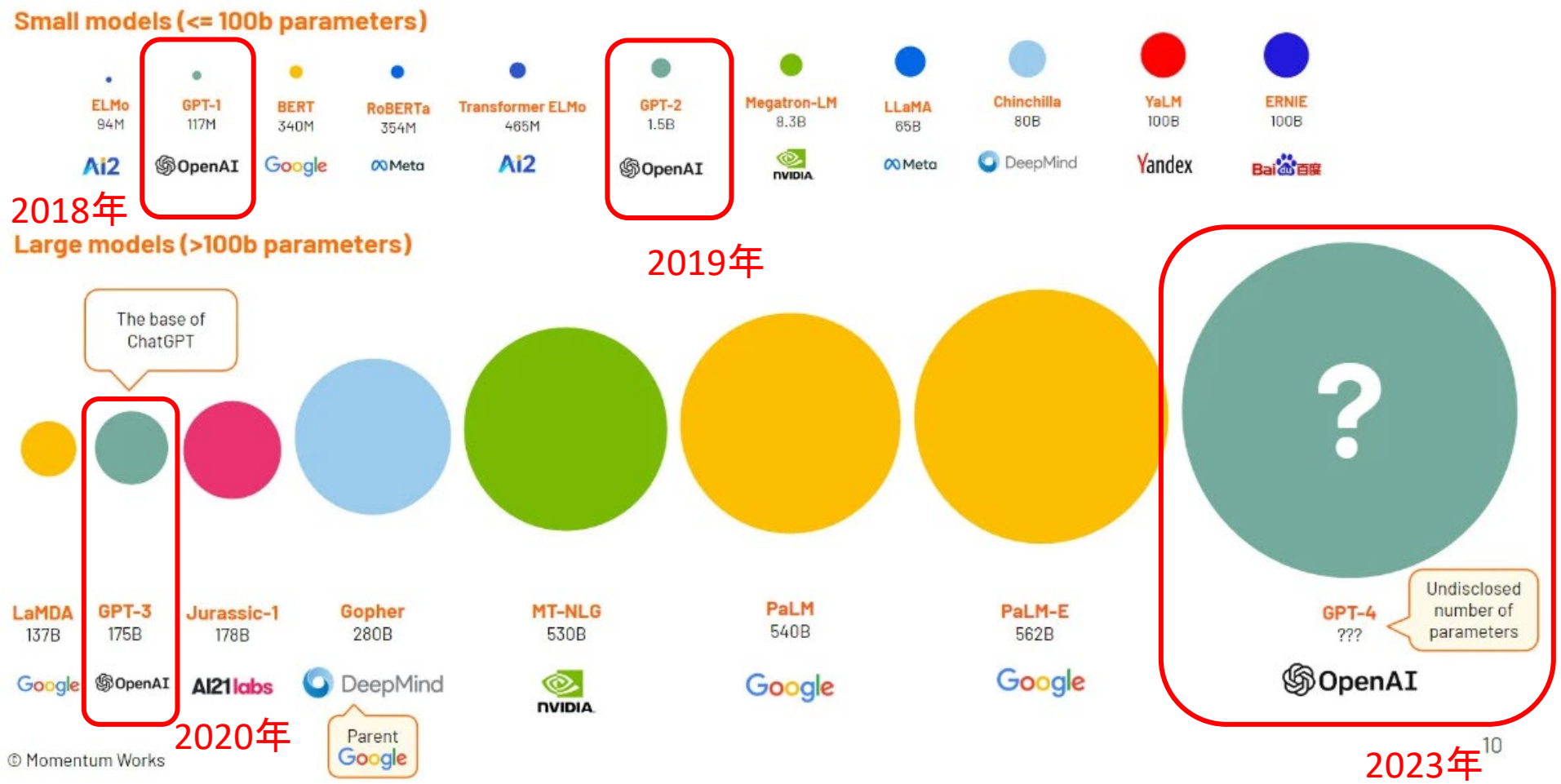


Original: Language models determine word probability by analyzing text data

- 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

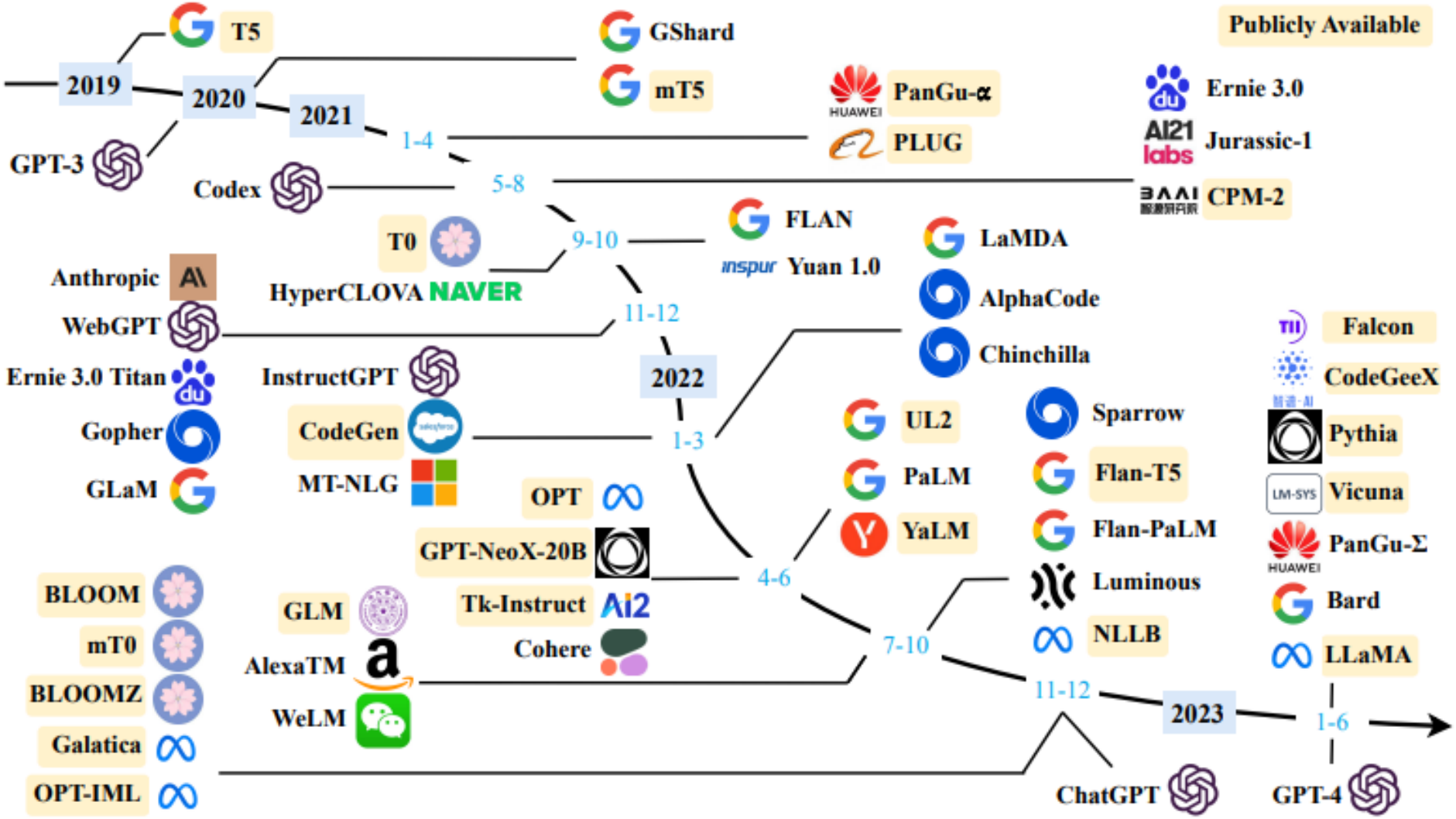


基本的にはいずれも2017年に発明されたTransformerと呼ばれる構造を利用.
GPT-3登場以降，米国企業を中心に複数の研究機関が独自の大規模言語モデルを開発.

[3] Momentum Works 2023 “The future by ChatGPT” より引用し,一部改変

“A Survey of Large Language Models”, 2023年9月にアクセス（3月投稿からすでに11回Revision）

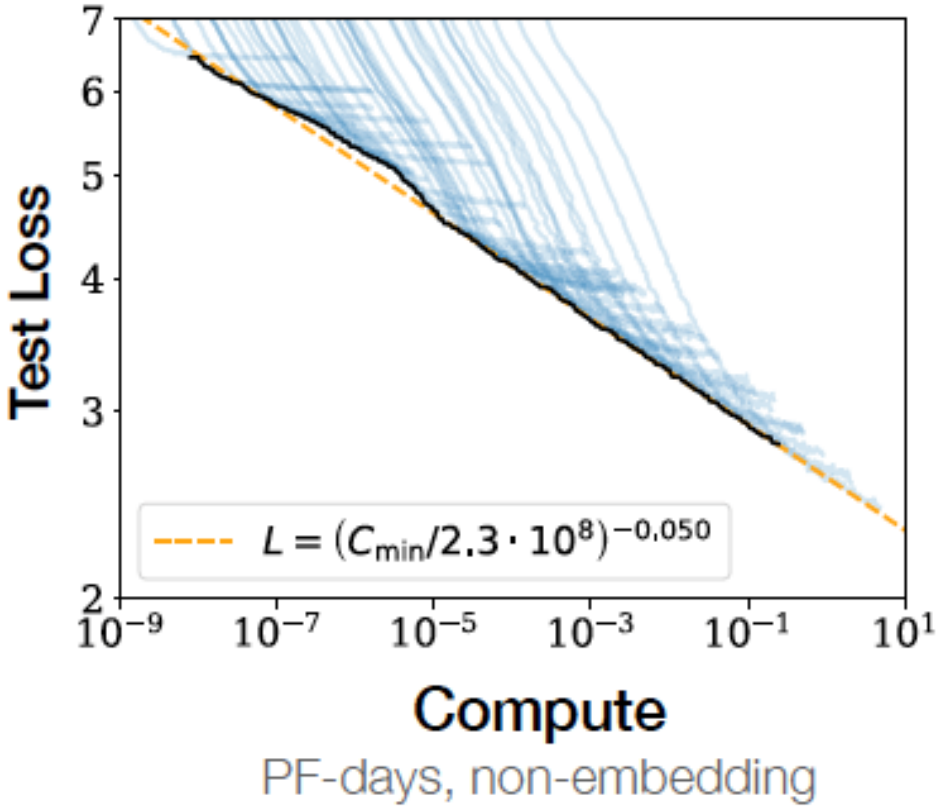
2020年のGPT-3登場後，2022年後半から加速度的に増加.



[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

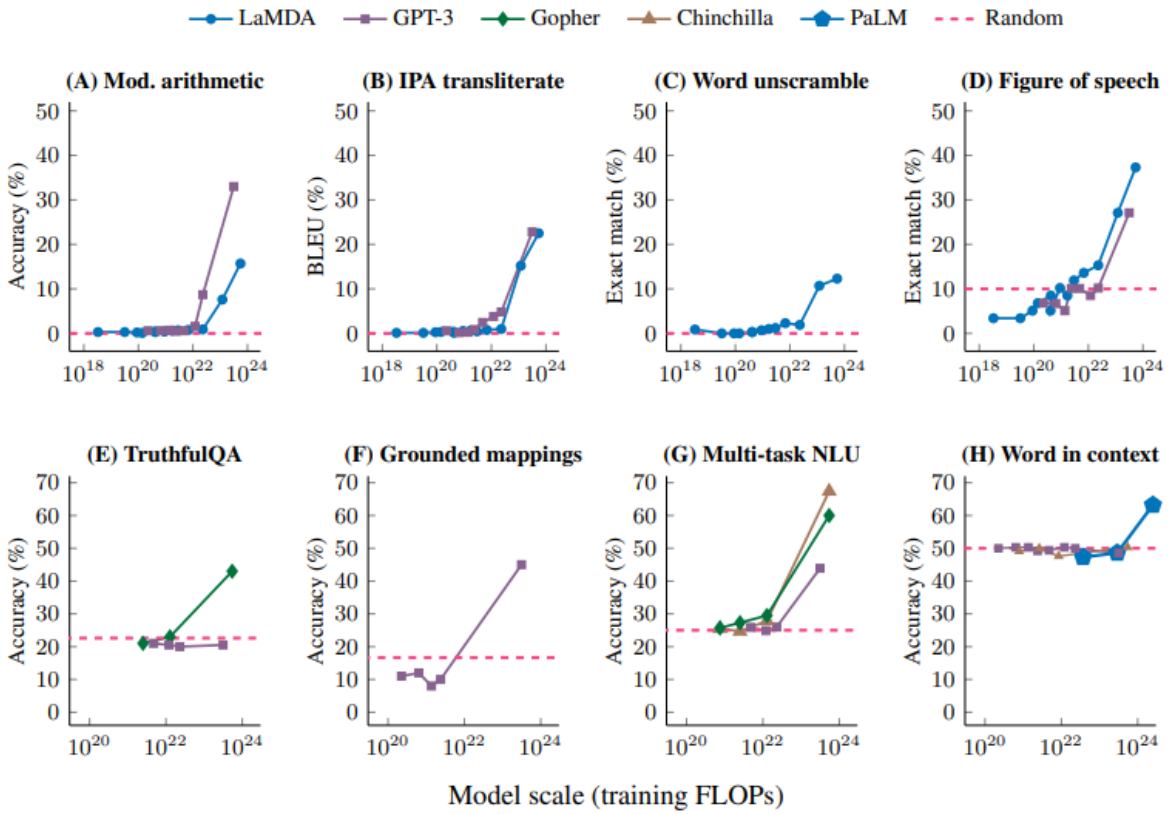
Scaling Law



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*

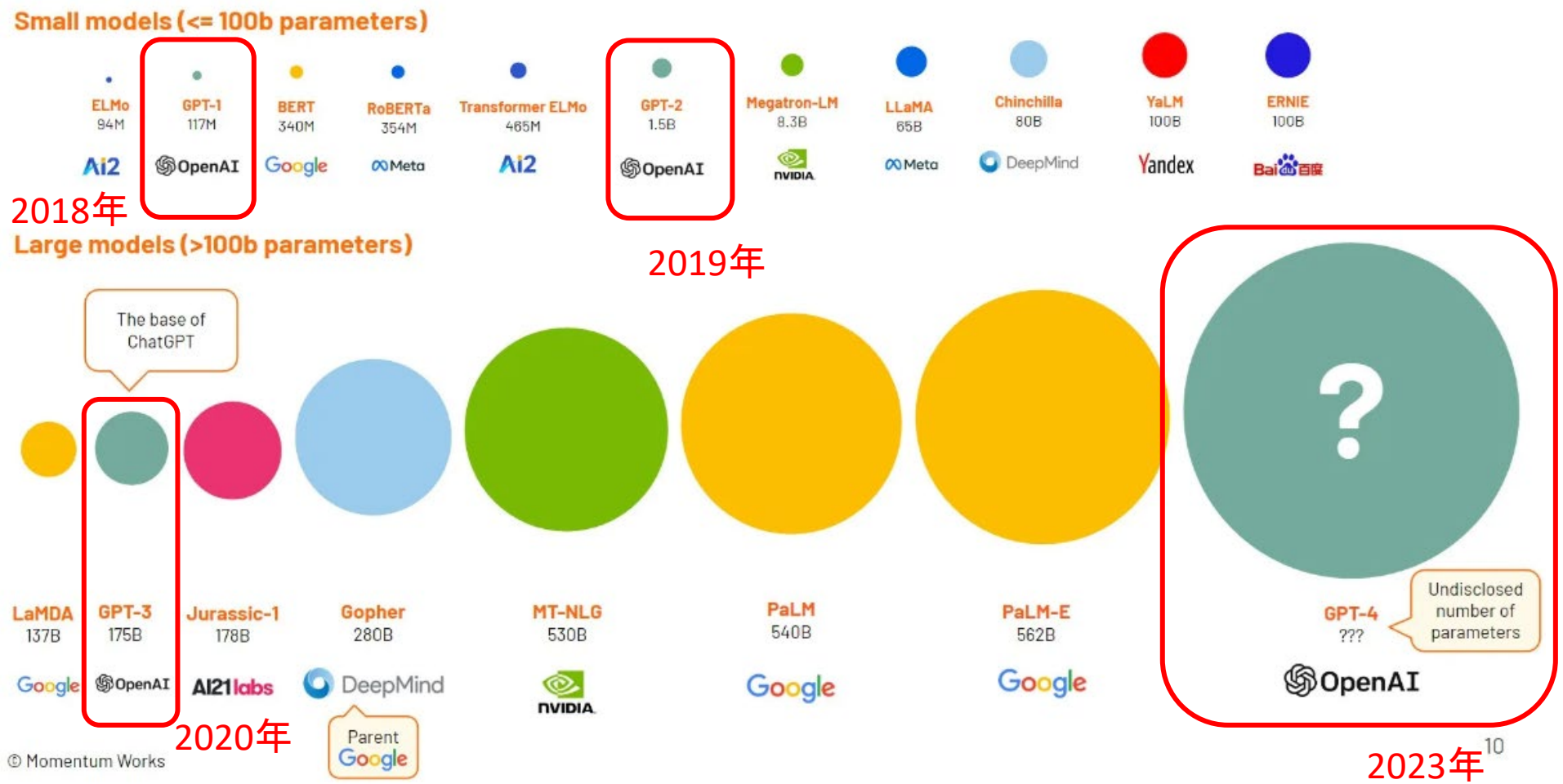
Emergent Ability



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

“Language Models are Few-Shot Learners”, 2020

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

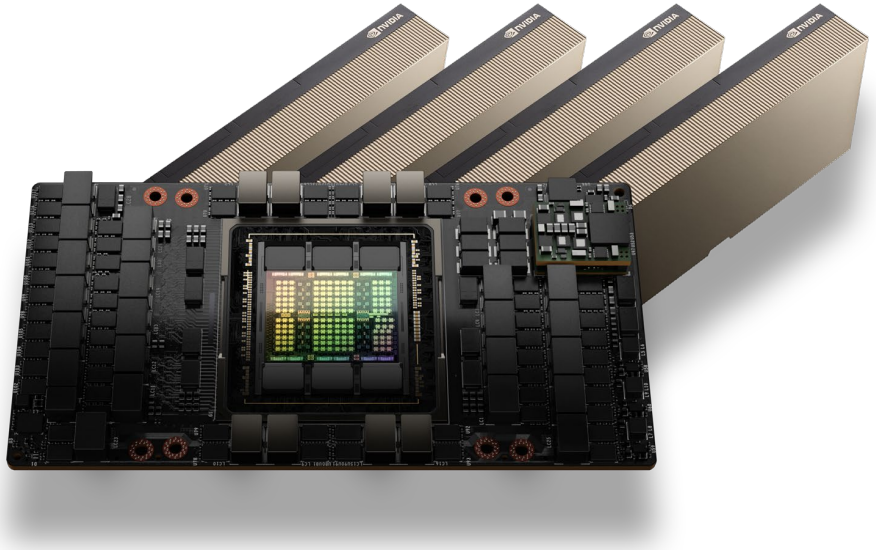
  参考：東大図書館が**約130万冊**,

  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）

ABCI
AI Bridging Cloud Infrastructure

■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.html[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

Translation (Few-Shot)

```
Translate English to French:        ← task description
sea otter => loutre de mer          ← examples
peppermint => menthe poivrée        ← examples
plush girafe => girafe peluche      ← examples
cheese =>                           ← prompt
```

Translation (Zero-Shot)

```
Translate English to French:        ← task description
cheese =>                           ← prompt
```

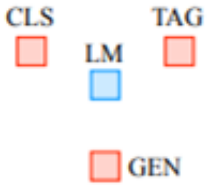
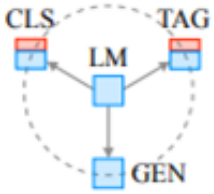
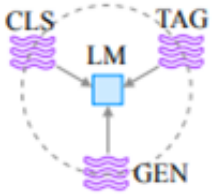
Summarization (Zero-Shot)

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

Many other examples

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

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

“On the Opportunities and Risks of Foundation Models”, 2021

■ 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

“Evaluating gpt-4 and ChatGPTt on Japanese medical licensing examinations”2023

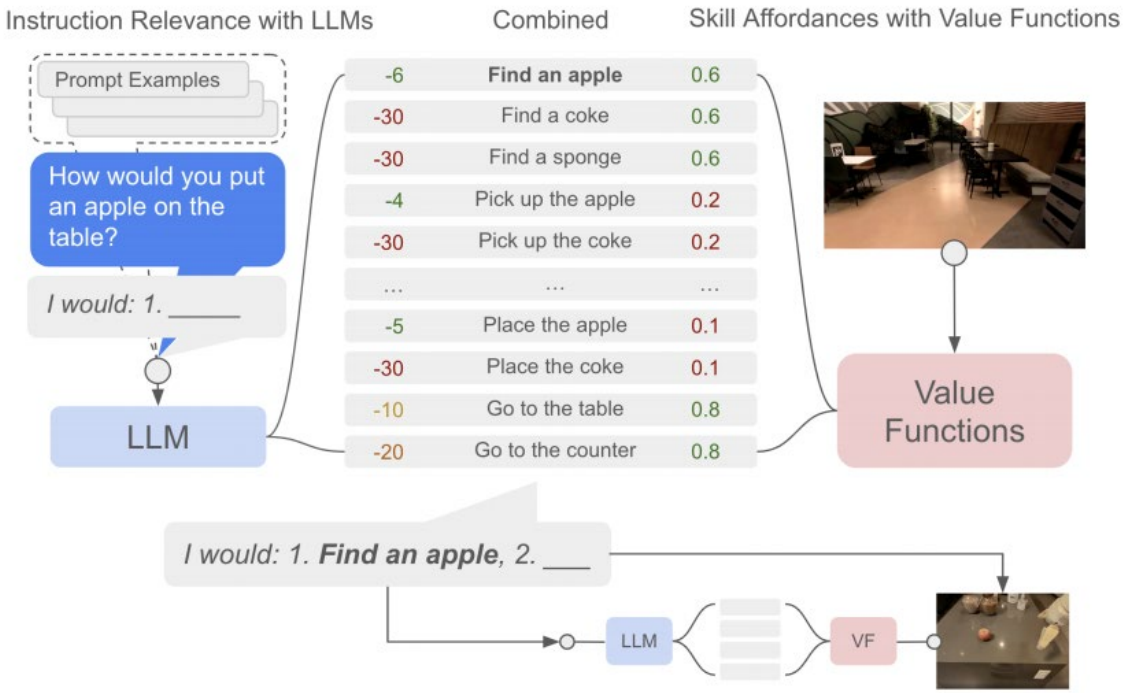
Igaku-QA | GPT-4の専門的知識の検証

	2018			2019			2020			2021			2022			2023		
Model	Req.	Gen.	P.↓	Req.	Gen.	P.↓	Req.	Gen.	P.↓	Req.	Gen.	P.↓	Req.	Gen.	P.↓	Req.	Gen.	P.↓
ChatGPT	123	143	1	100	150	5	118	148	2	143	154	3	124	163	2	120	140	–
ChatGPT-EN	123	158	2	117	157	3	116	147	2	110	167	0	140	187	1	142	159	–
GPT-3	105	104	5	93	117	5	97	111	4	94	109	3	106	111	6	86	113	–
GPT-4	161	221	0	170	215	1	168	219	0	173	225	0	164	228	1	170	221	–
Student Majority	196	276	0	196	274	0	195	276	0	200	277	0	195	287	0	–	–	–
Total	200	299	33	200	296	40	197	299	26	200	300	26	197	297	26	200	295	–
Passing Score	160	208	3	160	209	3	158	217	3	160	209	3	157	214	3	160	220	–

