



大規模言語モデル入門

NTT人間情報研究所

西田京介, 西田光甫, 風戸広史

2023年8月24日 ソフトウェアエンジニアリングシンポジウム2023



西田光甫

専門：自然言語理解・生成

NTT人間情報研究所 研究員.

1. LLMに関する基礎知識

- ニューラルネットを用いた自然言語処理
- Transformer, BERT, GPT-3



西田京介

専門：機械読解（質問応答）, Vision-and-Language

NTT人間情報研究所
上席特別研究員.

2. 対話に最適化されたLLMおよびソフトウェアエンジニアリングへの応用

- ChatGPT, GPT-4, LLaMA
- Github co-pilot, コードに最適されたLLM

協力者紹介



風戸 広史

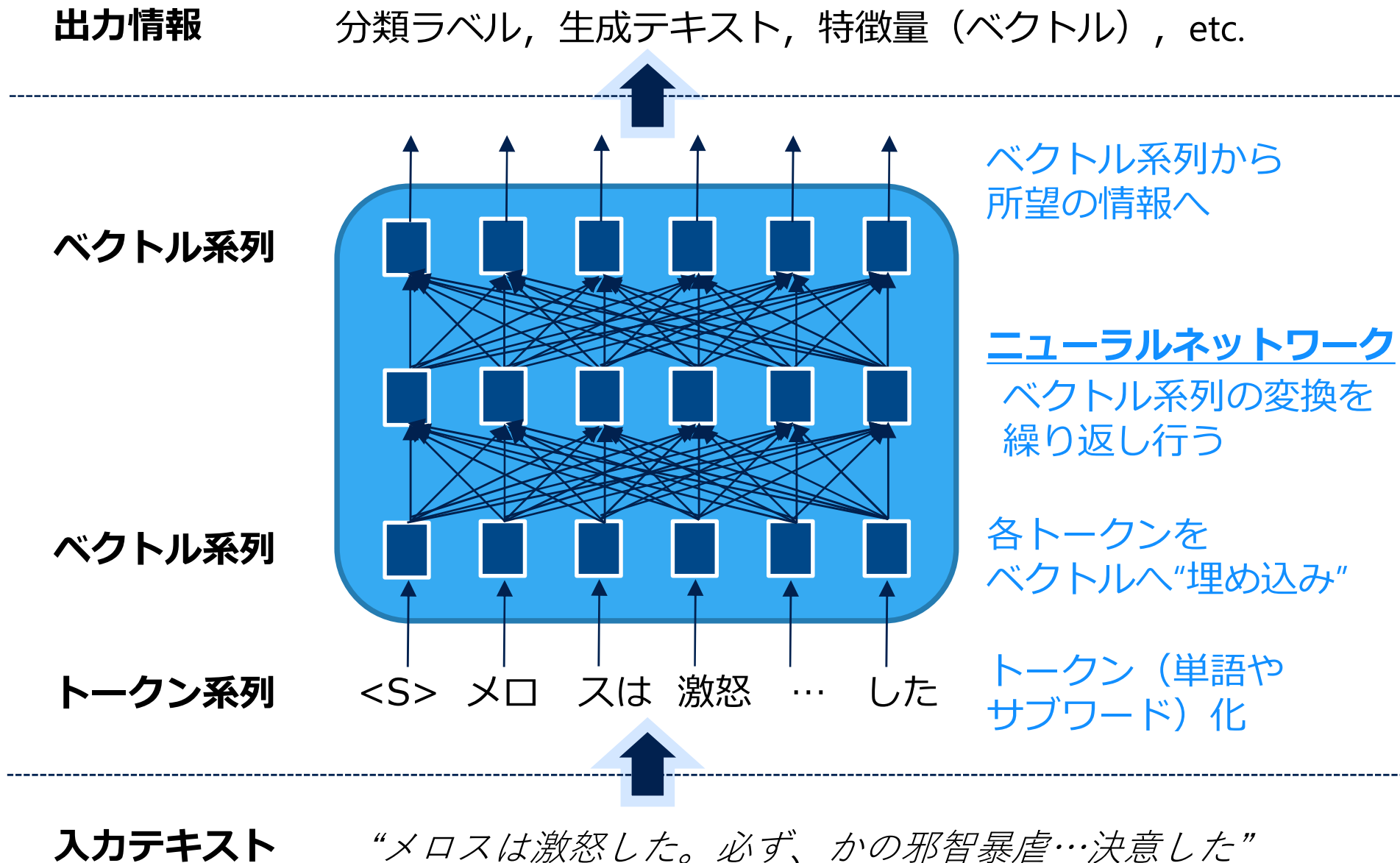
専門：ソフトウェア工学

NTT人間情報研究所
主任研究員。（2023/07より）

「ソフトウェアエンジニアリングへの応用」パート
の資料作成および調査に貢献

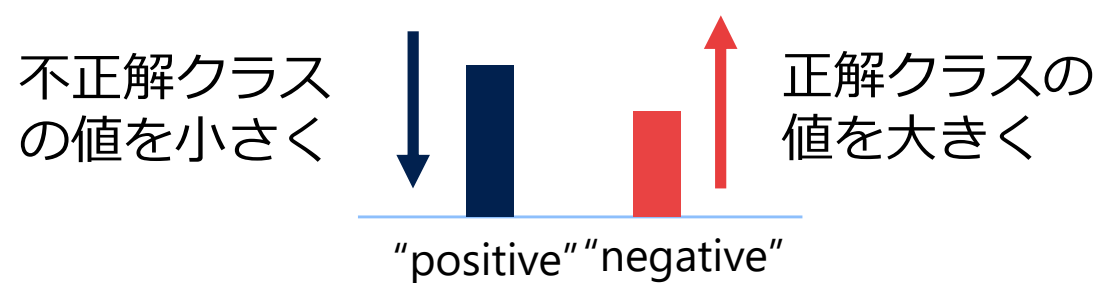
ニューラルネットワークを用いた 自然言語処理の基礎知識

深層学習時代の自然言語処理の概要



ニューラルネットワークの学習のイメージ

- 自然言語処理タスクの多くは「分類問題」として、ニューラルネットの出力を正解に近づけるように学習を行う

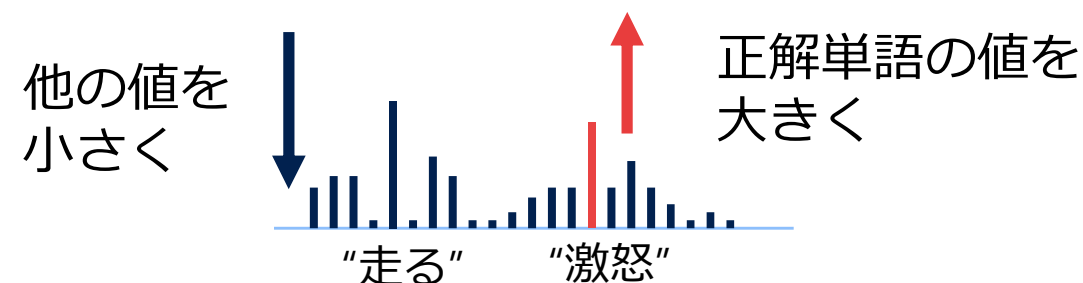


2次元ベクトルを出力

感情判定
モデル

“メロスは激怒した。”