[12] Jungo Kasai et al. (2023), “Evaluating gpt-4 and ChatGPTt on Japanese medical licensing examinations” より引用

- 言語モデル (GPT-4 and ChatGPT）を新たに作成した日本の医療ライセンス試験6年分のデータセット（Igaku-QA)を構築してベンチマーク
- (1) 人間の平均的な受験者よりは悪い，(2) 禁忌技を選択する傾向にある，といった問題はあるものの試験ボーダーは突破

■LLMの活用 | Say-Can and Say-Can-PaLM

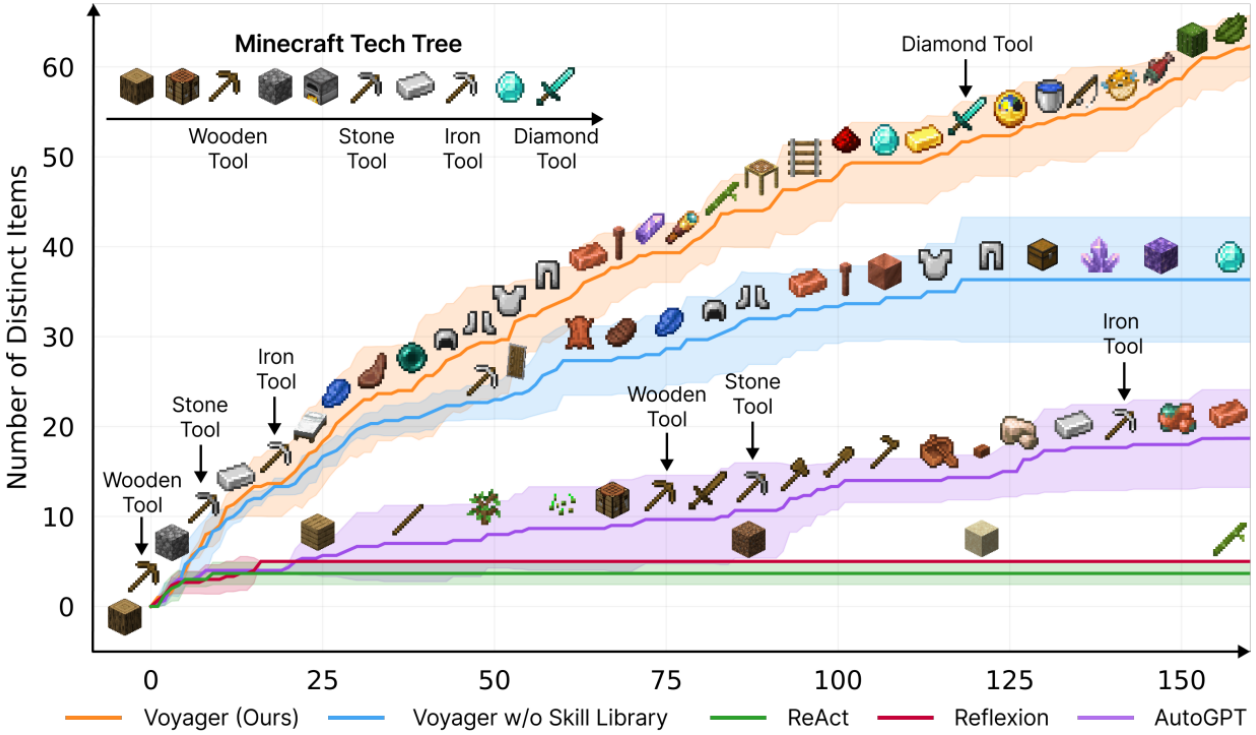


[13] Michael Ahn et al. (2022),“Do As I Can, Not As I Say: Grounding Language in Robotic Affordances” より引用

- 言語モデルが出力したスキルの実行可能性（Skill Affordance）を考慮して選択
  - 実行可能性はTDで学習
- 言語モデルをよくする（PaLMを使う）と性能が改善する

※ 実行可能なスキル（低レベル方策）はあらかじめ用意されている点に注意

Voyager | 言語モデルを使った方策の獲得



- LLMを使ってMinecraftをプレイする （右が動画）
  - 逐次的な行動獲得が必要，スパース報酬
  - RLが苦手
    （cf. Dreamre v3が初めてスクラッチで採掘に成功
- スキルをコードとして書く+LLMでプランニング

[14] Guanzhi Wang et al. (2023),
“Voyager: An Open-Ended Embodied Agent with Large Language Model” より引用

なぜいまLLMを学ぶのか？ 3. 他領域への影響

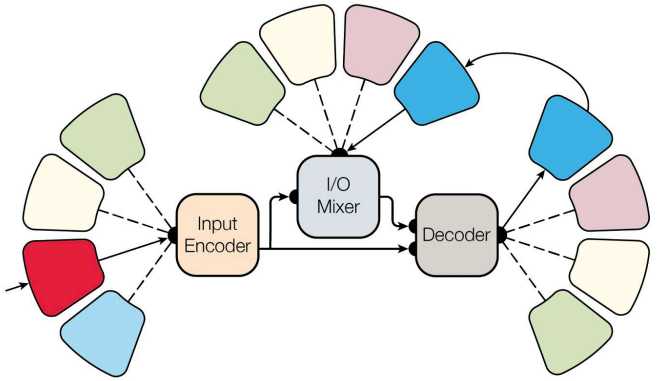
1. 方法論の共通化
(別ドメインでの大規模モデル構築)

大規模モデル（Transformer）
例：Gato, RT-1, X-Former
例：Dreamer v3

＋ 大規模なデータ
例: SAMでの1Bのマスクデータ
例：Gato, RT-1

＋ 大規模計算
例：スケール則は別ドメインでも成立

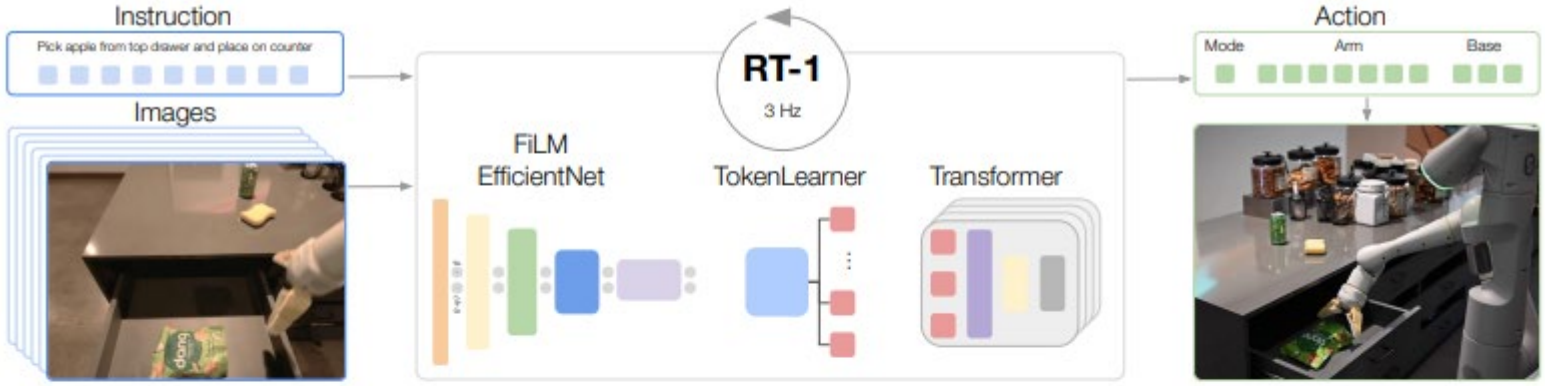
2. ドメインを超えたモデル共有



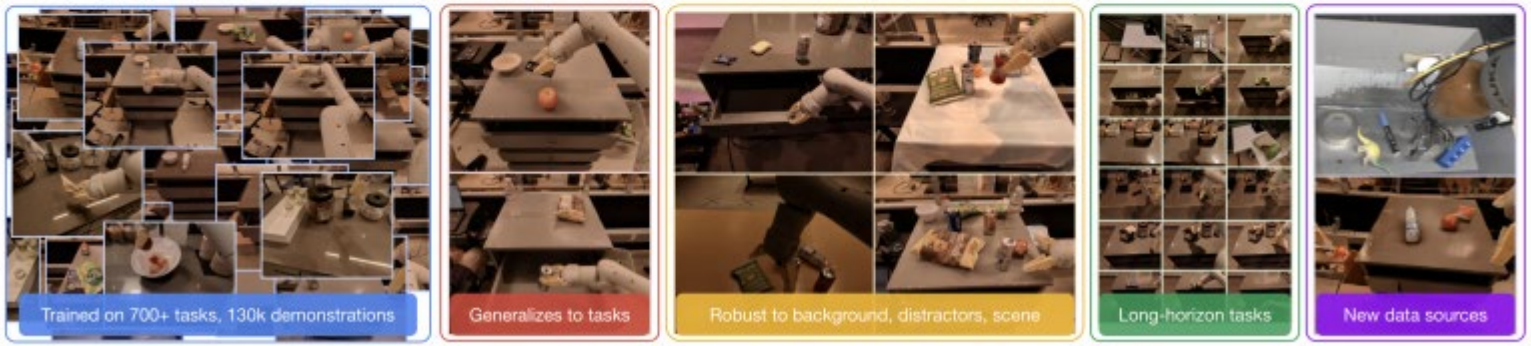
マルチモーダル化
”One model to Learn Them All”, 2017[15] 的な世界観（上図）
例： GPT4
言語モデルの活用
例：Say-Can, Voyager, など

“RT-1: Robotics Transformer for Real-World Control at Scale”, 2022

Robot Transformer（RT-1）



(a) RT-1 takes images and natural language instructions and outputs discretized base and arm actions. Despite its size (35M parameters), it does this at 3 Hz, due to its efficient yet high-capacity architecture: a FiLM (Perez et al., 2018) conditioned EfficientNet (Tan & Le, 2019), a TokenLearner (Ryoo et al., 2021), and a Transformer (Vaswani et al., 2017).



(b) RT-1's large-scale, real-world training (130k demonstrations) and evaluation (3000 real-world trials) show impressive generalization, robustness, and ability to learn from diverse data.

[16] Anthony Brohan et al. (2022), “RT-1: Robotics Transformer for Real-World Control at Scale” より引用

モデル

- Efficient NetとTransformerの組み合わせ
- インストラクションに従い動作生成

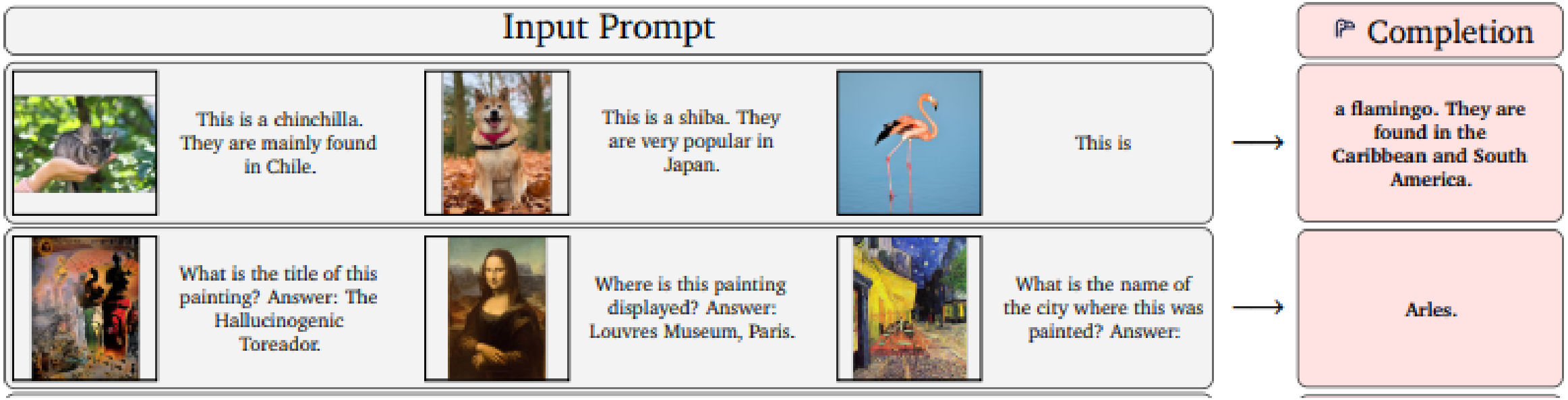
データ

- EDR13台，17ヶ月，744タスク，13万デモ
- 訓練：97%で動作
- 汎化：種々の意味で大幅向上（未知タスク，未知ソース等
- Long Horizonなタスクも可

※ 類似研究にGato，BC-Zなど

“Flamingo：a Visual Language Model for Few-Shot Learning”, 2022, DeepMind

マルチモーダルデータを扱う大規模モデルの例 | Flamingo



[17] Jean-Baptiste Alayrac et al. (2022), “Flamingo : a Visual Language Model for Few-Shot Learning”, NeurIPS2022 より引用

- 学習済Vision Model(NF-Net) とLanguage Model (Chinchilla, 70B) を統合．計80B.
  - ペアデータで接続部分 (Perceiver ResamplerとGated Xattn) .
- フラミンゴの写真を見て「フラミンゴ。カリブ諸島や南アメリカで見られます」などと返すなど画像・言語で様々な補完ができる.

https://www.deepmind.com/blog/tackling-multiple-tasks-with-a-single-visual-language-model[18]

ここまでのまとめと本講座の趣旨

■ ここまでのまとめ

- 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

- 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に（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

■本講座の趣旨

- LLMの技術的背景，原理を理解することで，ハイプとしてではなく活用する技術として捉えられるようになる

小島 武（こじま たけし）

❏略歴

❏2023.3 東京大学大学院 工学系研究科 技術経営戦略学専攻 博士課程修了

❏2023.4～ 東京大学大学院 工学系研究科 技術経営戦略学専攻 特任研究員

＊以前はITエンジニアをしていました.

❏研究分野、興味分野

❏深層学習、大規模言語モデル

❏基盤モデルの効率的な知識転移に関する研究

❏画像の基盤モデル（ViT）のテスト時適応の改善

❏LLMの思考の連鎖（CoT）による推論能力の改善

❏最近：LLMの構築. Weblab-10Bの開発

https://soco-st.com/13472[19]

目次

- LLMの概況
- **各回の概要**
- 日本のLLMを取り巻く環境

講座を組み立てるにあたって

まず, LLMの活用法を知ってもらう
（作った後の話）

次に, LLMの作り方（モデル, 学習, データ, etc)
について理解してもらう

最後に, LLMの最前線を知ってもらう

https://deeplearning.jp/llm2023/

各回の概要

- 第1回：Overview of Language Models  いまココ
- 第2回：Prompting and Augmented Language Model
- 第3回：Pre-training Pipeline
- 第4回：Scaling Pre-training
- 第5回：Parameter Efficient Fine-Tuning
- 第6回：RLHF
- 第7回：Going Beyond LLM

各回のダイジェストをお話します.

https://deeplearning.jp/llm2023/

各回の概要

- 第1回：Overview of Language Models
- 第2回：Prompting and Augmented Language Model
- 第3回：Pre-training Pipeline
- 第4回：Scaling Pre-training
- 第5回：Parameter Efficient Fine-Tuning
- 第6回：RLHF
- 第7回：Going Beyond LLM

LLMの活用法を知ってもらう

各回のダイジェストをお話します.

Prompting and Augmented Language Model（Day2）

- 目的：
  - ○ モデルのパラメータを変化させることなく、

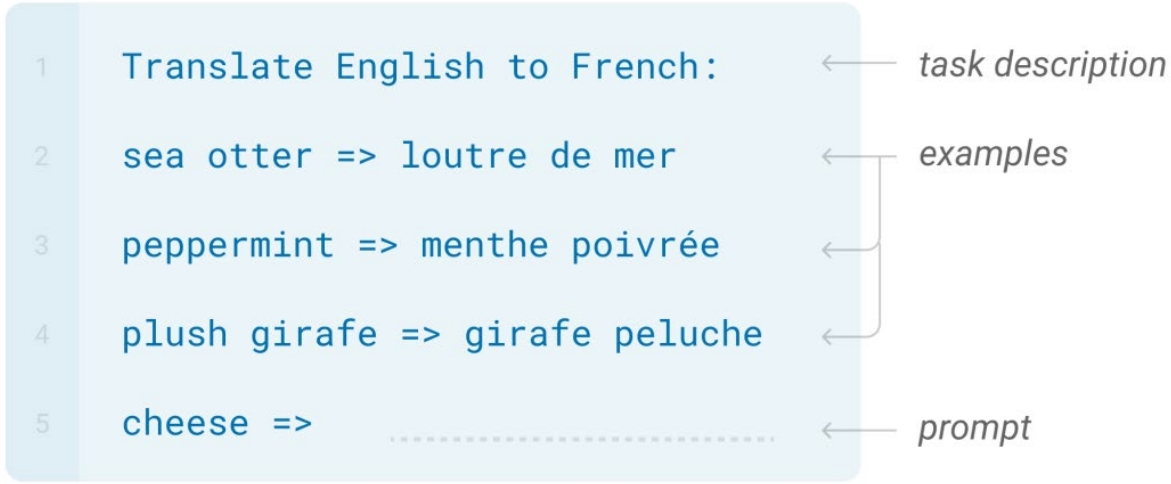
    LLMの性能を引き出す技術を会得する

- キーワード
  - ○ プロンプティング
  - ○ 文脈内学習（In-context Learning）
  - ○ 外部ナレッジ参照による性能の底上げ (Augmented Language Models)

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]

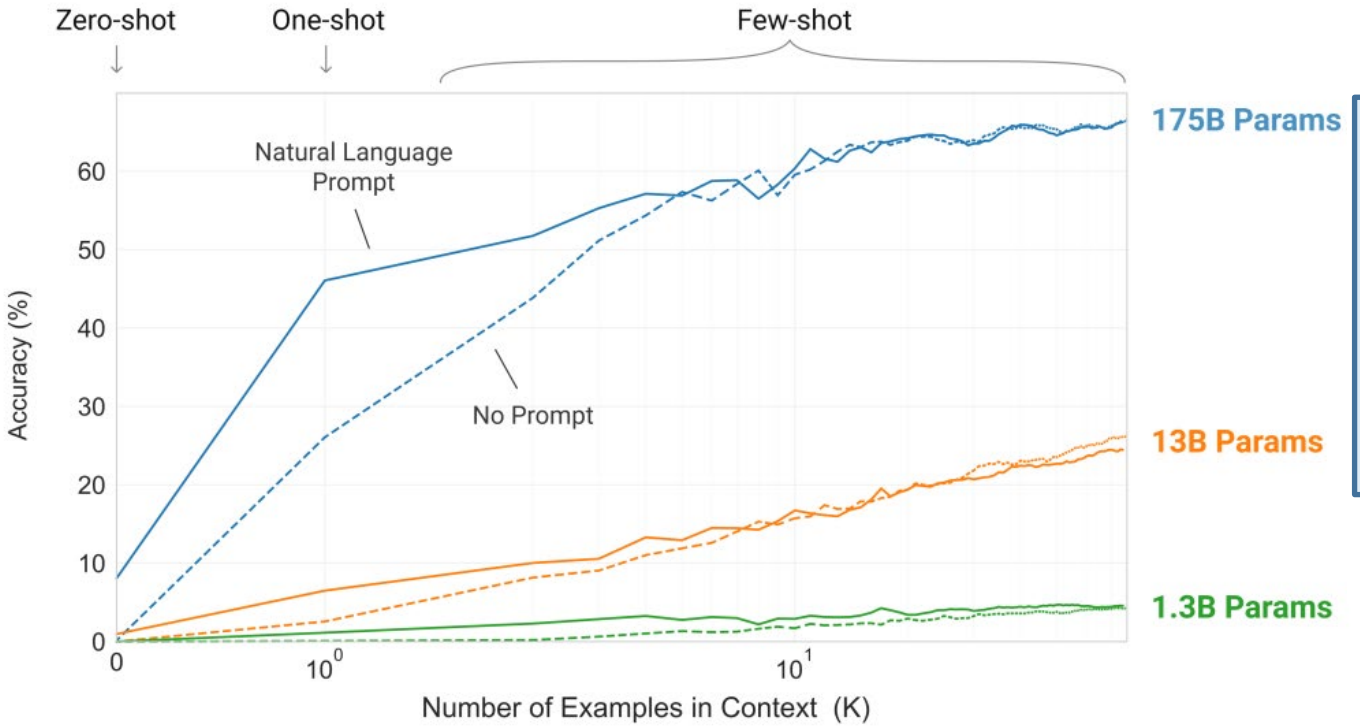
例：According toをつけると知識を参照してくれるようになる [2]

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

“Language Models are Few-Shot Learners”, NeurIPS2020

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



Demonstration (Few-Shot)

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

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がることが多い.

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

Chain-of-Though (CoT) Prompting

※ GSM8kは9-12歳の正解率が60%.

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?

Model Output

A: The answer is 27. ✗

Chain of Thought Prompting

Input

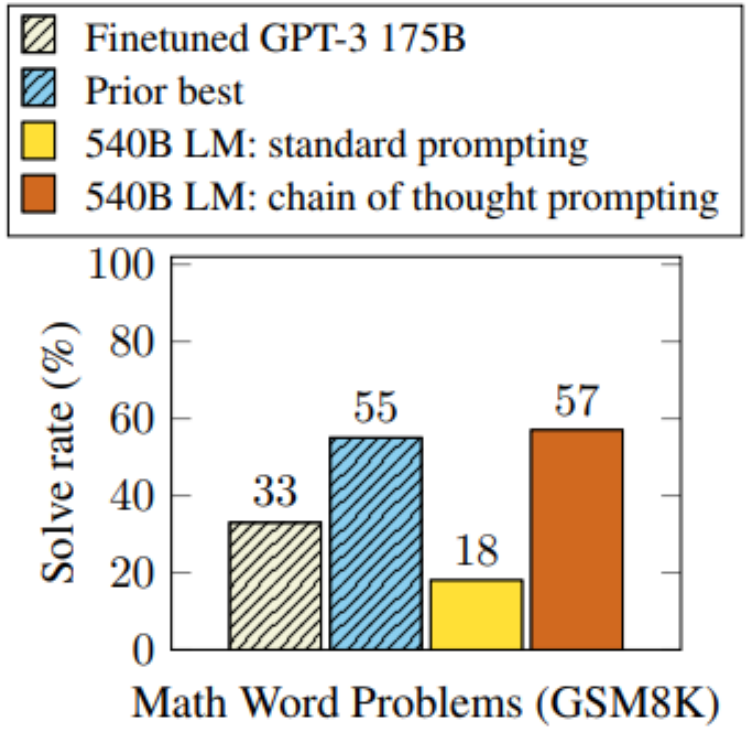
Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

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?

Model Output

A: 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. ✓



[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.

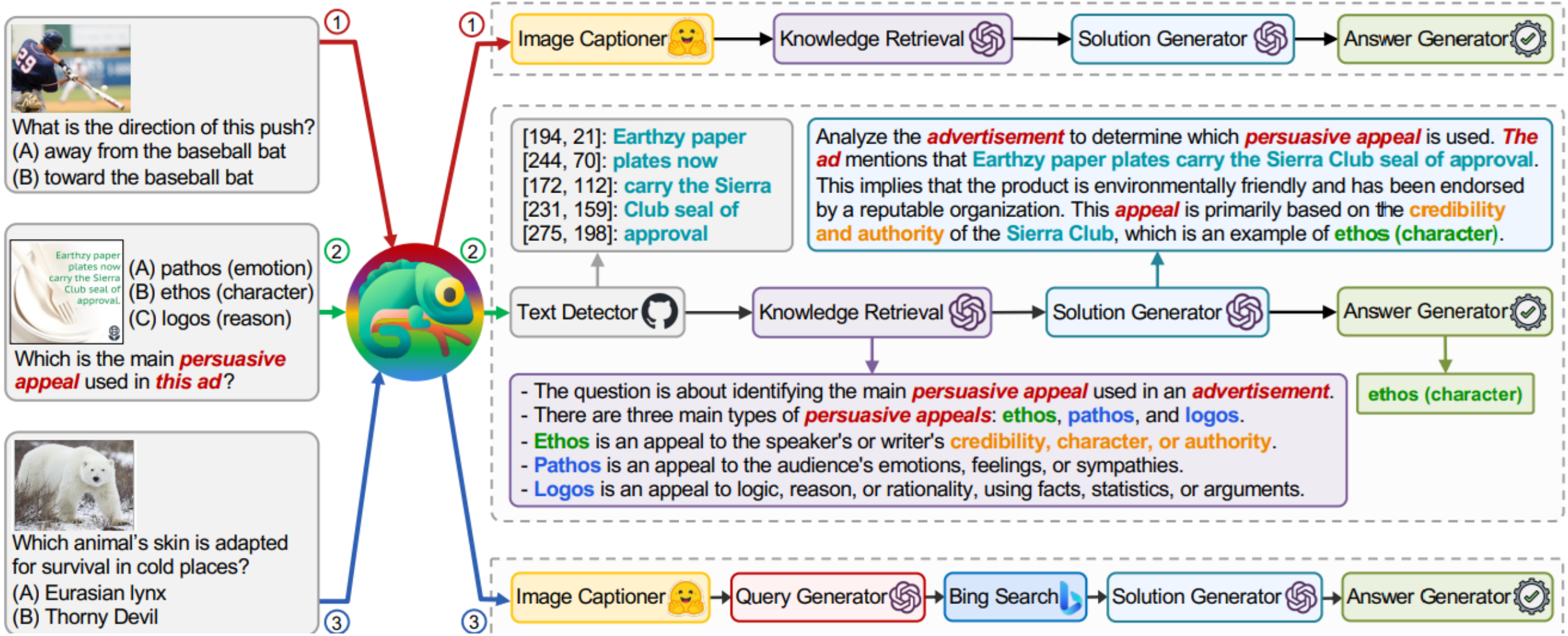
The Brown Act is California's law [WikiSearch("Brown Act") → The Ralph M. Brown Act is an act of the California State Legislature that guarantees the public's right to attend and participate in meetings of local legislative bodies.] that requires legislative bodies, like city councils, to hold their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

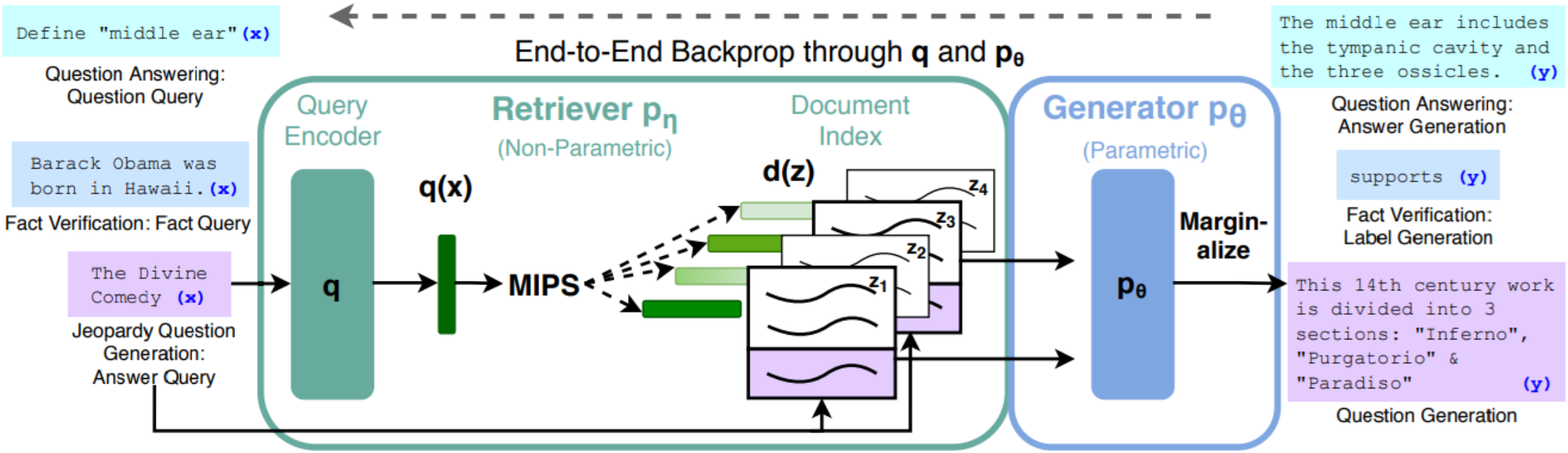
- The New England Journal of Medicine の登録商標者は、*[QA("The New England Journal of Medicineの発行元は？")→ Massachusets Medical Society]* MMSです。
- 1400人の参加者のうち、400人つまり *[計算機(400/1400)→0.29]* 29%が試験に合格した。
- その名前は"la tortuga"に由来しており，それはスペイン語で *[MT("tortuga") → 亀]* 亀です。

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）
  - LMの算出する尤度と回答の正確性が一部相関しているという研究も
  - いつtoolに頼るべきか、重みだけで算出するべきかALMの枠組みでは組み合わせられる
- 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っ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

各回の概要

- 第1回：Overview of Language Models
- 第2回： Prompting and Augumented Language Model
- 第3回：Pre-training Pipeline
- 第4回：Scaling Pre-training
- 第5回：Parameter Efficient Fine-Tuning
- 第6回：RLHF
- 第7回：Going Beyond LLM

LLMの作り方について
理解してもらう
（Part.1/4）

**各回の
ダイジェスト
をお話します.**

Pre-training Pipeline（Day3）

- 目的：
  - ○ LLMの主流なモデル構造であるTransformerと、
    その事前学習の仕組みを理解する.

- キーワード
  - ○ 事前学習
  - ○ Transformer
  - ○ Attention機構
  - ○ 大規模コーパス

LLM学習フロー

Step 1

事前学習

大規模コーパスによる自己教師あり学習を通し、大規模言語モデルに
語彙・文法・基本知識といった基礎的な言語理解を獲得させる段階

Step 2

ファインチューニング

ラベル付きデータによる教師あり学習を通し、事前学習済みモデルの
性能を改善したり、特定のタスクやドメインへの適応を実現する段階

Step 3

RLHF

人間からのフィードバックを用いた強化学習を通し、大規模言語モデルの
出力がより人間の価値観に沿ったものとなるよう調整する段階

事前学習とは

LLM以前

翻訳モデル

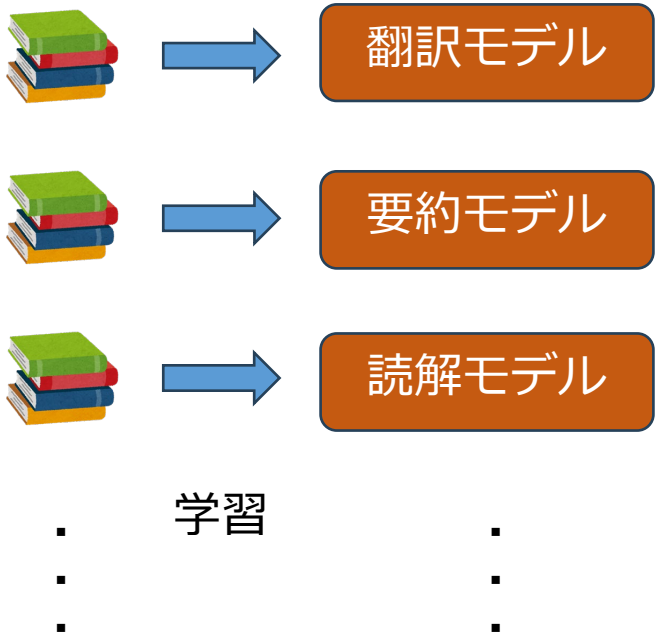
要約モデル

読解モデル

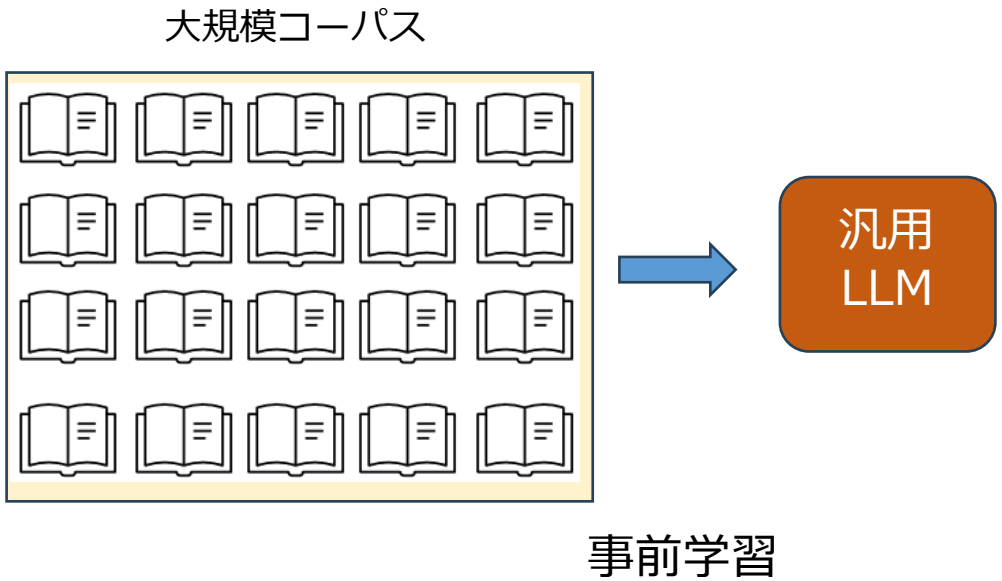
学習

事前学習とは

LLM以前

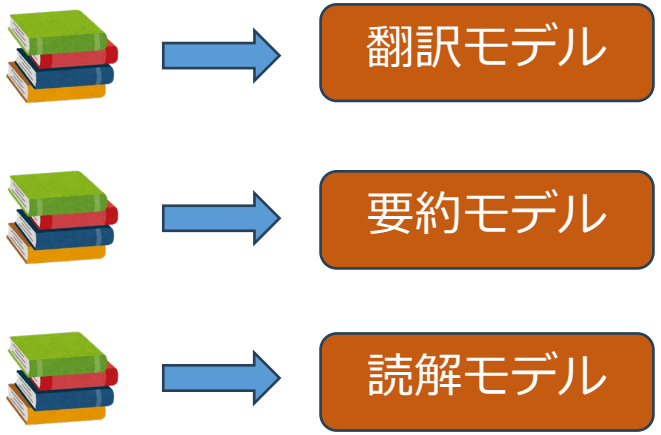


LLM

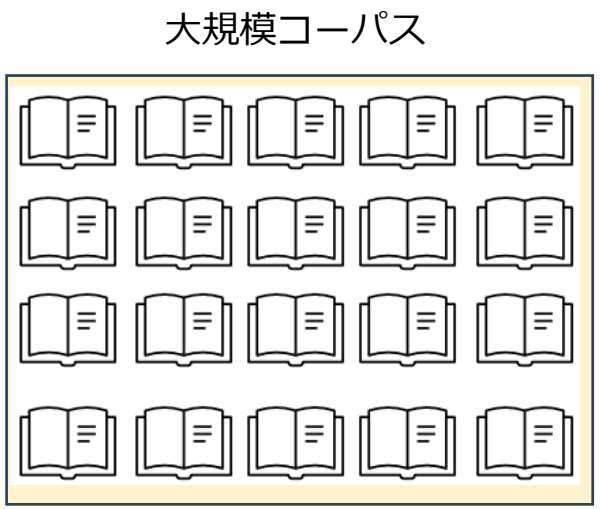


事前学習とは

LLM以前

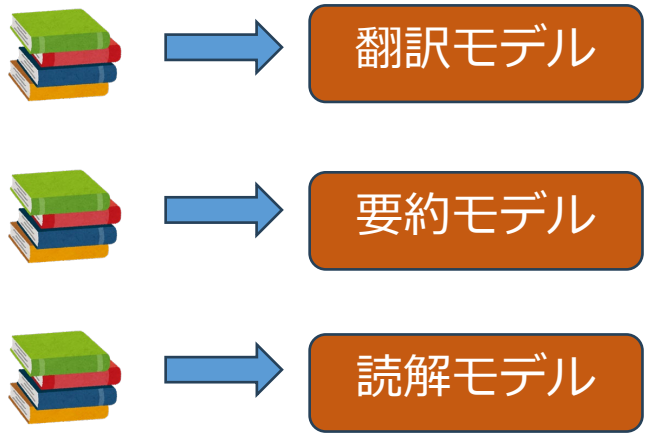


LLM



事前学習

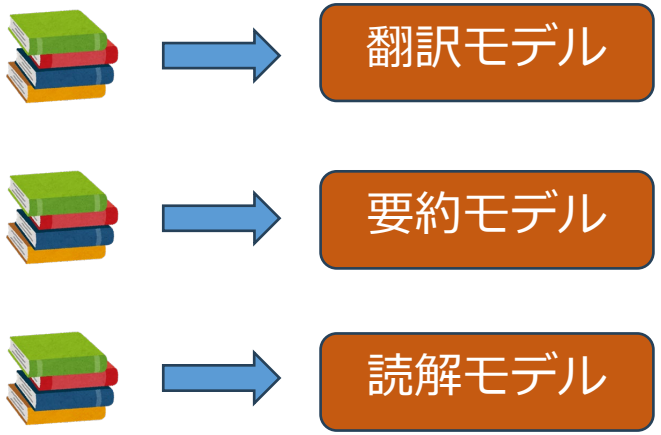
汎用
LLM



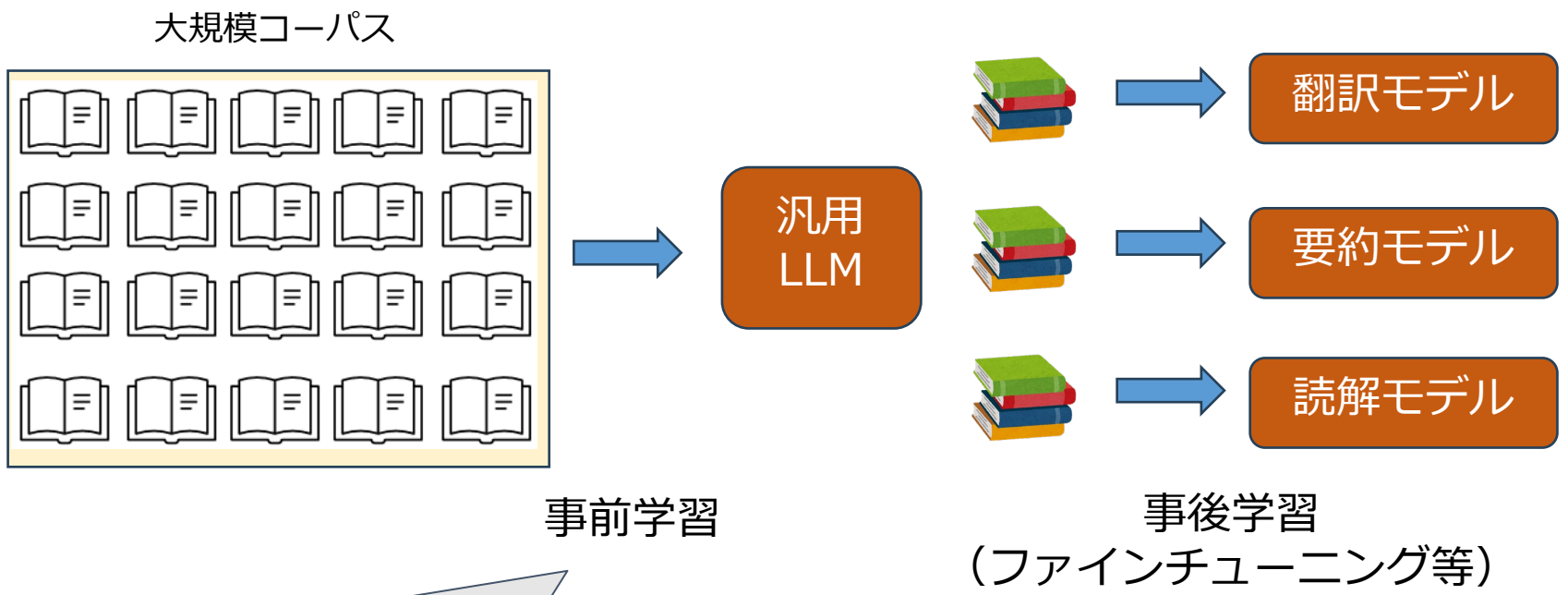
事後学習
（ファインチューニング等）

事前学習とは

LLM以前



LLM



・後続タスクに共通して必要な汎用知識を学習できる（例：読み書きそろばん）
・後続タスクのための良いパラメータの初期値が得られる.

LLMの事前学習方法（前半の言語モデルの解説復習）

- 単語の系列：$x_1, x_2, \cdots, x_L$，その生成確率$p(x_1, x_2, \cdots, x_L)$を割り当てる確率モデル：p.

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

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

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

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

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

- 条件付き確率がわかると，生成することもできる

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

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

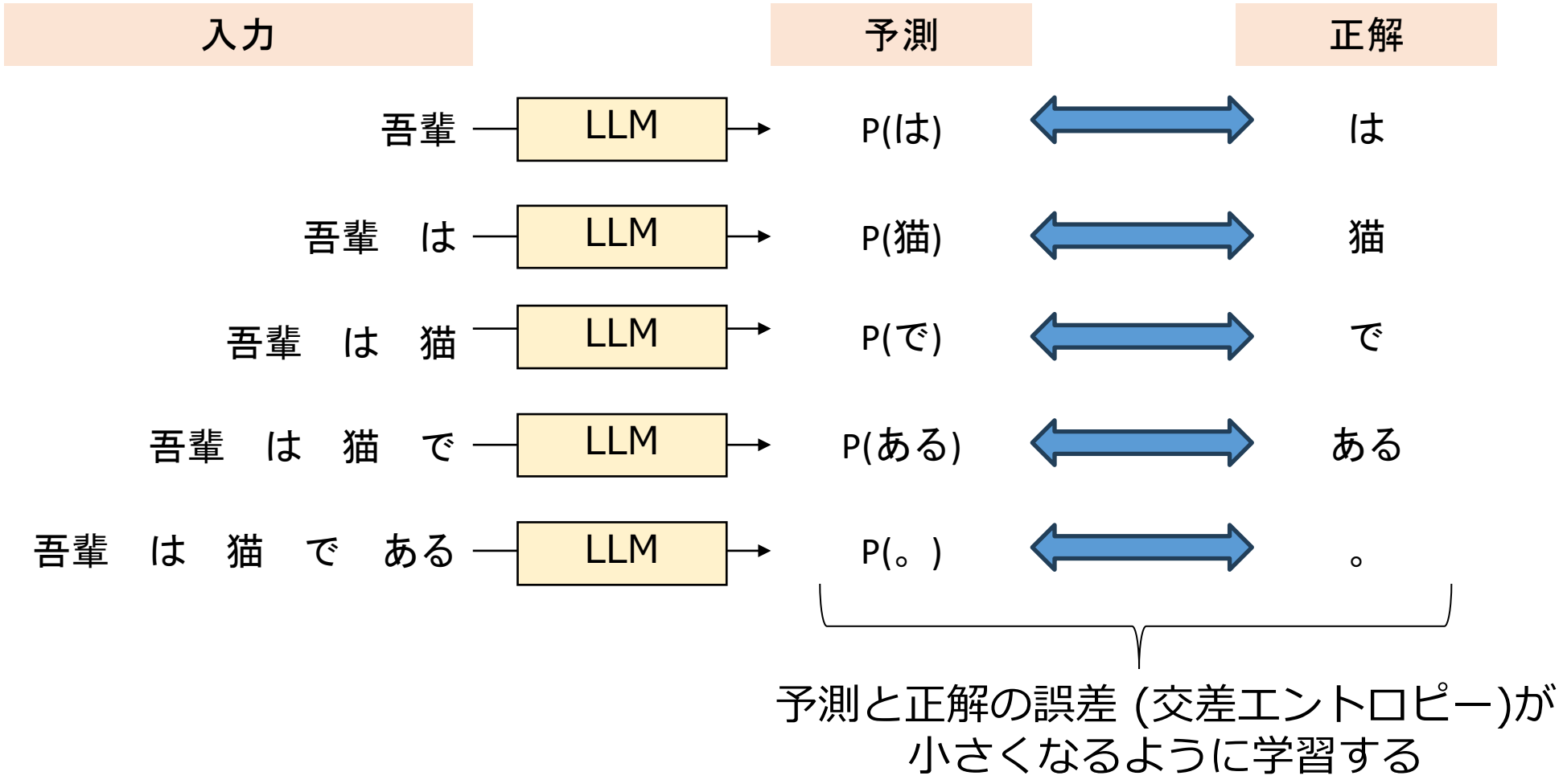
  $p(\text{カイロ}|\text{日本}, \text{の}, \text{首都}, \text{は}) = 0.000$

  } 日本の首都は → **東京**

- 生成確率をどう求めるか？どう学習するか？

LLMの事前学習方法

- Next Token Prediction（自己教師あり学習の一種）
  - 学習用のテキストデータを使って、次の単語の生成確率をひたすら予測する

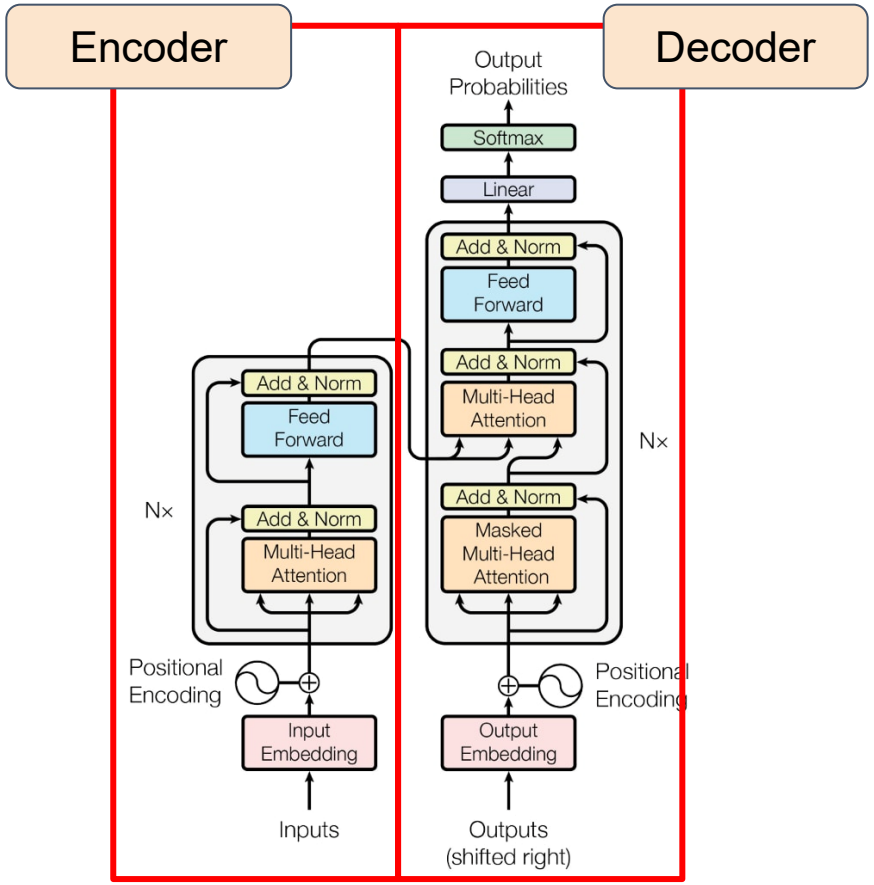


モデル構造

Transformerが主流（”Attention Is All You Need”という論文で初出）

アテンション機構を採用することで単語（トークン）の長距離依存関係を効率的に学習.