タスク：テキストを2つのクラスに分類



語彙サイズの次元数のベクトルを出力

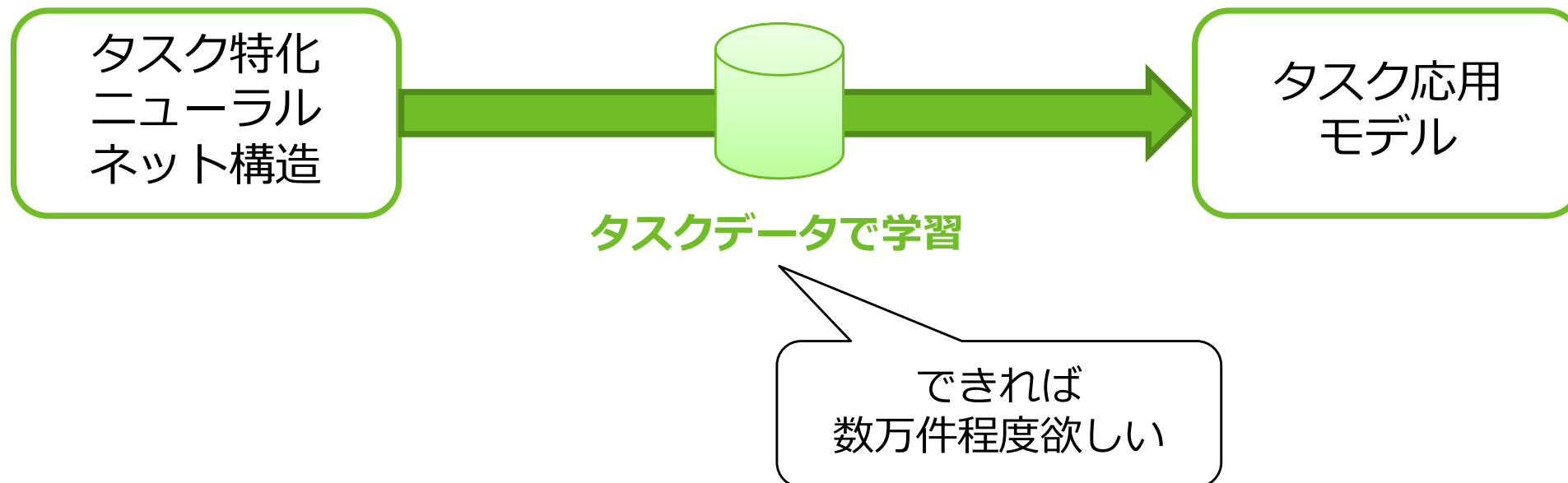
言語モデル

“メロスは”

タスク：入力に続く次の単語を生成

初期のニューラル自然言語処理

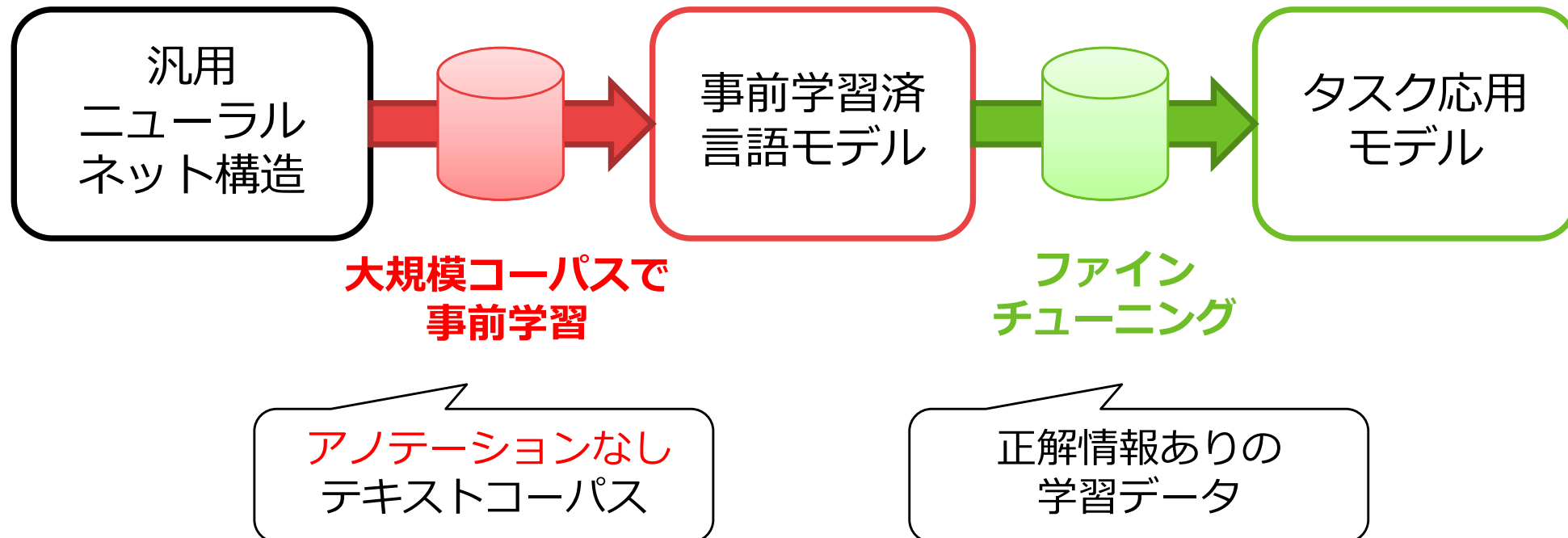
- タスク毎に適したニューラルネットワークを設計し，正解情報付きのデータセットで学習していた
- 深層学習により性能は高まったが，学習データが少ないと高い精度は実現しにくい課題が残っていた