学習時の並列計算も効率化できたことで大規模化（分散学習）しやすくなった.



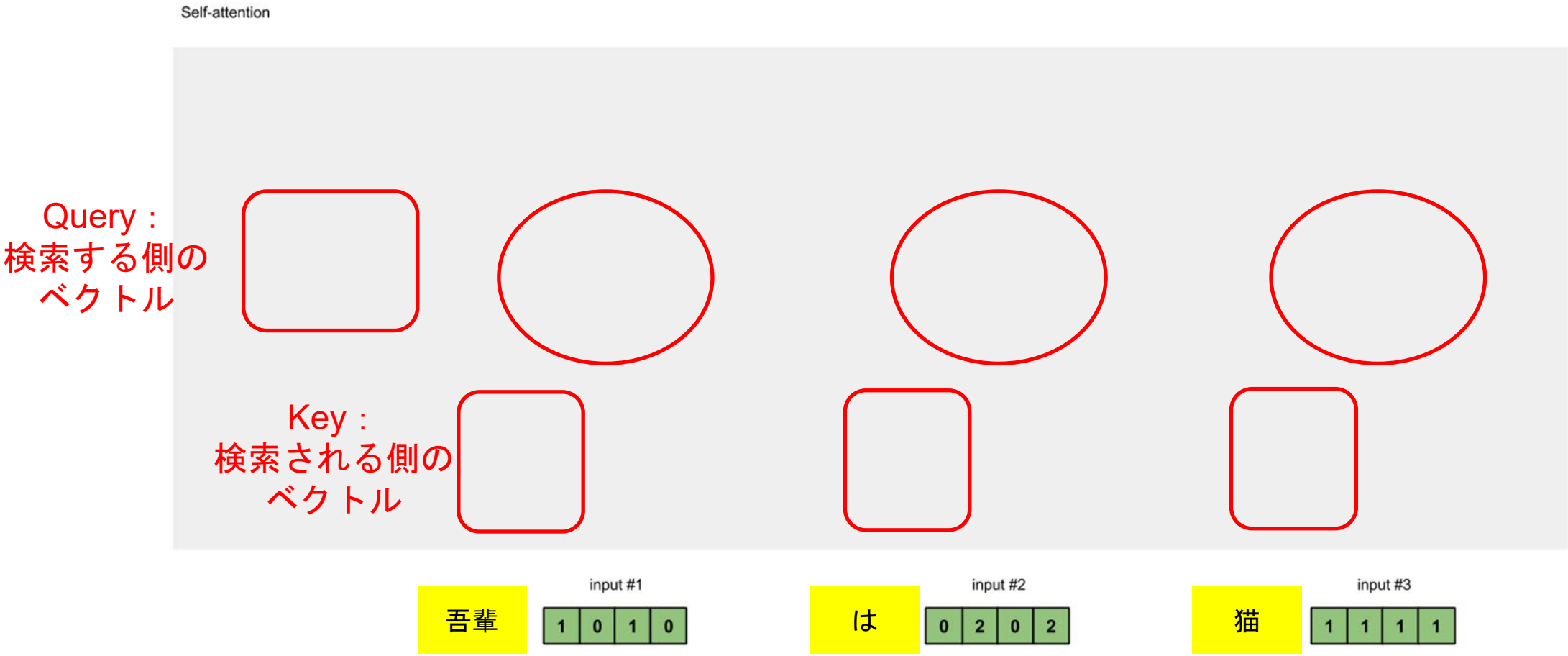
- Positional Embedding
- Transformer block
  - **GPT-3(175B)の場合, 潜在表現が12288次元のブロックを、96層積み重ねている (!!!!!)**
- Softmax
- Encoder
  - Multi-Head Self Attention
  - Position-wiseな全結合層
- Decoder
  - Multi-Head Source-Target Attention
- Layer normalization

[1] Ashish Vaswani et al. (2017) “Attention Is All You Need” NeurIPS 2017 より引用し,一部改変

“Attention Is All You Need”, 2017

モデル構造

アテンション機構：**全**単語（トークン）間の類似度を測ることによって、長距離の依存関係を把握することが可能.　＊類似度はベクトルの内積で測る.

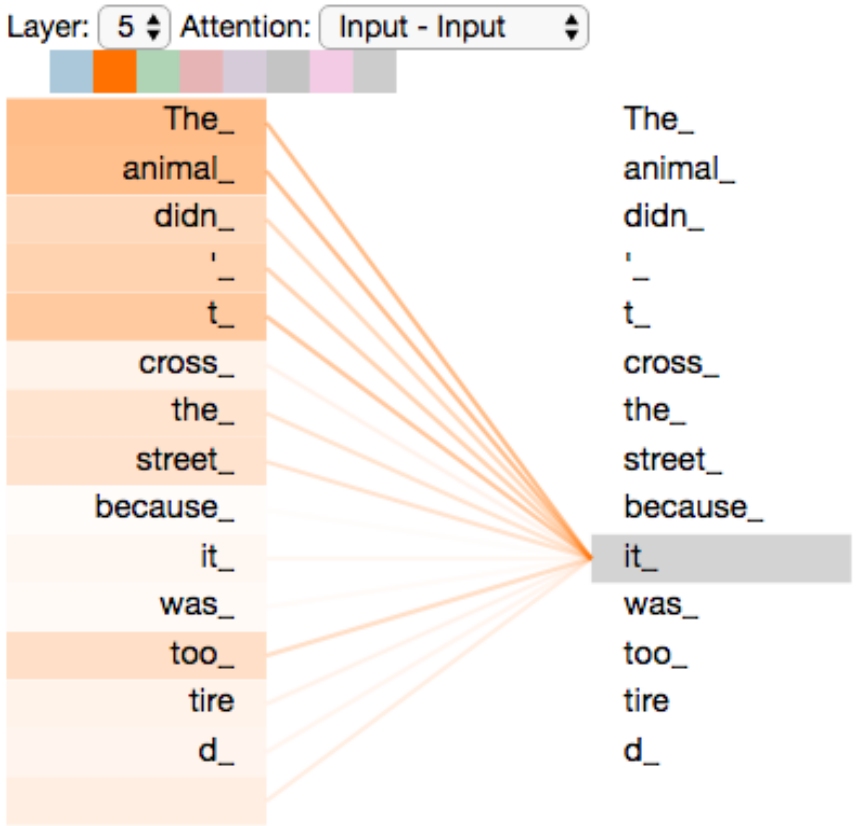


[24] Raimi Karim (2019) Illustrated: Self-Attention. A step-by-step guide to self-attention… | by Raimi Karim | Towards Data Science より引用し,一部改変

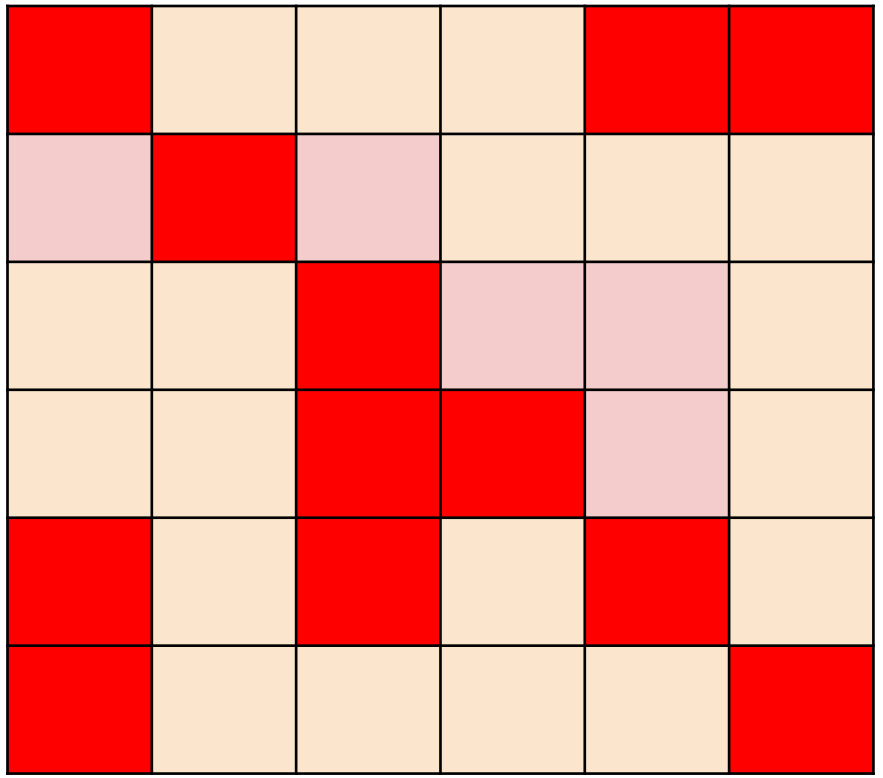
モデル構造

http://jalammar.github.io/illustrated-transformer/

アテンション機構：全単語（トークン）間の類似度を測ることによって、長距離の依存関係を把握することが可能.　＊類似度はベクトルの内積で測る.



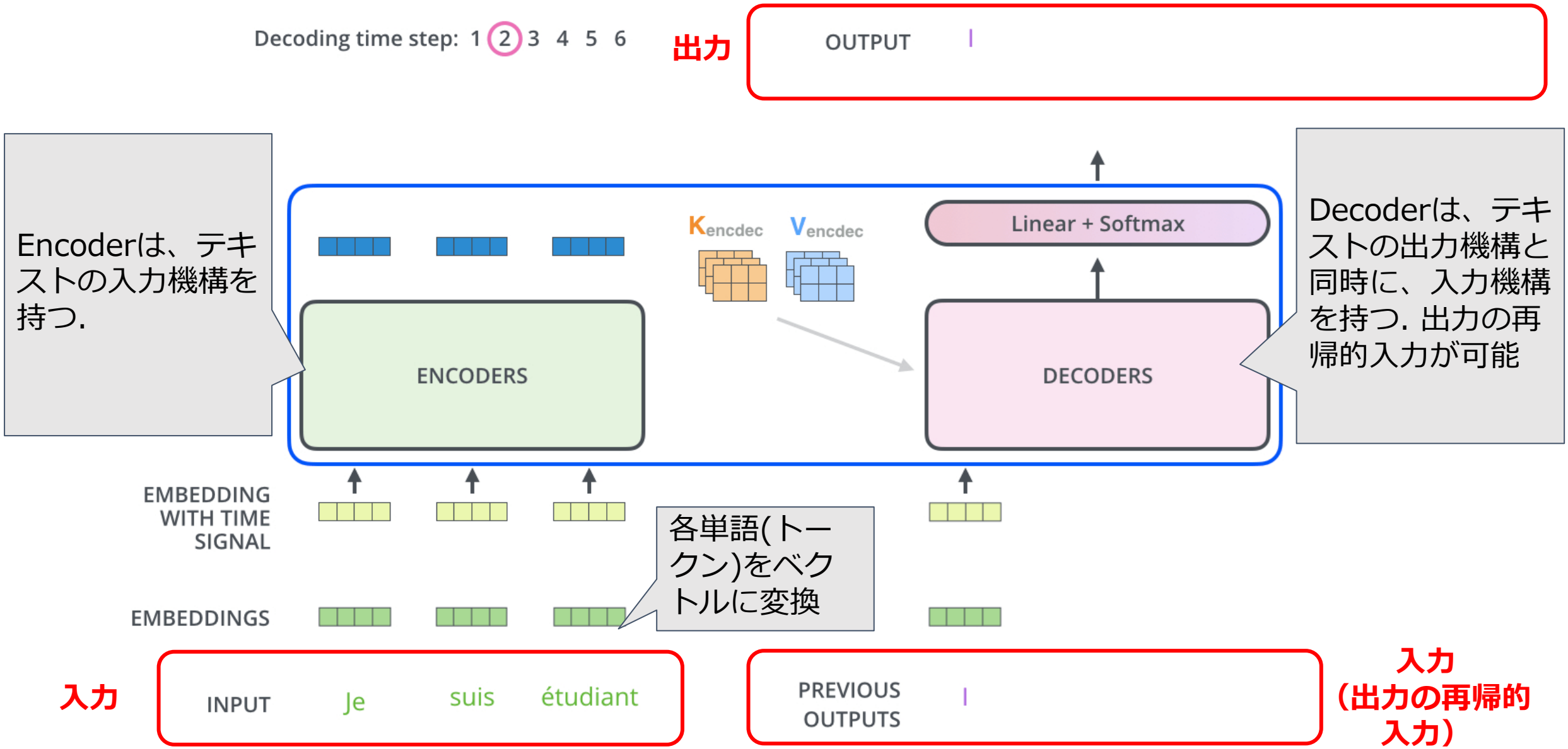
“it”は、”The” “animal” に対して強いアテンションがかかっていることがわかる.



全単語間のAttention Map（ヒートマップ）が作れる

[25] Jay Alammar (2018) The Illustrated Transformer – Jay Alammar – Visualizing machine learning one concept at a time. より引用

モデル構造（＊フランス語→英語の翻訳タスクで説明）



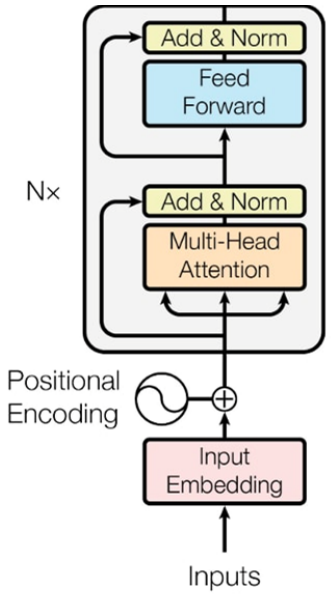
[25] Jay Alammar (2018) The Illustrated Transformer – Jay Alammar – Visualizing machine learning one concept at a time. より引用

Transformerの分類

Encoder-only

BERT、RoBERTaなど

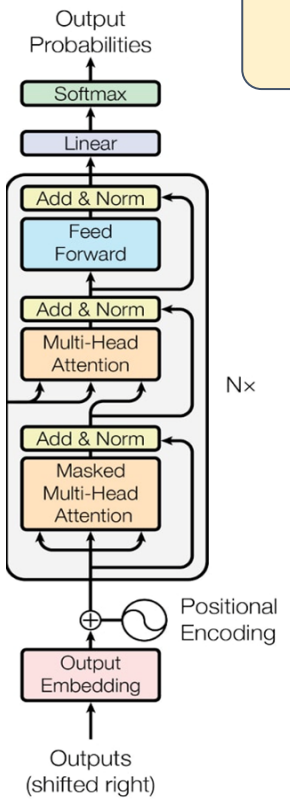
認識系
（クラス分類）



Decoder-only

GPT、PaLMなど

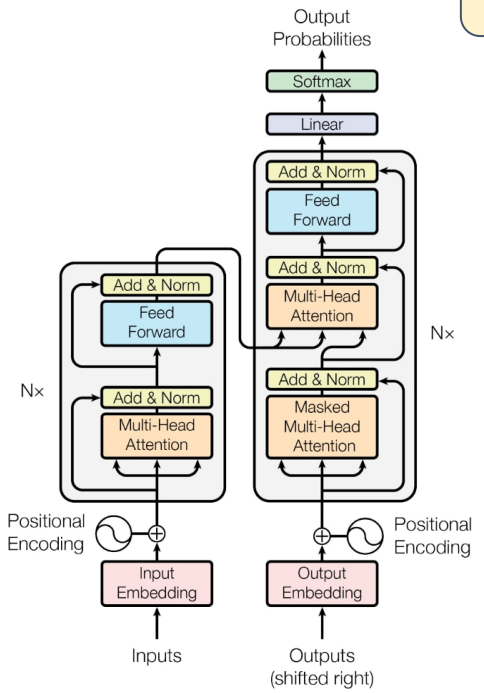
テキスト
生成系



Encoder-decoder

BART、T5など

テキスト
生成系



[1] Ashish Vaswani et al. (2017) “Attention Is All You Need” NeurIPS 2017 より引用し,一部改変

データの収集

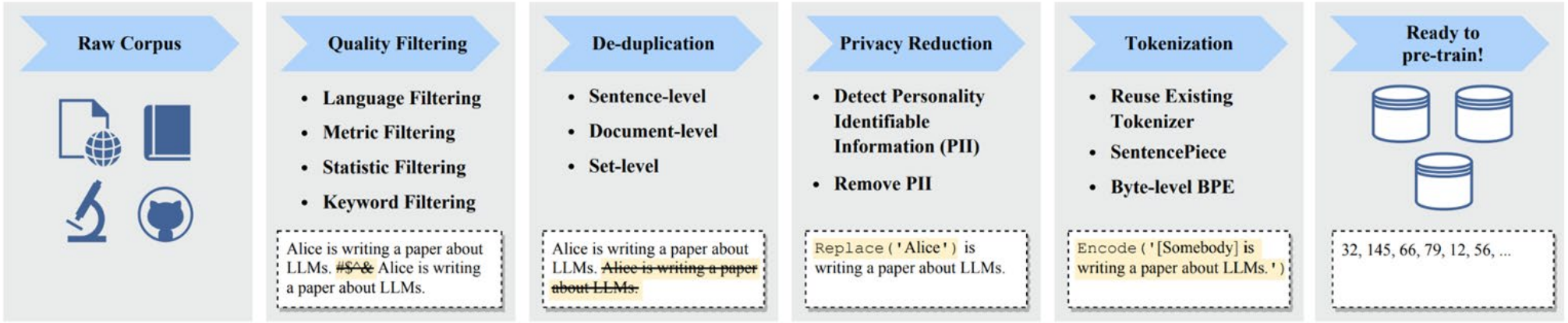
事前学習用データは、一般的にＷＥＢからの大規模クロールデータ

Dataset	Sampling prop.	Epochs	Disk size
CommonCrawl	67.0%	1.10	3.3 TB
C4 一般的なWEBサイト (ニュース, ブログ, HP)	15.0%	1.06	783 GB
Github プログラム言語	4.5%	0.64	328 GB
Wikipedia	4.5%	2.45	83 GB
Books 小説など	4.5%	2.23	85 GB
ArXiv 論文	2.5%	1.06	92 GB
StackExchange 技術QA	2.0%	1.03	78 GB

[26] Hugo Touvron et al. (2023), “LLaMA: Open and Efficient Foundation Language Models” より引用し,一部改変

データの前処理

LLMの事前学習のおいて典型的な前処理のパイプライン



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

- Quality Filtering

  分類器やヒューリスティックにより質の低いデータを取り除く
- De-dup

  近い場所で重複があると学習への悪影響が大きいため、文、文書、データセットなど様々な粒度で重複を排除する
- Privacy Reduction

  キーワードスポッティングのようなルールベースの手法でで個人を特定できる情報は取り除く
- Tokenization

  ＊次ページにて説明.