言語モデルの「事前学習」が主流に

- 言語モデル (Language Model; LM) は, 単語 (トークン) 列の生成尤度をモデル化し、将来の (あるいは欠落した) トークンの確率を予測する
- 2018年頃から, 言語モデルが自然言語処理の「基盤」として有効であると注目を集めるようになり, タスクは大きく2種類に集約されていった

$$P(\mathbf{x}^{(t+1)} | \mathbf{x}^{(t)}, \dots, \mathbf{x}^{(1)})$$



自然言語処理タスクの分類

- 自然言語理解（NLU）と自然言語生成（NLG）の2種類に大きく分類
- タスクごとに適したアーキテクチャがある

自然言語理解

Natural Language Understanding

テキストの意味を理解

- テキスト分類
- 情報検索
- 感情分析
- 固有表現抽出
- Etc.

自然言語生成

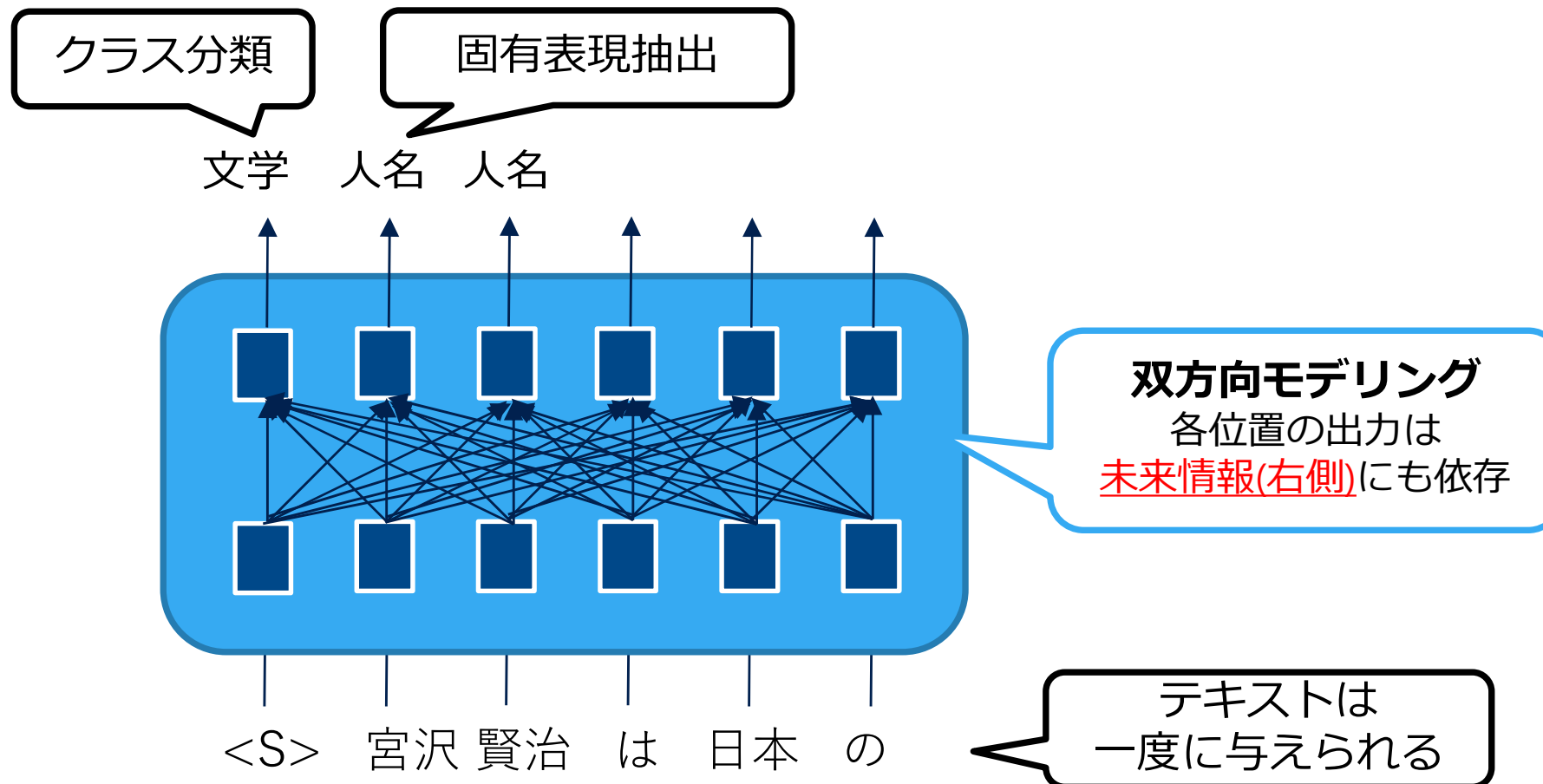
Natural Language Generation

入力からテキストを生成

- 機械翻訳
- 生成型要約
- 発話生成
- 言い換え
- Etc.

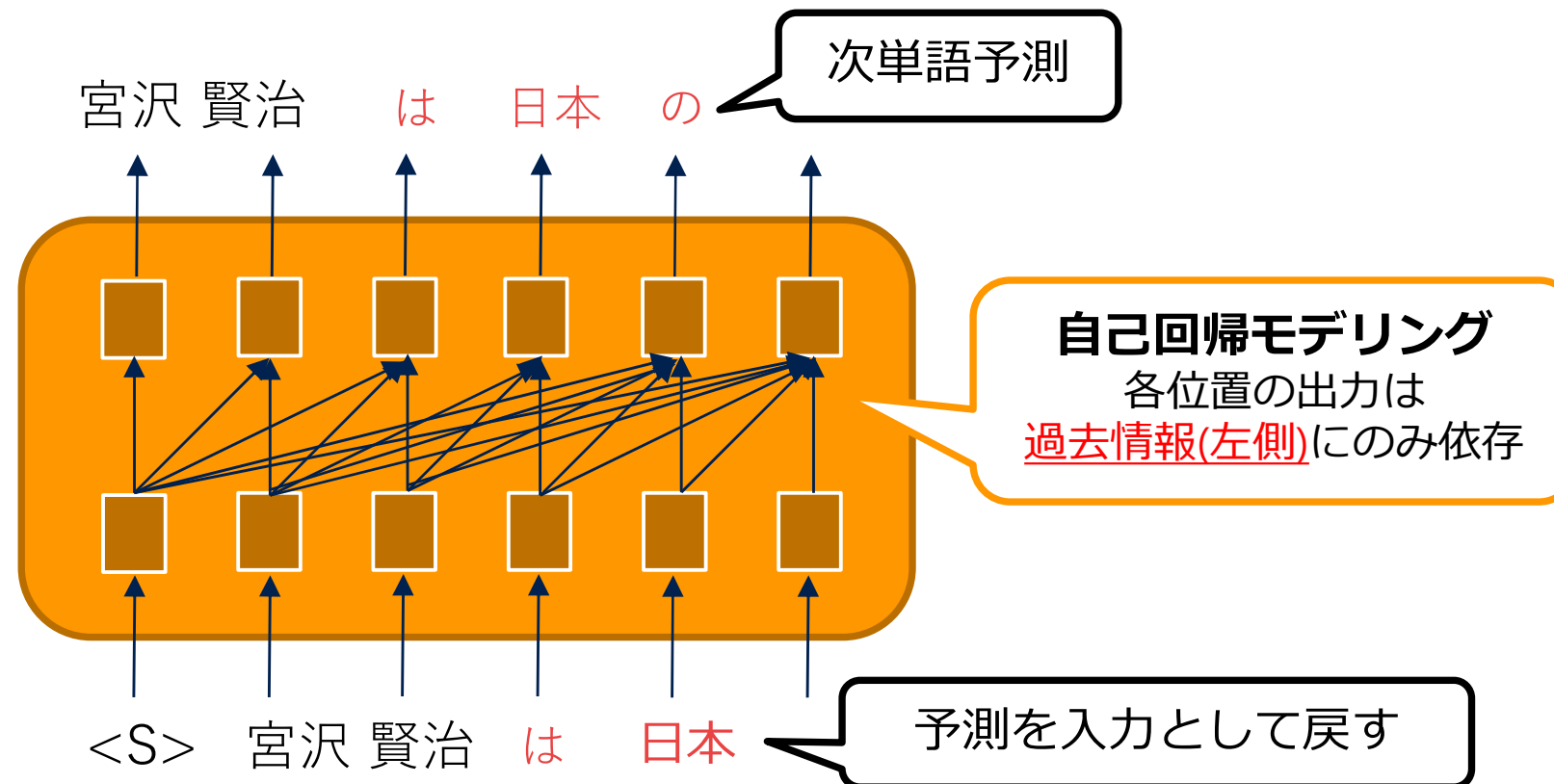
NLUタスクとモデルのイメージ

- エンコーダ型のモデルを用いて、テキストのクラス分類、各単語へのラベリングなどを行う
- 代表的モデル：BERT [Devlin+, NAACL'19]