データの前処理

テキストのTokenization(トークン分割)

代表手法:Byte Pair Encoding (BPE)

- サブワードによるトークン化の一種
- 単語によるトークン化と各単語の頻度のカウント
- 語彙サイズ（基本語彙数＋マージ数）はハイパーパラメータ
- GPT、GPT-2、RoBERTa、BART、DeBERTaなど多くのTransformerで用いられる
- 絵文字などの処理
  - トークン化している例がトレーニングコーパスにない文字を使用している場合、その文字は<unk>に変換される
  - そのため、多くのNLPモデルが絵文字でコンテンツを分析するのが苦手としている
  - GPT-2とRoBERTaのトークナイザは、これに対処するためにバイトレベルでBPEをおこなう

“吾輩は猫である。”

↓

“吾輩”, “は”, “猫”, “で”, “ある”, “。”

例）’hug’, ‘pug’, ‘pun’, ‘bun’, ‘hugs’
（語彙 [‘b’,’g’, ‘h’, ‘n’, ‘p’, ‘s’, ‘u’] ）

1. それぞれの単語のコーパスでの出現回数をカウント
   ('hug', 10), ('pug', 5), ('pun', 12), ('bun', 4), ('hugs', 5)

2. 単語を文字に分割
   ('h' 'u' 'g', 10), ('p' 'u' 'g', 5), ('p' 'u' 'n', 12),
   ('b' 'u' 'n', 4), ('h' 'u' 'g' 's', 5)

3. 最も頻出のペアから、トークナイザーによって学習された最初のマージルールは (‘u’, ’g’) → (‘ug’) となる
   ('h' 'ug', 10), ('p' 'ug', 5), ('p' 'u' 'n', 12),
   ('b' 'u' 'n', 4), ('h' 'ug' 's', 5)

4. 希望の語彙サイズまで頻度の高い組のマージを繰り返す
   ('hug', 10), ('p' 'ug', 5), ('p' 'un', 12), ('b' 'un', 4), ('hug' 's', 5)
   （語彙 [‘b’,’g’, ‘h’, ‘n’, ‘p’, ‘s’, ‘u’, ‘ug’, ‘un’, ‘hug’] ）

各回の概要

- 第1回：Overview of Language Models
- 第2回：Prompting and Augumented Language Model
- 第3回：Pre-training Pipeline
- 第4回：Scaling Pre-training
- 第5回：Parameter Efficient Fine-Tuning
- 第6回：RLHF
- 第7回：Going Beyond LLM

LLMの作り方について理解してもらう（Part.2/4）

各回のダイジェストをお話します.

Scaling Pre-training（Day4）

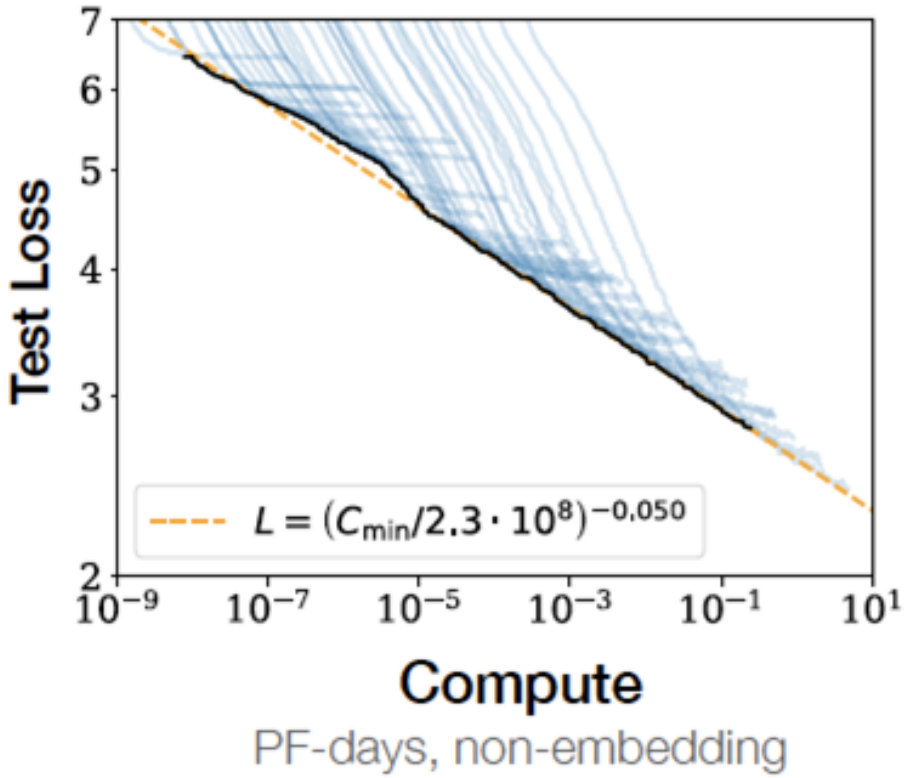
- 目的：
  - ○ モデルをスケールする理由を知る
  - ○ モデルのスケールにおける課題を知る
  - ○ スケールしたモデルを学習する方法について学ぶ

- キーワード
  - ○ スケール則
  - ○ パラメータ数、計算量、データ

スケーリングを動機付けるもの

モデル・計算量・データのスケーリングにより以下のことが実現される

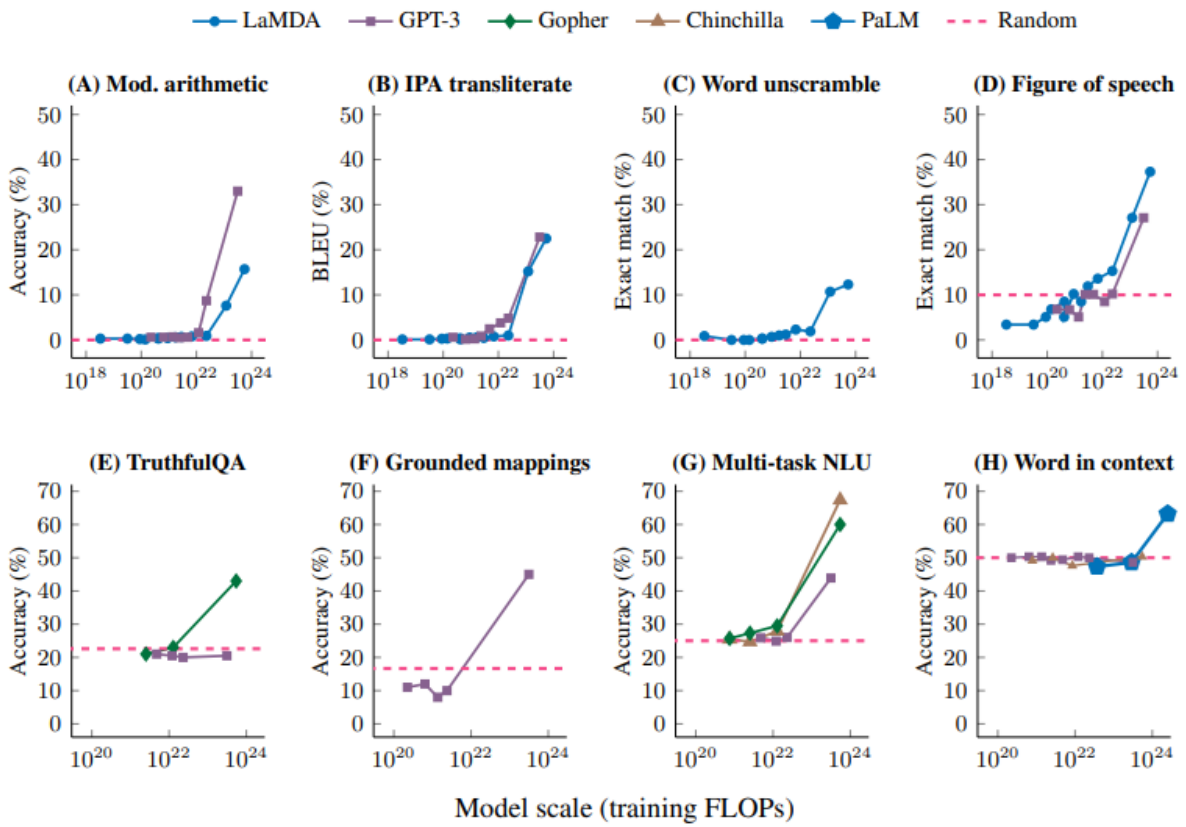
Scale Law



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)

スケールさせる上での問題点

各要素をスケールを困難にする課題

パラメータ数（N）：
モデルがスケール
するにつれて
コストが増加する

計算量（C）：
十分な計算量/
メモリ量を確保して
効率よく訓練する必要

データ(D)：
性能を発揮させるため
の学習用データを用意
する必要

用語説明　計算量

・単位：**FLOPS** (1秒間に浮動小数点演算を何回できるかという能力)

・**LLM学習に必要な総計算量（の近似 *Decoder-Onlyの場合）：**

モデルサイズ(パラメータ数)　×　学習データサイズ(トークン数)　×　6

（例）GPT3の場合

175B × 0.3T × 6 ≒ 3.14 * E+23 FLOPS

・**計算環境の計算能力：**

GPU V100 ×1基： *O*(E+13~E+14 FLOPS) *実際にはUtility Rateをかける.

GPU A100 ×1基： *O*(E+14 FLOPS) *実際にはUtility Rateをかける.

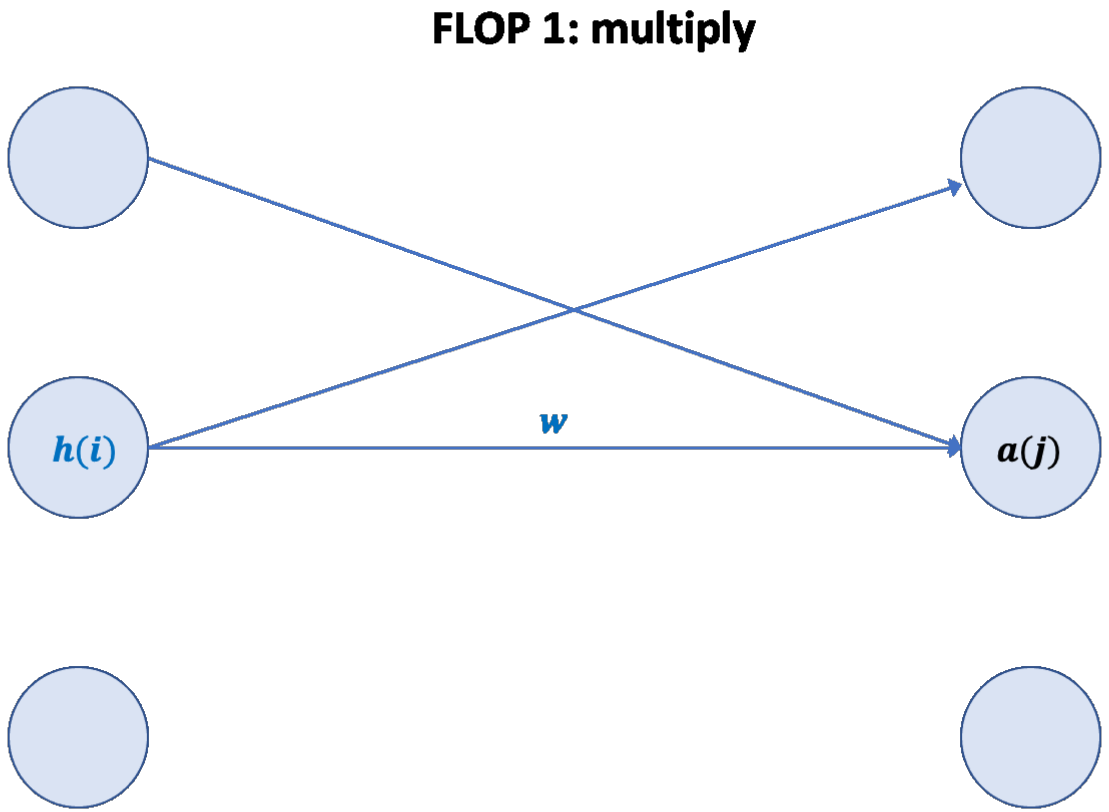
・**所要日数：**

LLM学習に必要な総計算量／計算環境の計算能力

[7] Tom Brown et al. (2020), "Language Models are Few-Shot Learners"を参考

用語説明　計算量

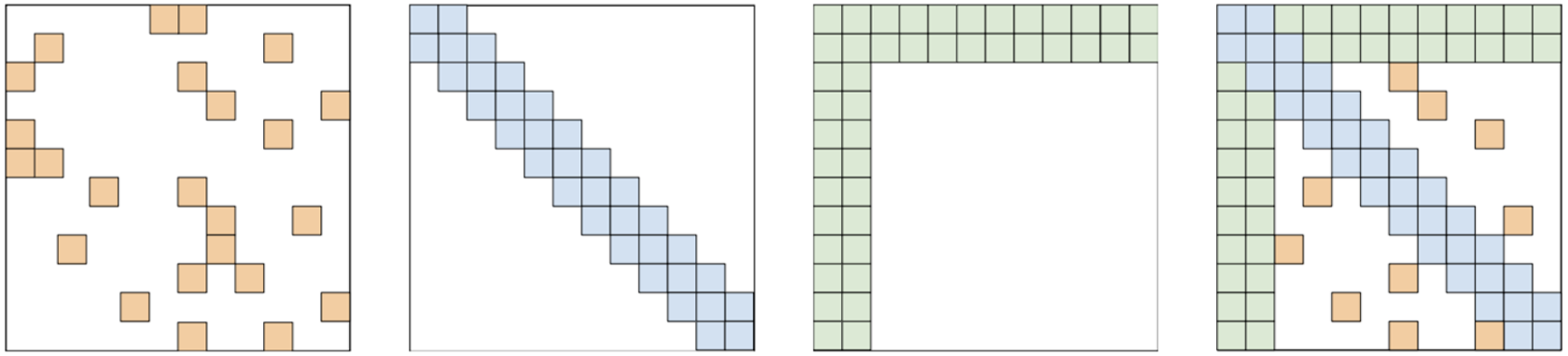
・なぜ６をかけるのか？



[27] Dzmitry Bahdanau (2022), The FLOPs Calculus of Language Model Training | by Dzmitry Bahdanau | Medium より引用

Sparse(疎)なAttentionの提案

パラメータ数の増加とともに
増大する計算コストの課題への対応例



(a) Random attention (b) Window attention (c) Global Attention (d) BIGBIRD

方法

複数のSparseな
Attentionを
組み合わせて,
Attentionを近似し,
長い系列に対応する

Model	HotpotQA			NaturalQ		TriviaQA		WikiHop
	Ans	Sup	Joint	LA	SA	Full	Verified	MCQ
HGN [26]	**82.2**	88.5	**74.2**	-	-	-	-	-
GSAN	81.6	88.7	73.9	-	-	-	-	-
ReflectionNet [32]	-	-	-	77.1	**64.1**	-	-	-
RikiNet-v2 [61]	-	-	-	76.1	61.3	-	-	-
Fusion-in-Decoder [39]	-	-	-	-	-	84.4	90.3	-
SpanBERT [42]	-	-	-	-	-	79.1	86.6	-
MRC-GCN [87]	-	-	-	-	-	-	-	78.3
MultiHop [14]	-	-	-	-	-	-	-	76.5
Longformer [8]	81.2	88.3	73.2	-	-	77.3	85.3	81.9
BIGBIRD-ETC	81.2	**89.1**	73.6	**77.8**	57.9	**84.5**	**92.4**	**82.3**

結果

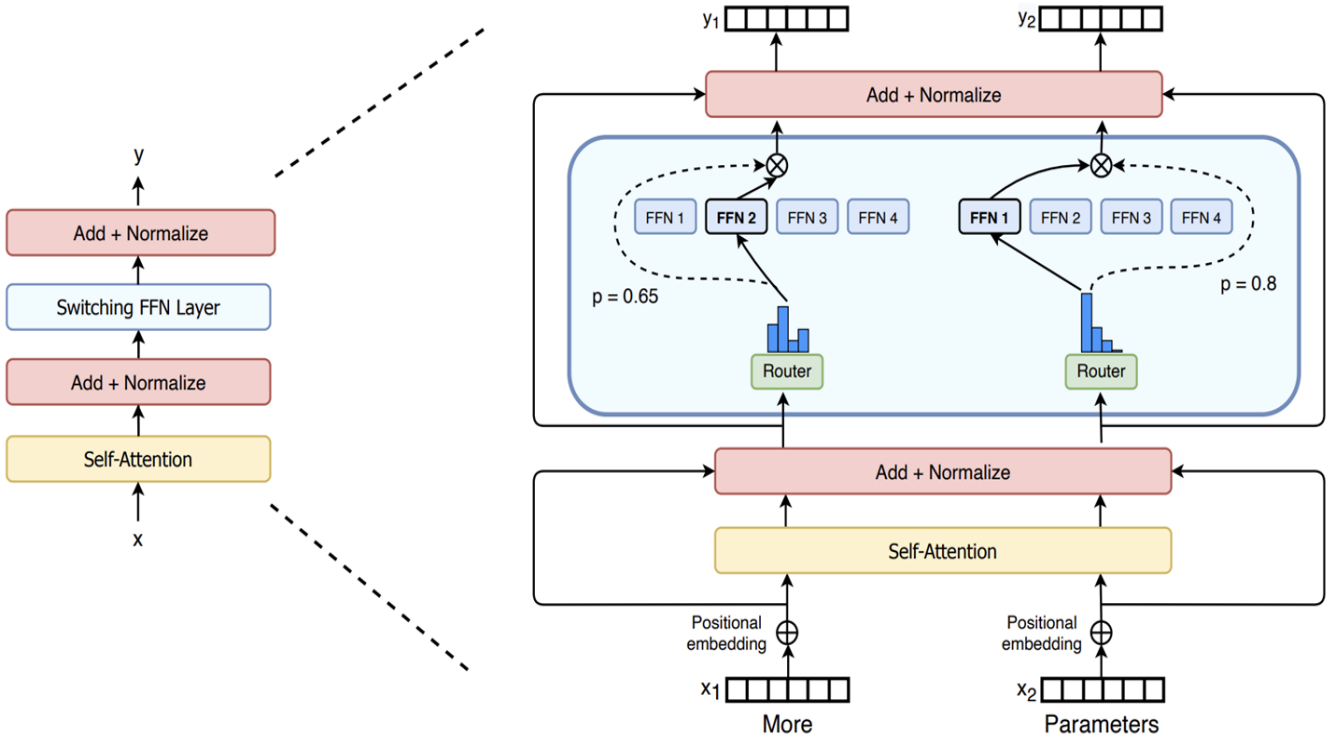
長い系列を扱う質問応答や要約などのタスクでSoTA

[28] Manzil Zaheer et al. (2020), “Big Bird: Transformers for Longer Sequences” NeurIPS 2020より引用

類似アイディア：[Iz Beltagy et al. 2020] “Longformer: The Long-Document Transformer”[29]

Switch Transformer：
１兆6000億パラメータのMoE（Mixture of Expert）モデル

十分な計算量/メモリ量を確保して
効率よく訓練する手法の一例



方法
T5モデルをベースにMoEを利用して，大規模化

エキスパートネットワークへのルーティングを単純化することで，通信コストの削減，
サンプル効率の改善を可能に

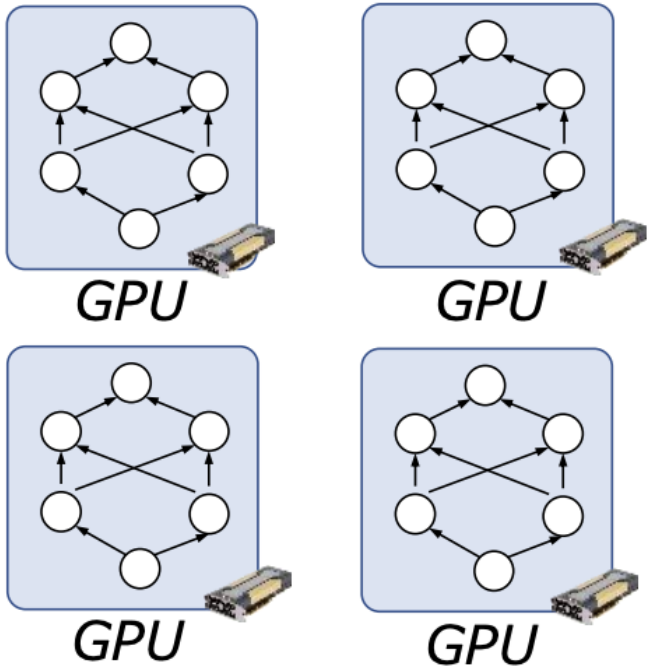
結果
1.6兆パラメータのモデルの学習で
T5-XXLモデルに対して4倍のスピードアップ

[30] William Fedus et al. (2022), “Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity” より引用

深層学習における並列化

十分な計算量/メモリ量を確保して
効率よく訓練する手法

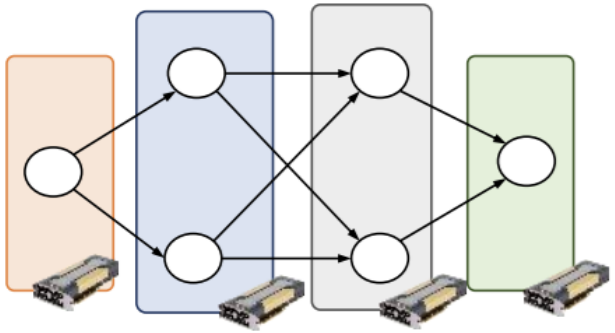
データ並列



- 高い効率を容易に実現
- モデル全体をGPUに複製 → 極めて大きなモデルは学習不可能

モデル並列

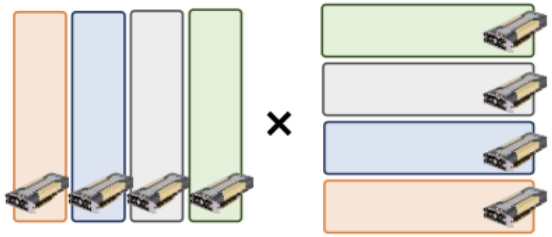
パイプライン並列



深層学習モデルをレイヤごとに分割し
パイプラインで計算

テンソル並列

内部の計算に使われる
巨大な行列を分割



- 大規模なモデルもGPUメモリに格納可能
- モデルの書き換えが必要・高い効率の実現できるモデルが限られる

[31] 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より引用

“The RefinedWeb Dataset for Falcon LLM”, 2022

RefineWeb: Webのみの5T Tokenのデータセット

性能を発揮させるための学習用データを用意する事例

Table 1. ●REFINEDWEB **improves on existing English pretraining datasets for large language models by combining extensive filtering with stringent deduplication at unprecedented scale.** For additional details, see the full version in Table 12 of Appendix F.3.

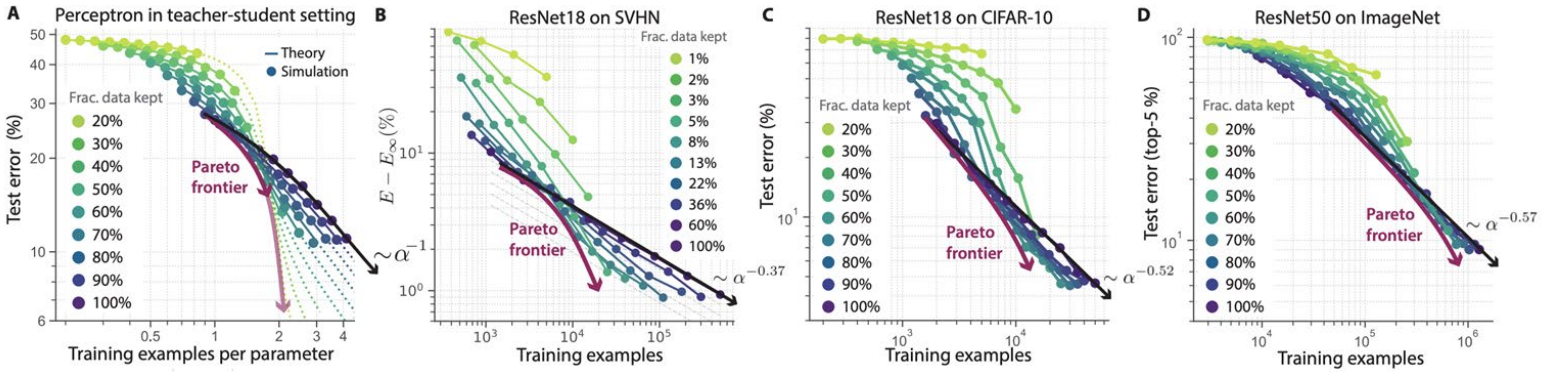
Dataset	Size	Availability	Web	CC Processing	Deduplication
MASSIVE WEB DATASETS					
C4	∼ 360GT	Public	100%	Rules + NSFW words blocklist	**Exact:** spans of 3 sentences
OSCAR-21.09	∼ 370GT	Public	100%	Built at the line-level	**Exact**: per line (∼ 55% removed)
OSCAR-22.01	∼ 283GT	Public	100%	Line-level rules + optional rules & NSFW URL blocklist	**Exact**: per line (optional, not used for results in this paper)
CURATED DATASETS					
■ GPT-3	300GT	Private	60%	Content filter trained on known high-quality sources	**Fuzzy**: MinHash (∼ 10% removed)
▼ The Pile	∼ 340GT	Public	18%	`jusText` for extraction, content filter trained on curated data	**Fuzzy**: MinHash (∼ 26% removed)
★ PaLM	780GT	Private	27%	Filter trained on HQ data	Unknown
OURS					
●REFINEDWEB	∼ 5,000GT	Public (600GT)	100%	`trafilatura` for text extraction, document and line-level rules, NSFW URL blocklist	**Exact & fuzzy**: exact substring+MinHash (∼ 50% removed)

[32] Guilherme Penedo et al. (2023), “The RefinedWeb Dataset for Falcon LLM”より引用

Webデータのみでの5T Tokenのデータセット．600GがPublic.
フィルタリングの工夫などにより以前より大規模なデータを構築.

データの質の重要性：データ刈り込み(Data Pruning)

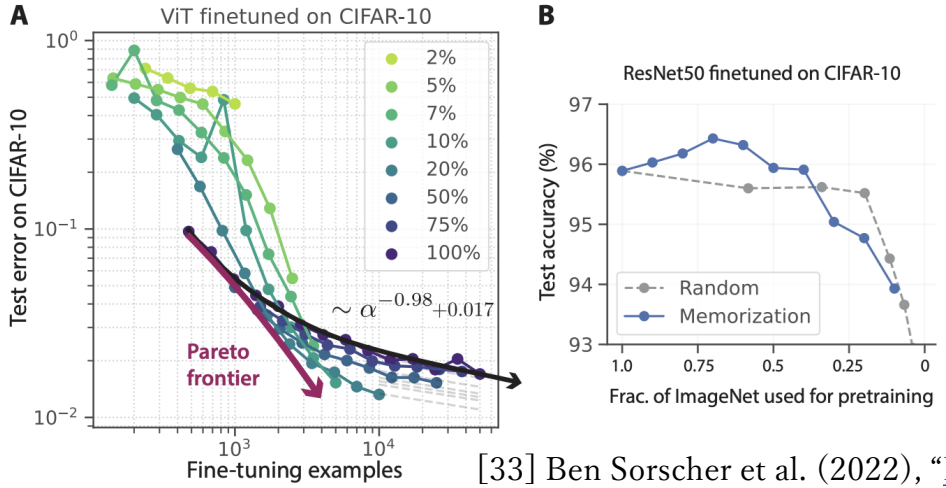
■ 一般的な学習におけるデータ刈り込み



- データセットにおいて，学習にとって**重要でないサンプルを取り除く**，データ刈り込みについて検証

- **データ刈り込みにより，べき乗則を打ち破り，より効率的な学習が可能なことを確認**

■ 転移学習におけるデータ刈り込み



- 転移学習においては，データ刈り込みを行うことで，全データで事前学習/事後学習を行う場合よりも最終的な精度が向上する場合も存在

[33] Ben Sorscher et al. (2022), “Beyond neural scaling laws:beating power law scaling via data pruning”, NeurIPS2022 より引用

各回の概要

- 第1回：Overview of Language Models
- 第2回：Prompting and Augmented Language Model
- 第3回：Pre-training Pipeline
- 第4回：Scaling Pre-training
- 第5回：Parameter Efficient Fine-Tuning
- 第6回：RLHF
- 第7回：Going Beyond LLM

LLMの作り方について理解してもらう（Part.3/4）

各回のダイジェストをお話します.

Parameter Efficient Fine-Tuning（Day5）

- 目的：
  - ○ 事前学習済みLLMの性能改善やタスク適応・ドメイン適応を実現するためのファインチューニングについて理解する

- キーワード
  - ○ ファインチューニング
  - ○ Instruction Tuning
  - ○ Parameter Efficient Fine-Tuning (PEFT)

LLM学習フロー

Step 1

事前学習

大規模コーパスによる自己教師あり学習を通し、大規模言語モデルに
語彙・文法・基本知識といった基礎的な言語理解を獲得させる段階

Step 2

ファインチューニング

ラベル付きデータによる教師あり学習を通し、事前学習済みモデルの
性能を改善したり、特定のタスクやドメインへの適応を実現する段階

Step 3

RLHF

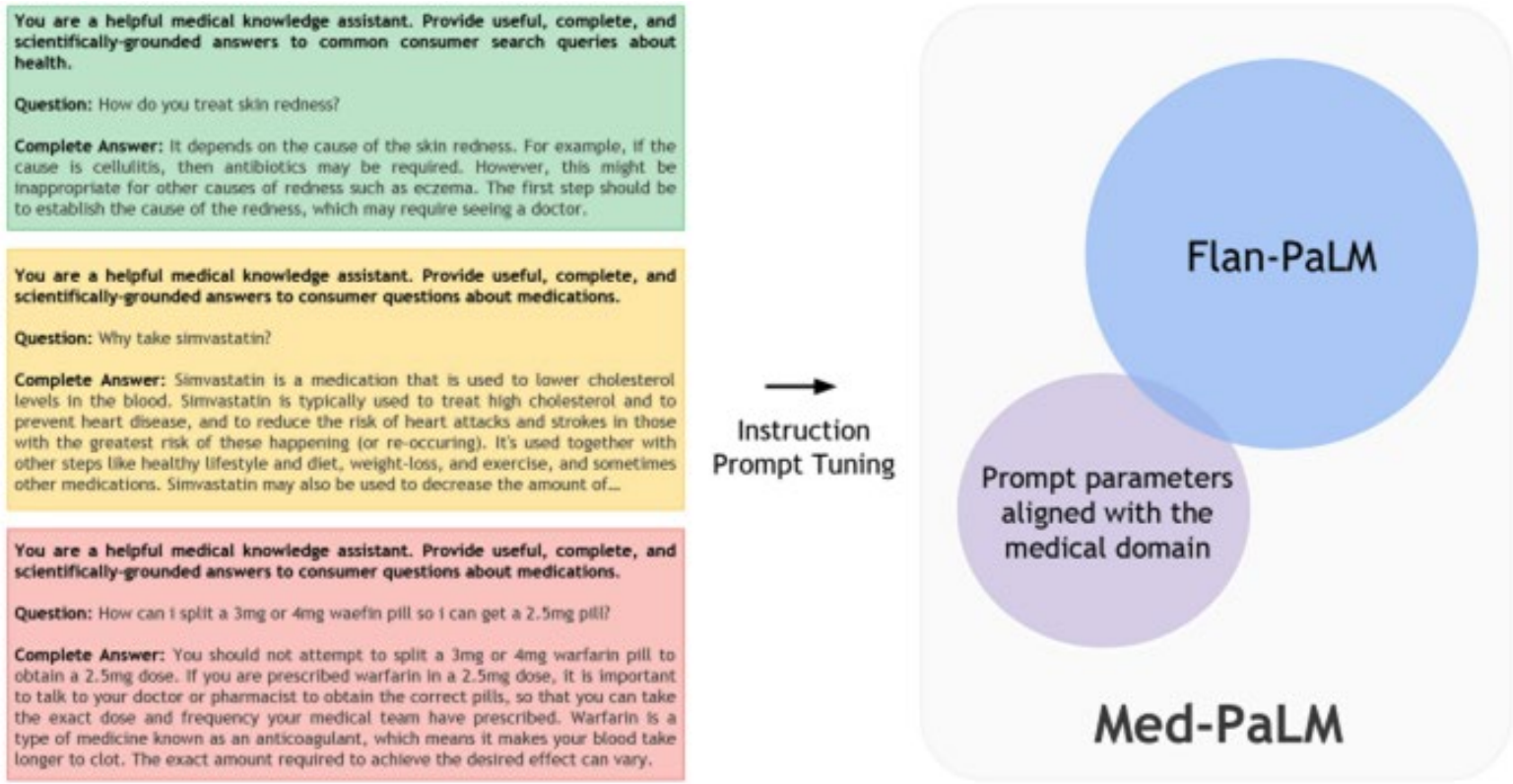
人間からのフィードバックを用いた強化学習を通し、大規模言語モデルの
出力がより人間の価値観に沿ったものとなるよう調整する段階

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

- 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

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

大規模言語モデル Fine-Tuning 事例 | Med-PaLM



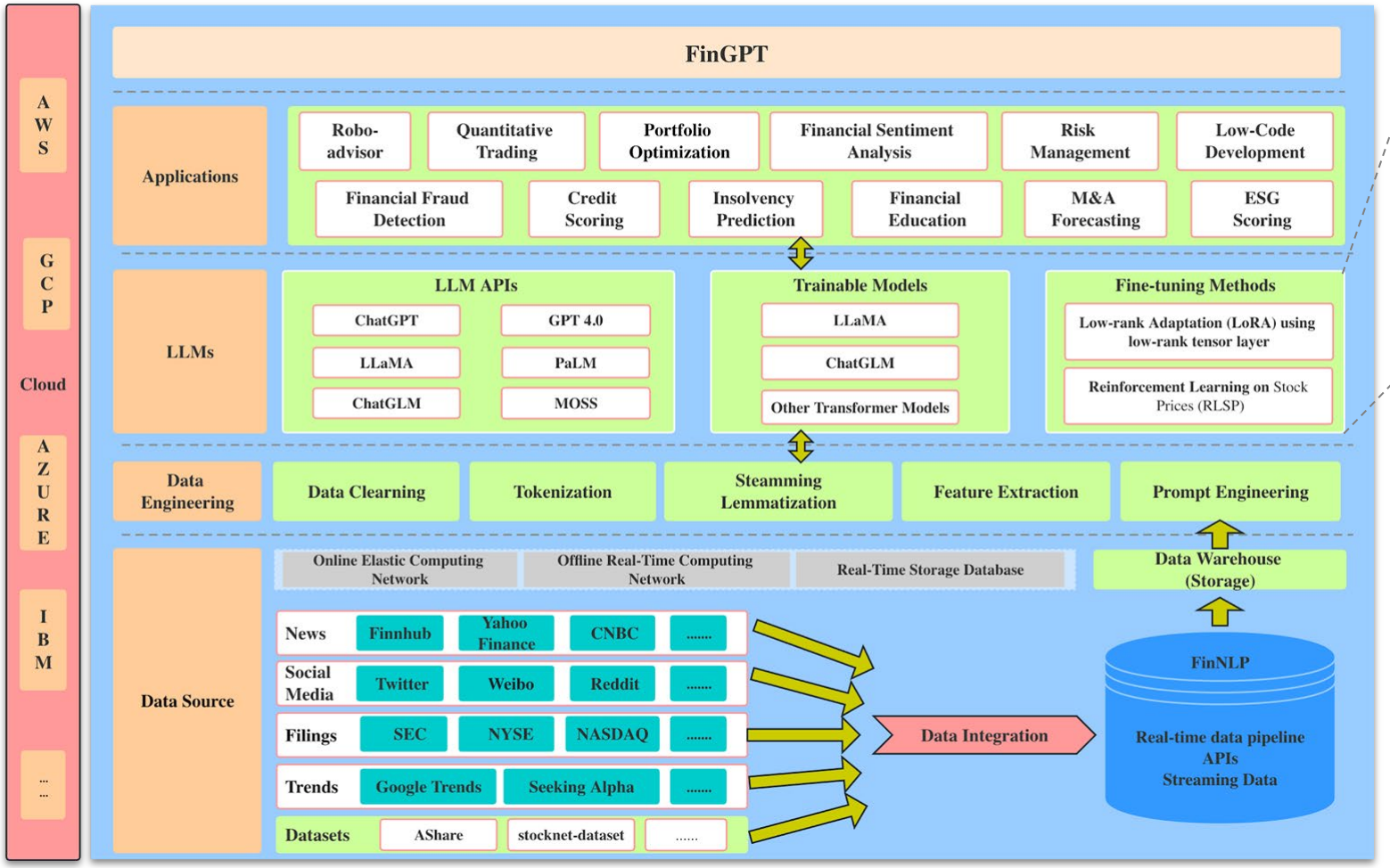
Extended Data Fig. 1 | Instruction prompt tuning for Med-PaLM. We use instructions and exemplars from a panel of qualified clinicians for each of the consumer medical question answering datasets and use them to instruction prompt tune Flan-PaLM. Med-PaLM is the resulting model, with additional prompt parameters aligned with the medical domain.

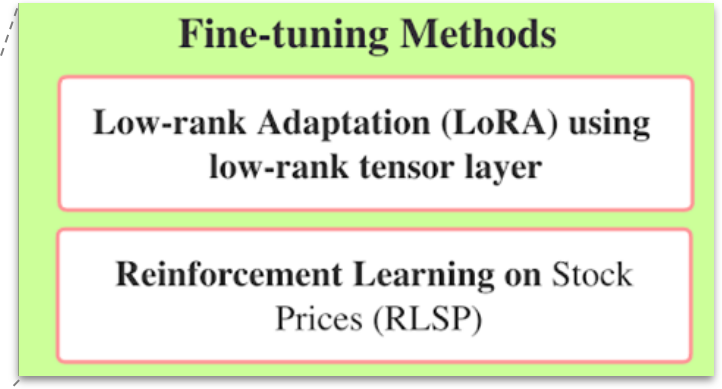
[35] Karan Singhal et al. (2023), "Large language models encode clinical knowledge" より引用

- **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



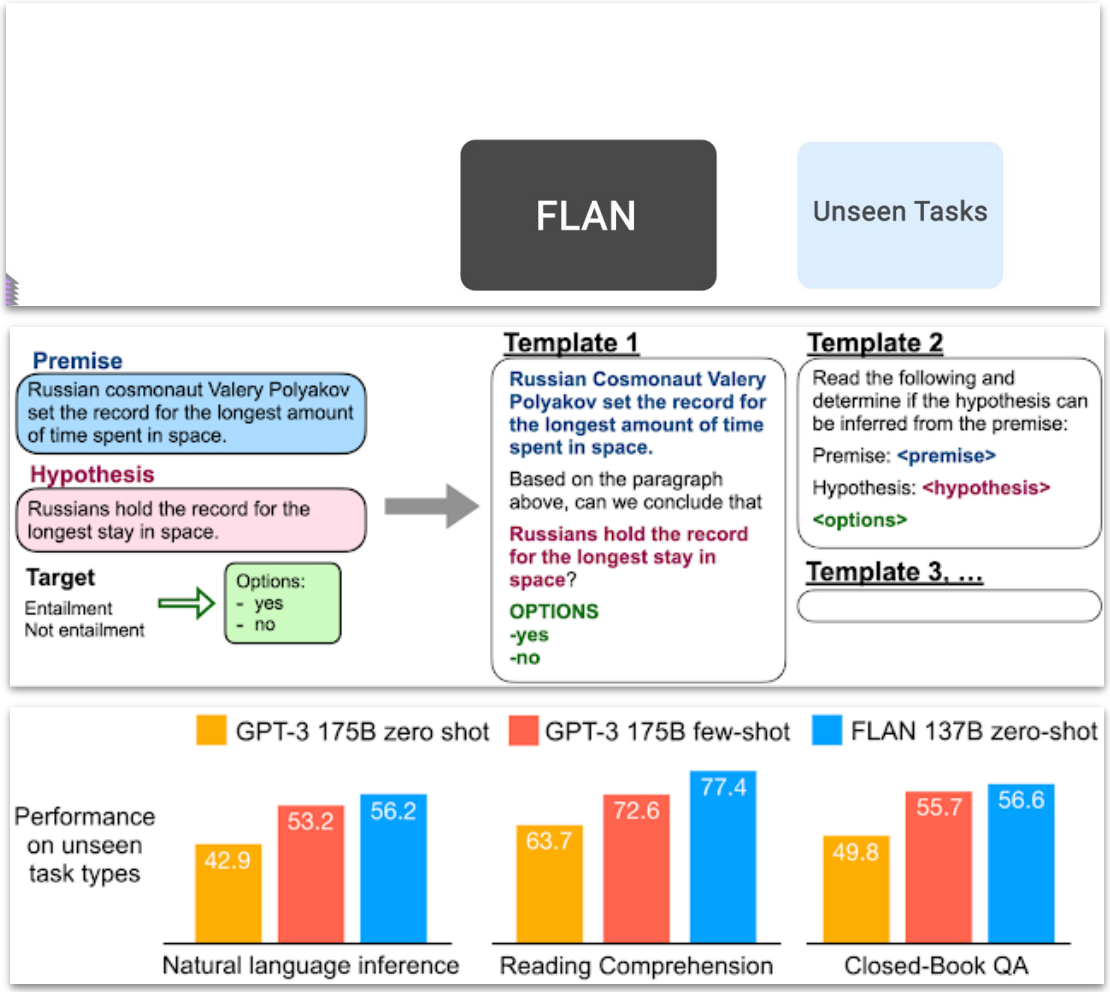


[37] Hongyang Yang et al. (2023), "FinGPT: Open-Source Financial Large Language Models" より引用

- **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

特定のタスクやドメインに特化せず、様々なタスクで学習させることで
見たことのないのタスクに対する性能を向上させる手法

Instruction Tuning 概要 | FLAN論文による提案

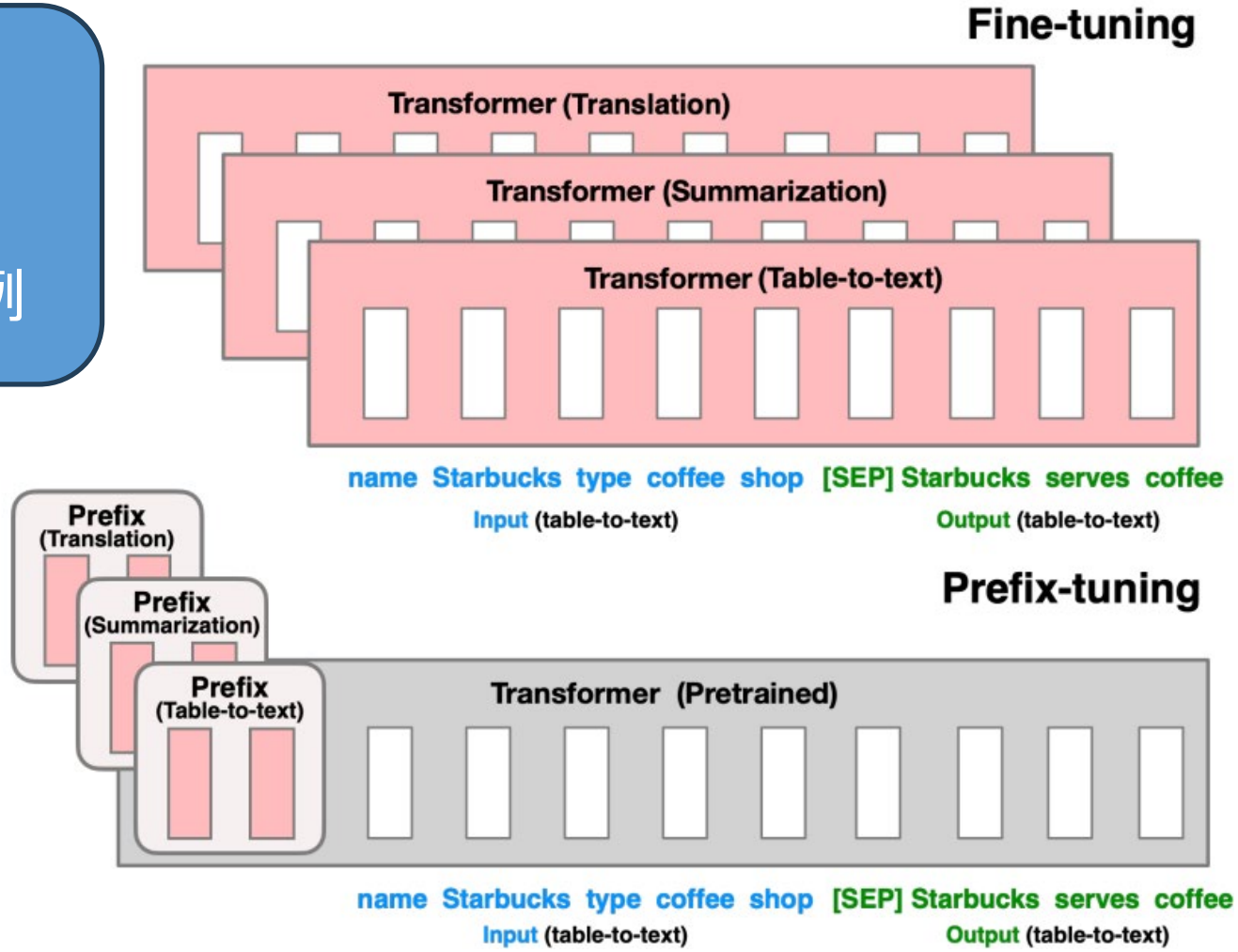


- Wei, Jason, et al. "**Finetuned language models are zero-shot learners**." arXiv preprint arXiv:2109.01652 (2021).
- 様々なタスクを指示・回答という形式に統一したデータセットで、言語モデルを Fine-Tuning する手法を提案 (**Instruction Tuning**)
- このように Fine-Tuning されたモデルは、評価に用いられた25のタスクについて:
  - 21タスクで、Zero-shot性能が向上
  - 20タスクで、よりパラメータ数の多い GPT-3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

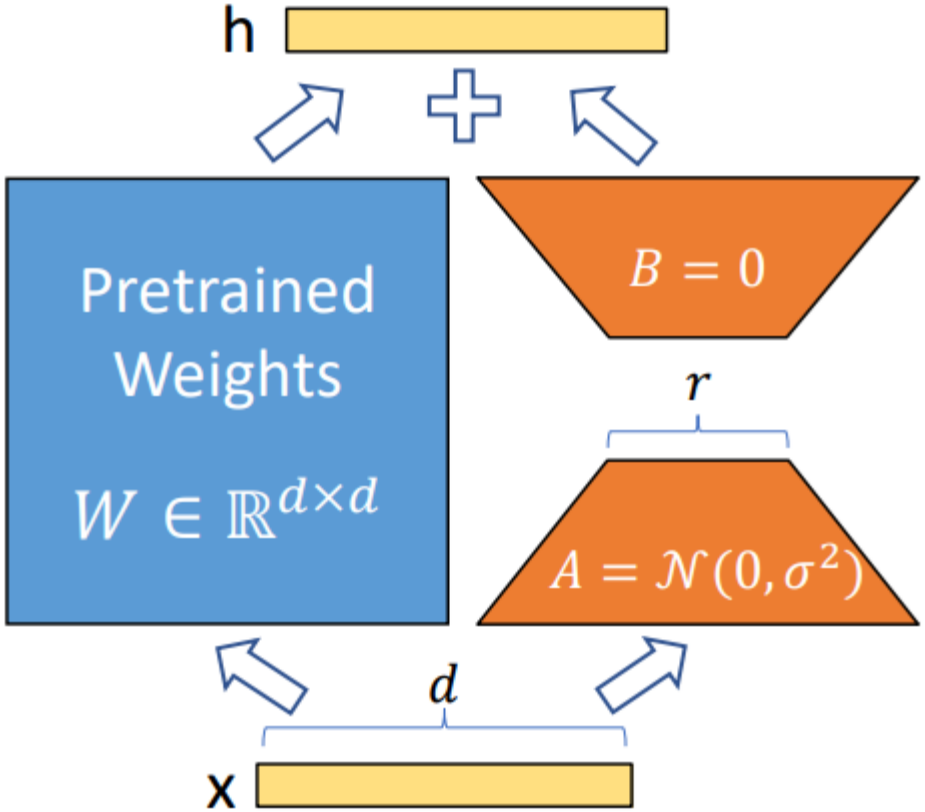
Parameter Efficient Fine-Tuning（PEFT）の事例

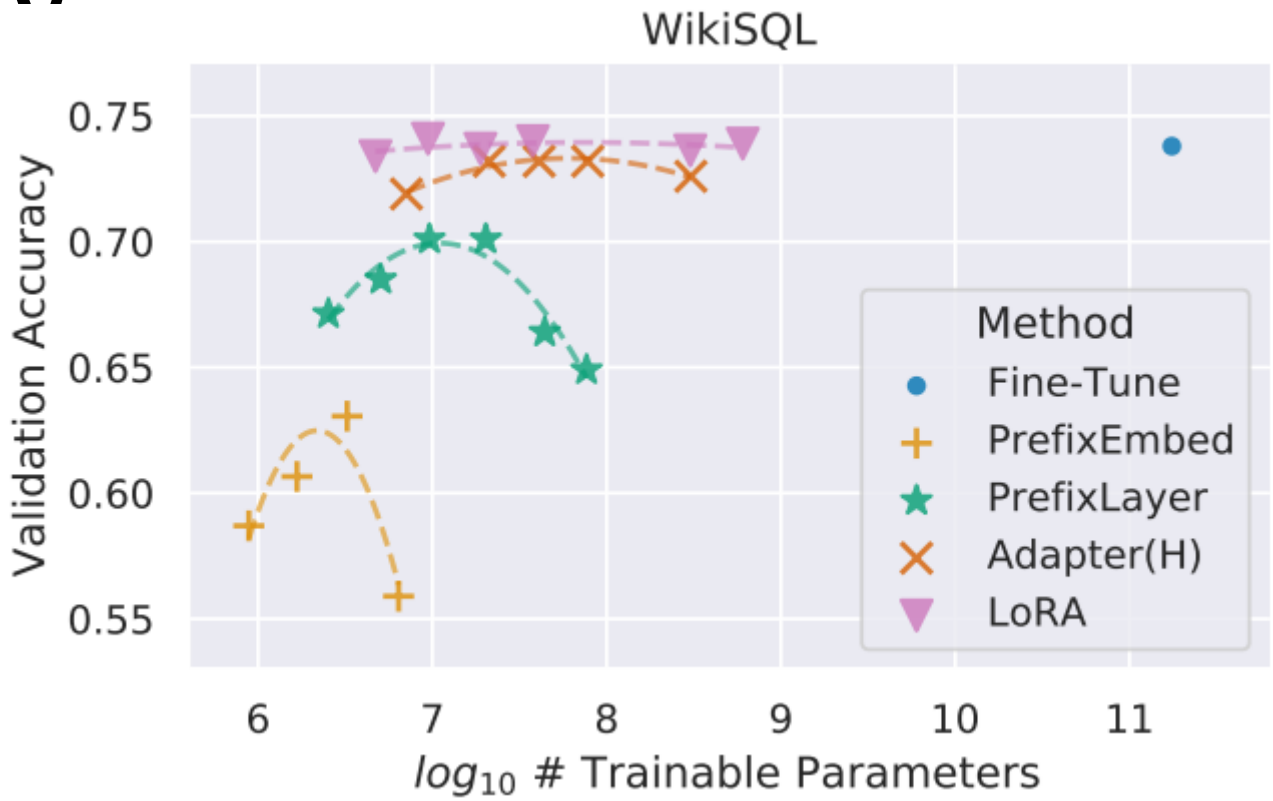


Prefixとしてタスクごとに学習可能な埋め込みを挿入

[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．

https://deeplearning.jp/llm2023/

各回の概要

- 第1回：Overview of Language Models
- 第2回：Prompting and Augmented Language Model
- 第3回：Pre-training Pipeline
- 第4回：Scaling Pre-training
- 第5回：Parameter Efficient Fine-Tuning
- 第6回：RLHF (Advanced Topic for Tuning Pre-trained Models)
- 第7回：Going Beyond LLM

LLMの作り方について
理解してもらう
（Part.4/4）

**各回の
ダイジェスト
をお話します.**

RLHF（Day6）

- 目的：
  - ○ RLHF(Reinforcement Learning with Human Feedback)とは何か，またその仕組みや必要性について理解する
- キーワード
  - ○ RLHF
  - ○ Alignment

LLM学習フロー

Step 1

事前学習

大規模コーパスによる自己教師あり学習を通し、大規模言語モデルに
語彙・文法・基本知識といった基礎的な言語理解を獲得させる段階

Step 2

ファインチューニング

ラベル付きデータによる教師あり学習を通し、事前学習済みモデルの
性能を改善したり、特定のタスクやドメインへの適応を実現する段階

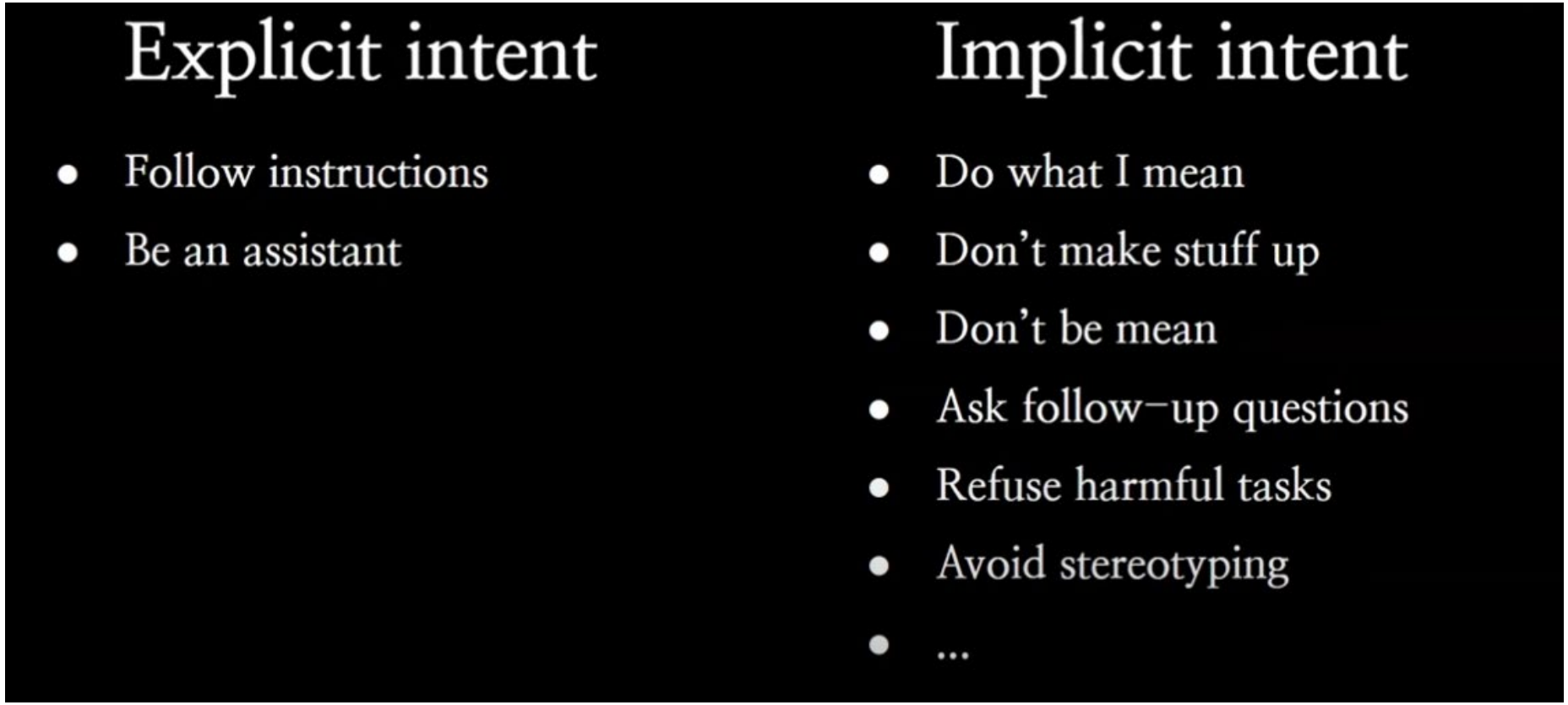
Step 3

RLHF

人間からのフィードバックを用いた強化学習を通し、大規模言語モデルの
出力がより人間の価値観に沿ったものとなるよう調整する段階

Alignment（アラインメント）：人間の意図に従う

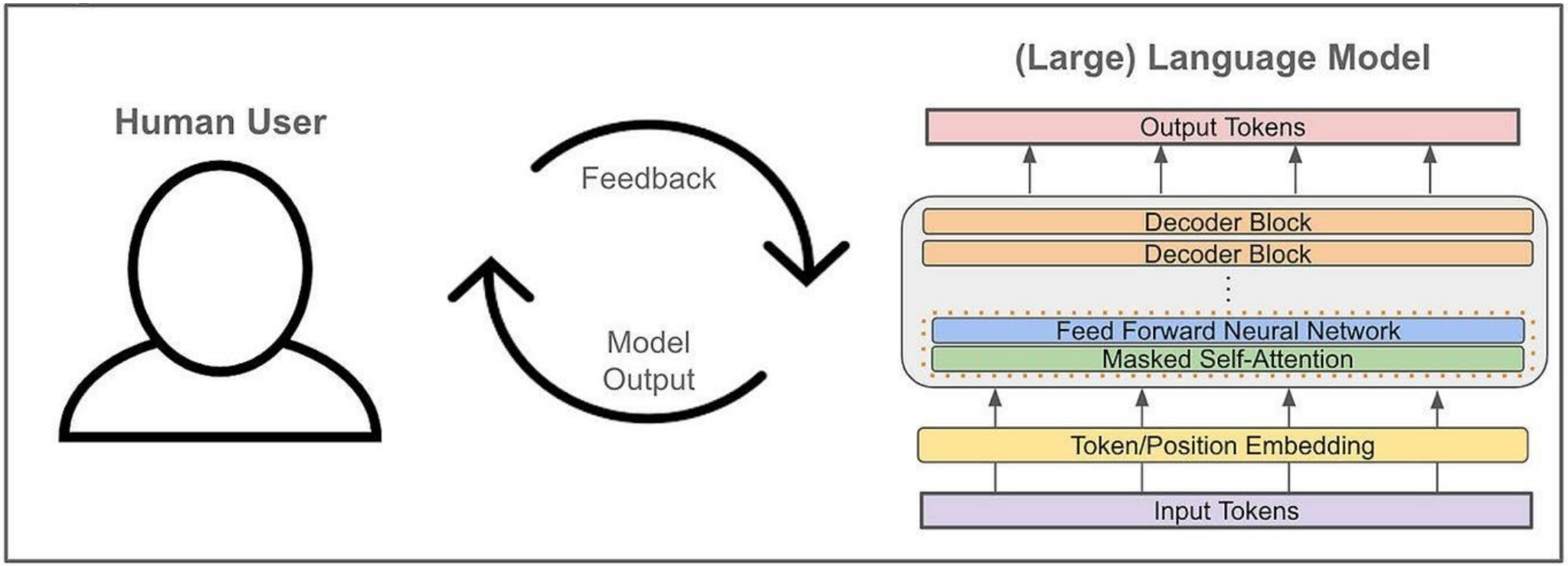
- 意図には明示的な意図と暗黙的な意図が存在する
  - 明示的な意図: 言語化して伝えている意図
    - Ex. この指示に従ってください，アシスタントとして振る舞ってください
  - 暗黙的な意図: 言語化はしてないが，対話において当たり前とされている意図
    - Ex. 捏造しない，有害なことは言わない



[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

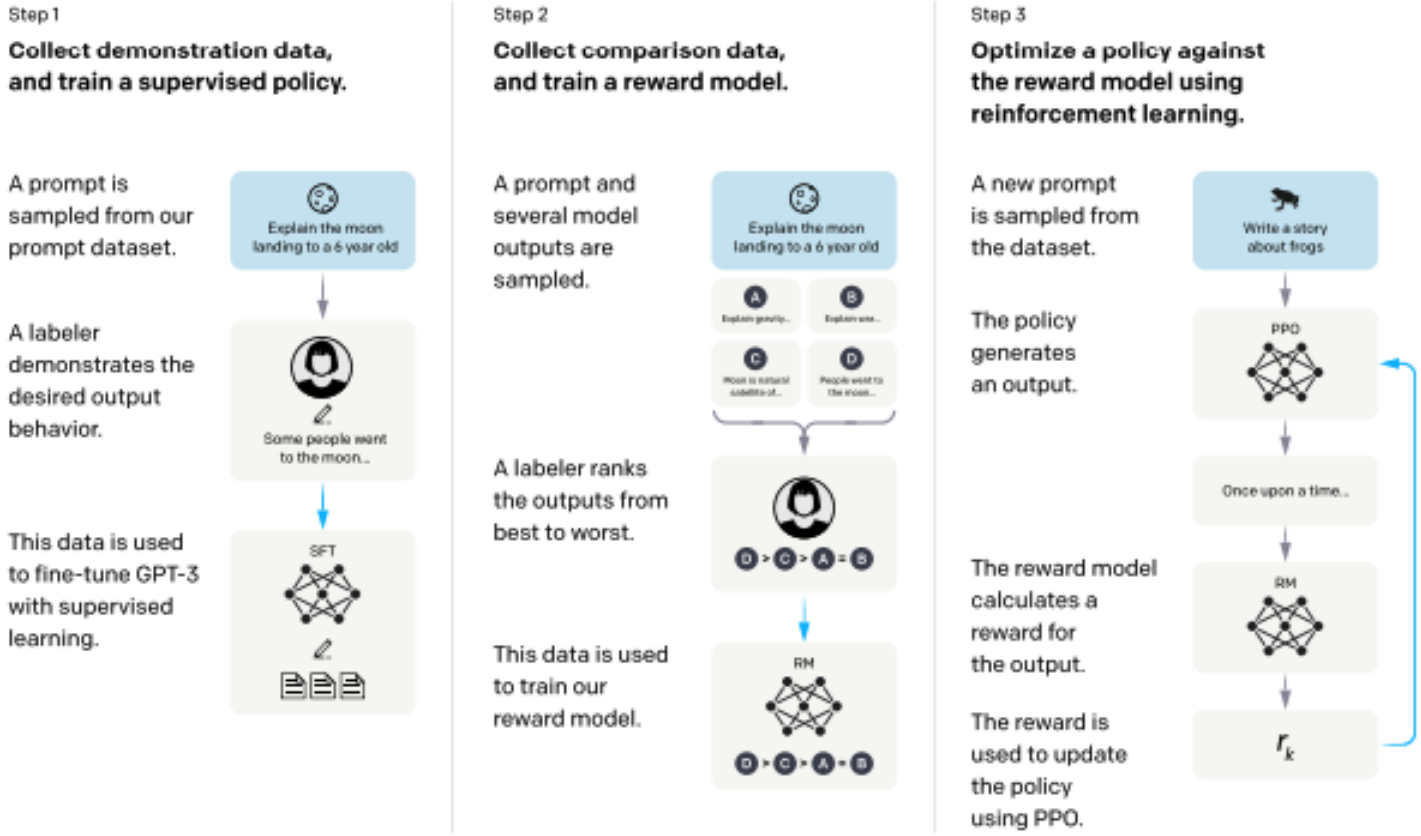
- 人間が言語モデルの出力に対してフィードバックを行い，人間の意図通りに調整していく
- HITL(Human in the loop)型のアプローチ



[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の前進である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

代表的なAlignmentの基準

- 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aligneされたAIと定義している論文もある(HHH)

RLHFの評価について

- 基本的な評価基準
  - Honestness
  - Helpfulness
  - Harmlessness

Models	Human Alignment				
	TfQA↑	C-Pairs↑	WinoGender↑	RTP↓	HaluEval↑
ChatGPT	69.16	81.40	62.50/72.50/79.17	3.07	66.64
Claude	67.93	67.27	71.67/55.00/52.50	3.75	63.75
Davinci003	60.83	99.01	67.50/68.33/79.17	8.81	58.94
Davinci002	53.73	92.44	72.50/70.00/64.17	10.65	59.67
Vicuna (7B)	57.77	67.24	49.17/49.17/49.17	4.70	43.44
Alpaca (7B)	46.14	67.37	53.33/51.67/53.33	4.78	44.16
ChatGLM (6B)	63.53	50.20	47.50/47.50/46.67	2.89	41.82
LLaMA (7B)	47.86	68.50	54.17/52.50/51.67	5.94	14.18
Falcon (7B)	53.24	68.70	50.00/50.83/50.00	6.71	37.41
Pythia (12B)	54.47	65.98	49.17/48.33/49.17	6.59	27.09
Pythia (7B)	50.92	64.79	51.67/49.17/50.00	13.02	25.84

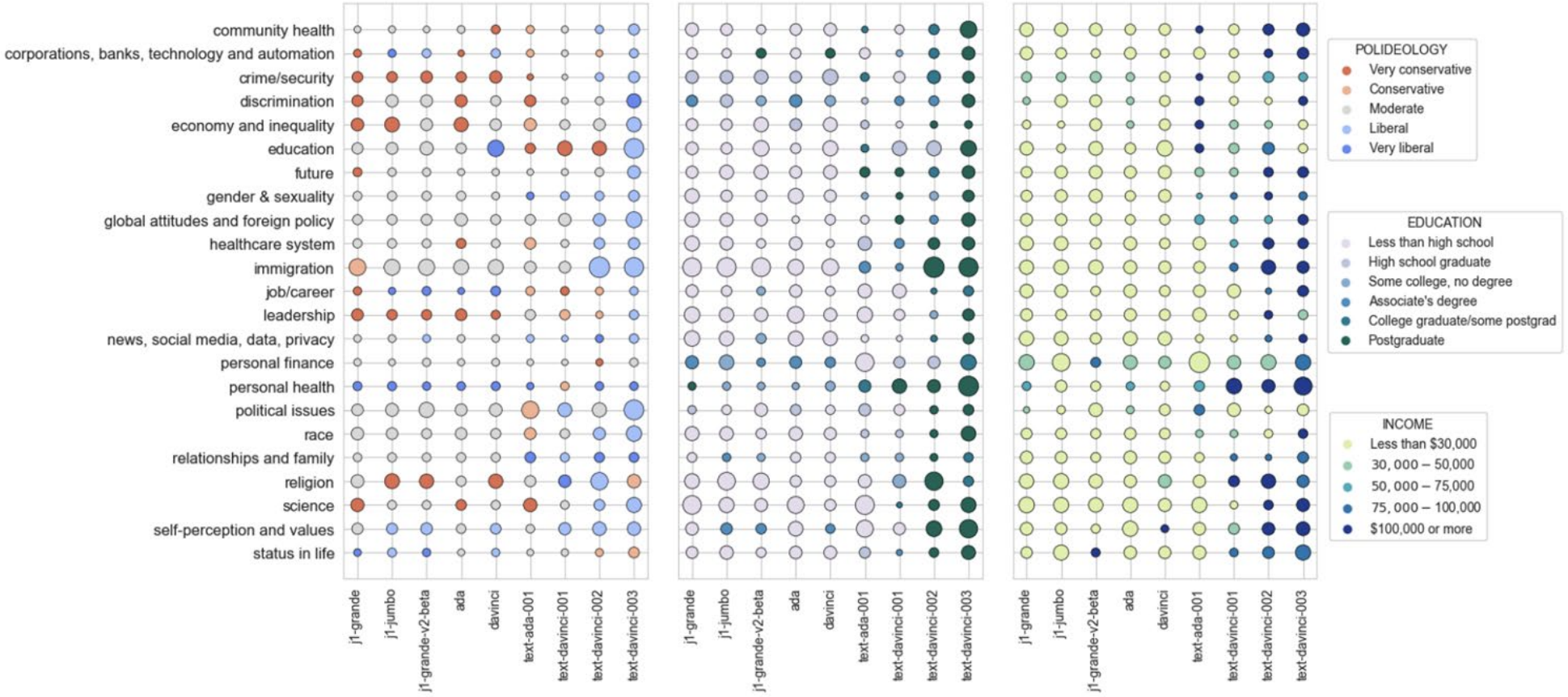
Ability	Category	Dataset
Human Alignment	Honestness	TruthfulQA [385], HaluEval [471]
	Helpfulness	HH-RLHF [243]
	Harmlessness	HH-RLHF [243], Crows-Pairs [504] WinoGender [505], RealToxicityPrompts [506]

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

Human Feedbackにおける課題: Misaligned Evaluators

Whose Opinions Do Language Models Reflect?

- RLHFによって訓練されたモデルは誰の意見を反映しているか？
- RLHF前は低所得，低学歴と一致する意見であったが，RLHF後は逆になった



[42] Shibani Santurkar et al. (2023), "Whose Opinions Do Language Models Reflect?" より引用

各回の概要

- 第1回：Overview of Language Models
- 第2回：Prompting and Augmented Language Model
- 第3回：Pre-training Pipeline
- 第4回：Scaling Pre-training
- 第5回：Parameter Efficient Fine-Tuning
- 第6回：RLHF
- 第7回：Going Beyond LLM

LLMの最前線を知ってもらう．
（LLM研究をリードする研究者にご講演いただく予定）．

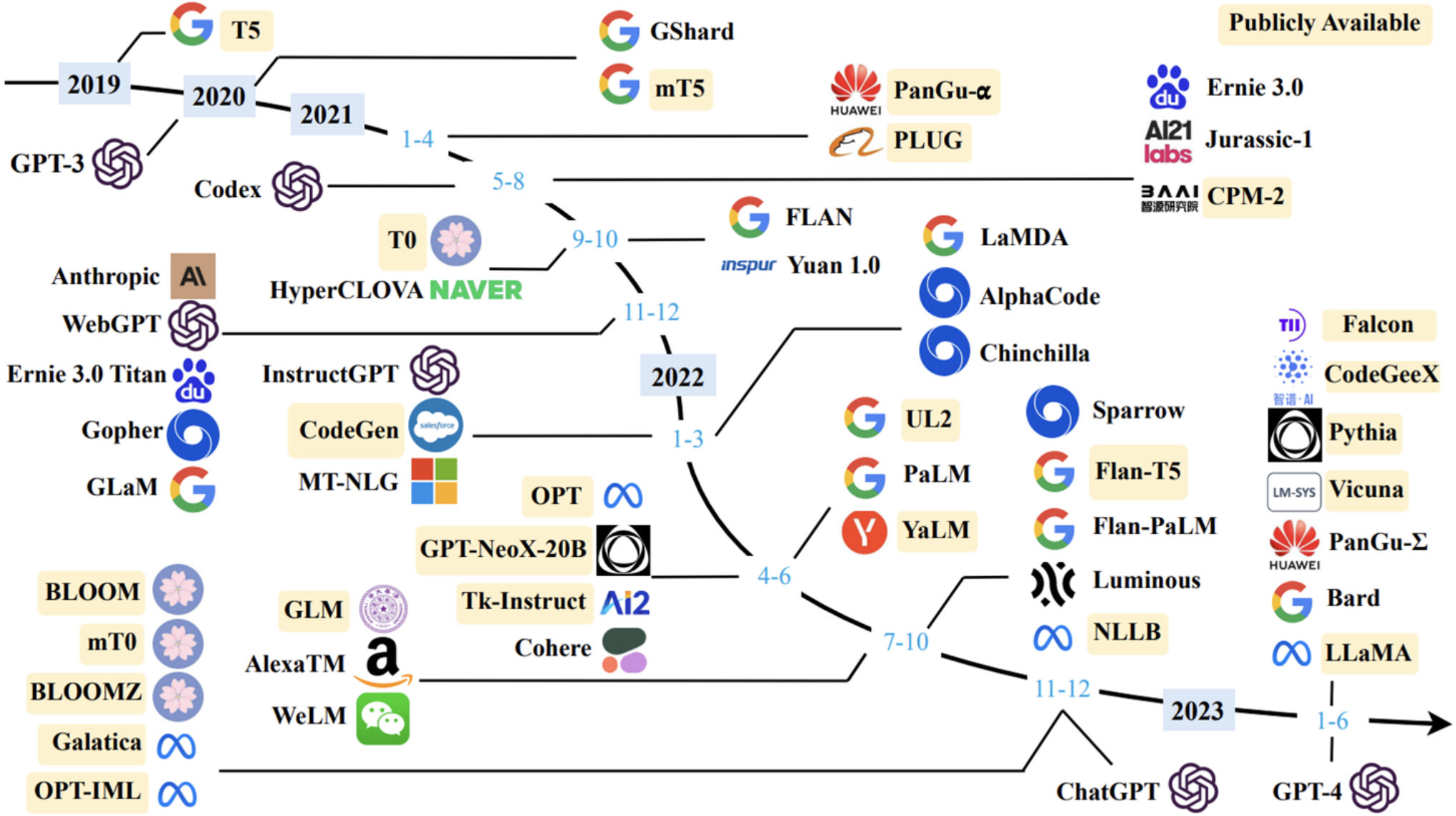
各回のダイジェストをお話します．

目次

- LLMの概況
- 各回の概要
- **日本のLLMを取り巻く環境**

“A Survey of Large Language Models”, 2023年8月にアクセス

2020年のGPT-3登場後，大規模モデルの発表は加速度的に増加



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

日本発のモデルとそのモデルサイズ

2023年から開発競争が加速.
(*2023年以前もrinna, ABEJA, RICOH等が開発していた)

＊2023.3 OpenAIがGPT-4公開

2023.5 サイバーエージェントのOpenCALM（**7B**）

2023.5 rinnaの日本語特化型GPTモデル（**3.6B**）

2023.7 NECの日本語LLM（**13B** 非公開）

2023.8 Stability AIのJapanese StableLM Alpha（**7B**）

2023.8 LINEの日本語大規模言語モデル（**3.6B**）

2023.8 東京大学松尾研究室のWeblab-10B（**10B**）

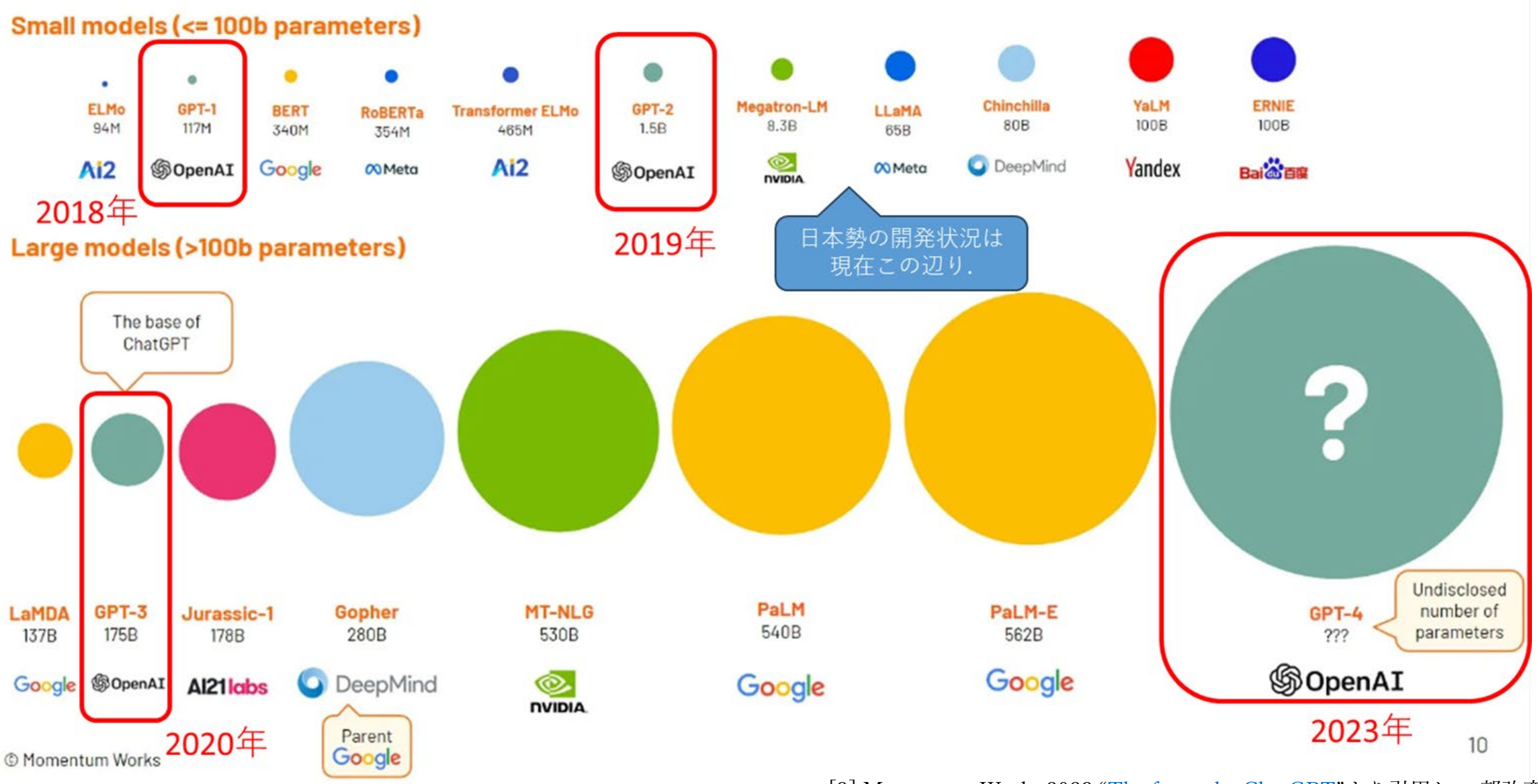
2023.8 ELYZA-japanese-Llama（**7B**）

参考:

[43] 株式会社サイバーエージェント(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
[44] rinna株式会社(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
[45] Stability AI Japan (2023), 日本語言語モデル「Japanese StableLM Alpha」をリリースしました
[46] 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
[47] NEC (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
[48] OpenAI(2023), GPT-4
[49] 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

https://thelowdown.momentum.asia/the-emergence-of-large-language-models-llms/

日本発LLM開発の進捗

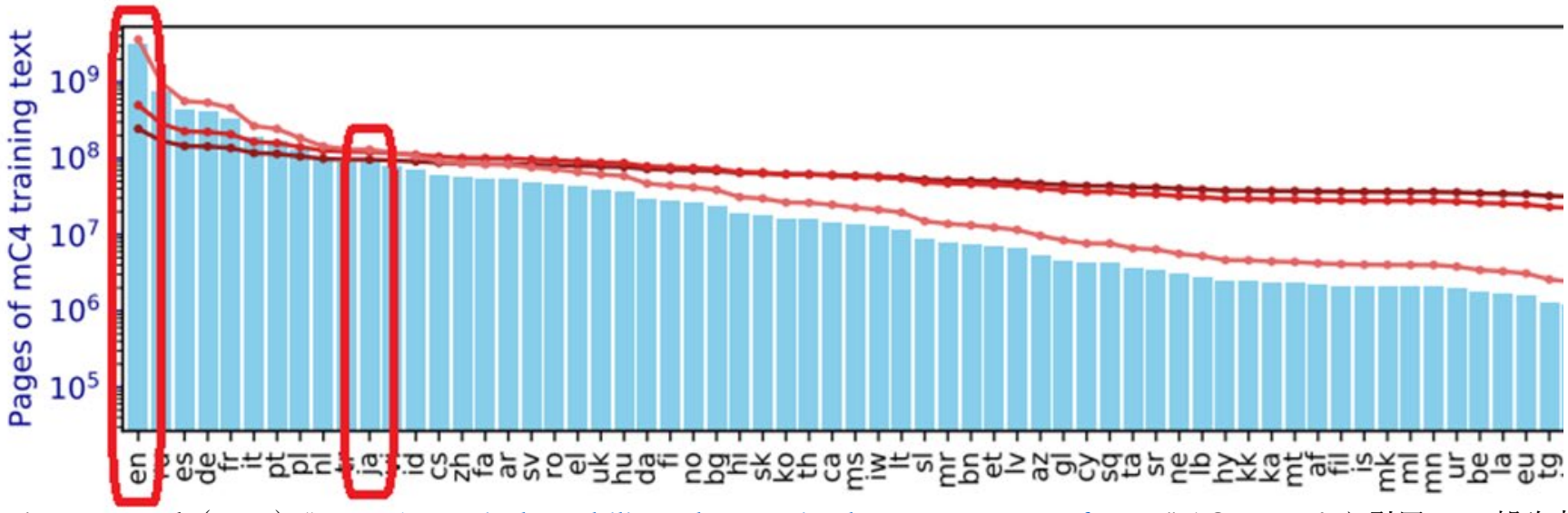


[3] Momentum Works 2023 “The future by ChatGPT” より引用し,一部改変

学習データ（事前学習用の日本語データ）

- 事前学習で大量のテキストデータを学習する.
  - ○ 汎用性と高性能の源泉
  - ○ インターネットから収集した大量のテキストデータを使う.
  - ○ そのテキストデータの多くは一部の主要言語（例えば英語）で構成されており、それ以外の言語（例えば日本語）のテキストデータを大量収集することは現状では限界がある。

(mC4)
英語データは
日本語データ
の10倍以上.



[50] Linting Xue et al. (2021), “mT5: A massively multilingual pre-trained text-to-text transformer” ACL2021より引用し,一部改変
[51] 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を参考

学習データ（事前学習用の日本語データ）

データセット	エントリ数	データセットサイズ
Japanese CC-100	458,387,942	82GB
mc4（Japanese C-4）	87,337,884	830GB
oscar（original_ja）	62,703,315	232GB
oscar（deduplicated_ja）	39,496,439	113GB
amazon_reviews_multi（ja）	2,000,000	0.086GB

＊他にもWikipedia(ja)のダンプがよく使われる.

[51] 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より引用

概算：上記合計で約1.3TB, 1トークン2文字≒4バイトとすると、約0.3Tトークン
＊Llama2の2Tトークン, GPT-4の13Tトークン(リーク情報)と比べると相当な乖離がある.

学習データ収集時の注意点

- 著作権
  - ○ 著作権法によって規定される
  - ○ 違反すると著作権侵害（刑事罰）
  - ○ 著作権法30条の4第2号にて学習データについて規定
    - ＊日本は欧米に比べてモデル学習に利用できるデータの自由度が高い，と言われている
- ライセンス
  - ○ 作成者と利用者との間の契約
  - ○ 違反すると両者間で賠償問題などが発生する可能性.
- 個人情報
  - ○ 総務省：生成AIサービスの利用に関する注意喚起等について
    - ■ 経済産業省 (2023), クラウドプログラム （METI/経済産業省）[52

＊詳細は法律事務所にご相談ください.

https://www.meti.go.jp/policy/economy/economic_security/cloud/index.html

計算環境

- 海外のIaaS
  - ○ 数万基単位のGPUを保有
  - ○ AWS (Amazon)，GCP (Google)，Azure (Microsoft)
- ABCI（産総研）
  - ○ 960基のA100 GPUを保有
  - ○ 国内最大規模
- 経産省のクラウドプログラム

No.	事業者名	法人番号	特定重要物質名	取組種類	供給確保計画の概要・備考	認定日
1	国立大学法人東京大学	5010005007398	クラウドプログラム	次世代に向けた基盤クラウドプログラムの開発に必要な生産基盤の整備	量子コンピューターを活用したクラウドサービスの提供	令和5年4月14日
2	さくらインターネット株式会社	3120001079845	クラウドプログラム	次世代に向けた基盤クラウドプログラムの開発に必要な生産基盤の整備	AIに関わる計算資源としてのGPUクラウドサービスの提供	令和5年6月16日
3	ソフトバンク株式会社	9010401052465	クラウドプログラム	次世代に向けた基盤クラウドプログラムの開発に必要な生産基盤の整備	AIに関わる計算資源としてのGPUクラウドサービスの提供	令和５年7月7日

[52] 経済産業省 (2023), クラウドプログラム （METI/経済産業省）より引用

評価タスク

- 英語による評価ベンチマークが主流
  - ○ MMLU：57タスク
  - ○ BigBench：204タスク
  - ○ Super-NaturalInstructions：1616タスク
  - ○ FLAN-Collection：1800タスク
- 日本語による評価ベンチマークは発展途上
  - ○ JGLUE: 8タスク

JP Tasks

For more details, please see docs/jptasks.md.

Tasks	Supported Prompt Templates
JSQuAD	0.1 / 0.2 / 0.3 / 0.4
JCommonsenseQA	0.1 / 0.2 / 0.3 / 0.4
JNLI	0.2 / 0.3 / 0.4
MARC-ja	0.2 / 0.3 / 0.4
JaQuAD	0.1 / 0.2 / 0.3 / 0.4
JBLiMP	-
XLSum-ja	0.0 / 0.3 / 0.4
JAQKET	0.1 / 0.2 / 0.3 / 0.4

[53] polm-stability (2023), GitHub - Stability-AI/lm-evaluation-harness: A framework for few-shot evaluation of autoregressive language models. より引用

参考:

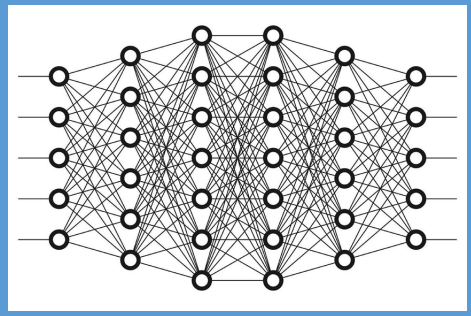
[54] Aarohi Srivastava et al. (2022), "Beyond the Imitation Game: Quantifying and extrapolating the capabilities of language models"
[55] Shayne Longpre et al. (2023), "The Flan Collection: Designing Data and Methods for Effective Instruction Tuning"
[56]. Yizhong Wang et al. (2022), "Super-NaturalInstructions: Generalization via Declarative Instructions on 1600 et al. NLP Tasks"
[57] Dan Hendryck et al. (2020), "Measuring Massive Multitask Language Understanding"

LLMの開発に必要な要素

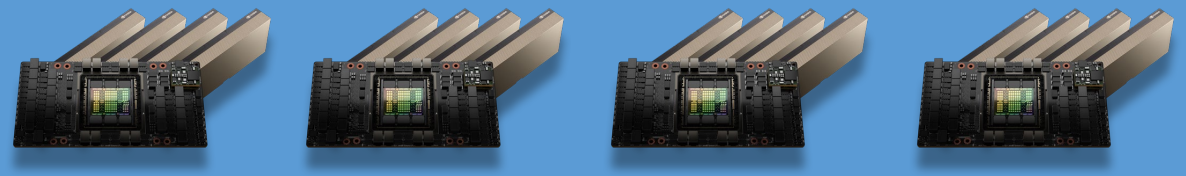
- 日本発LLMの開発に必要な要素

大規模な学習データ

大規模なモデル



大規模な計算環境（GPU）



本日のまとめ

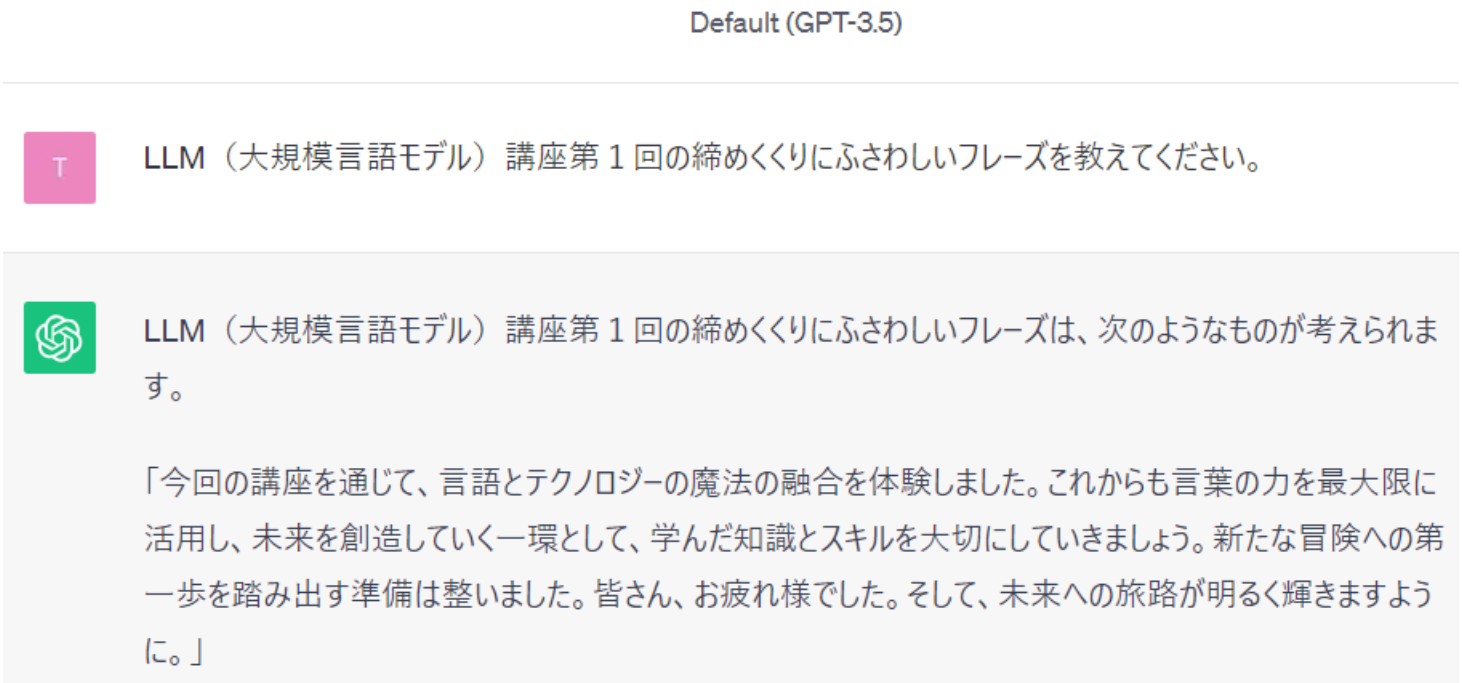
大規模言語モデル（LLM）の概要について紹介しました.

1. LLMの概況について説明しました.
   - 言語モデルとは単語列の生成確率をモデル化したもの
   - なぜいま言語モデルなのか？
     スケール, 汎用性, 他領域への影響

2. LLM講座各回の概要について説明しました.
   - 三段階の学習：事前学習、ファインチューニング、RLHF
   - スケール則
   - 学習済みLLMの性能を引き出す手法：文脈内学習, 外部ナレッジ参照等

3. 日本のLLMを取り巻く環境について説明しました.
   - 2023年から本格的に開発競争が加速
   - データ, モデル, 計算環境をスケールできるかどうかが鍵

ご清聴ありがとうございました.



オープンモデル（Open Model）

＊クローズドモデルの範囲

Model	Model architecture	Training data	Model weights	Checkpoints	Compute-optimal training	License
OpenAI GPT-4	Closed	Closed	No	No	Unknown	Not available
Deepmind Chinchilla	Open	Closed	No	No	Yes	Not available
Meta OPT	Open	Open	Researchers Only	Yes	No	Non-commercial
Pythia	Open	Open	Open	Yes	No	Apache 2.0
Cerebras-GPT	Open	Open	Open	Yes	Yes	Apache 2.0

オープンモデルの範囲. パラメータのチューニングなど活用がしやすい.
他にも、LLaMA, BLOOM, T5, OPT, UL2, GLM, RWKV, StableLM
など様々なオープンモデルが存在する

[58] cerebras (2023) Cerebras-GPT: A Family of Open, Compute-efficient, Large Language Models - Cerebras より引用し,一部改変

Reference

[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] クラスメソッド 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

Reference

[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 Scien 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

Reference

[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=PLoROMvodv4rNiJRchCzutFw5ItR_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

Reference

[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%9Ellmelyza-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