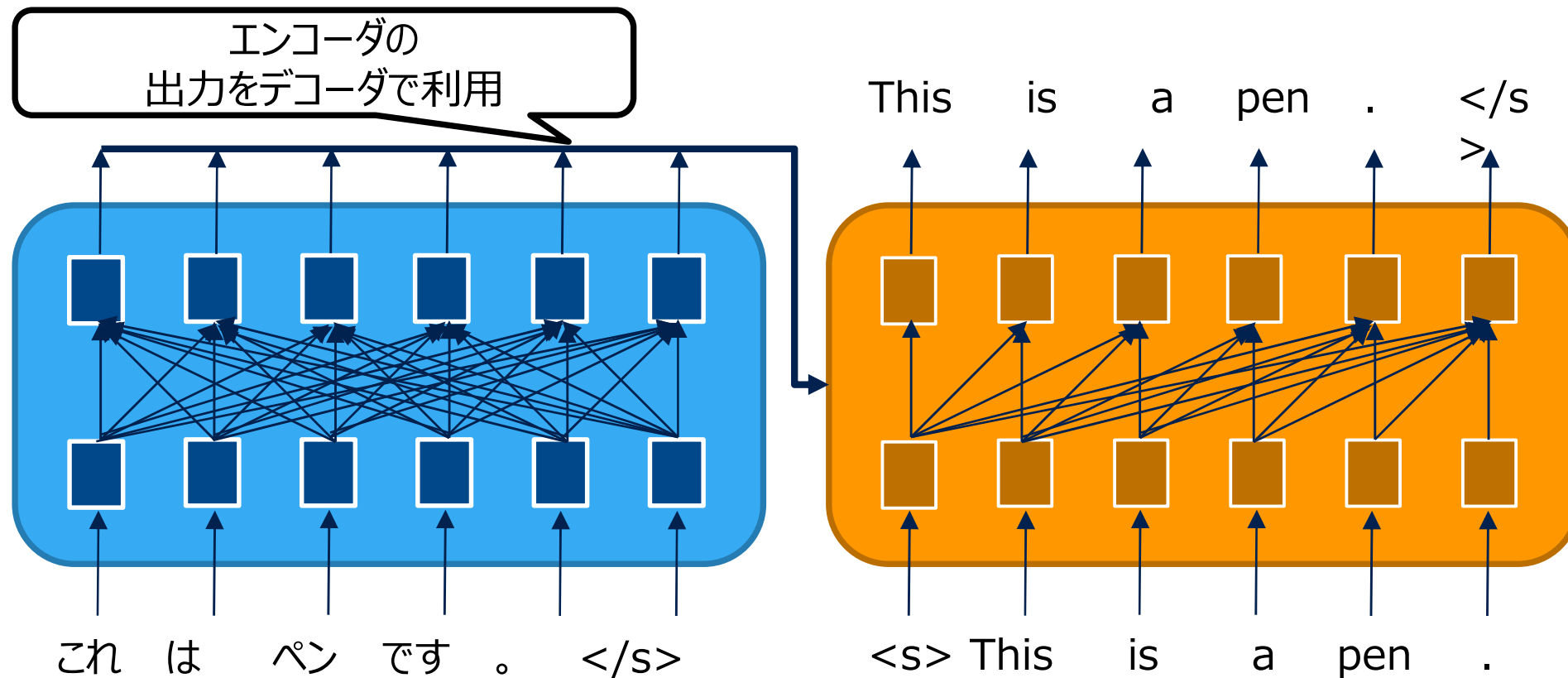
NLGタスクとモデルのイメージ (1/2)

- デコーダ型のモデルを用いて, 入力系列の続きを生成したりテキストAからテキストBへの変換を行う
- 最近はこのアーキテクチャが主流 (NLUタスクもこれで解く)
- 代表モデル: GPT-3 [Brown+, NeurIPS'20]



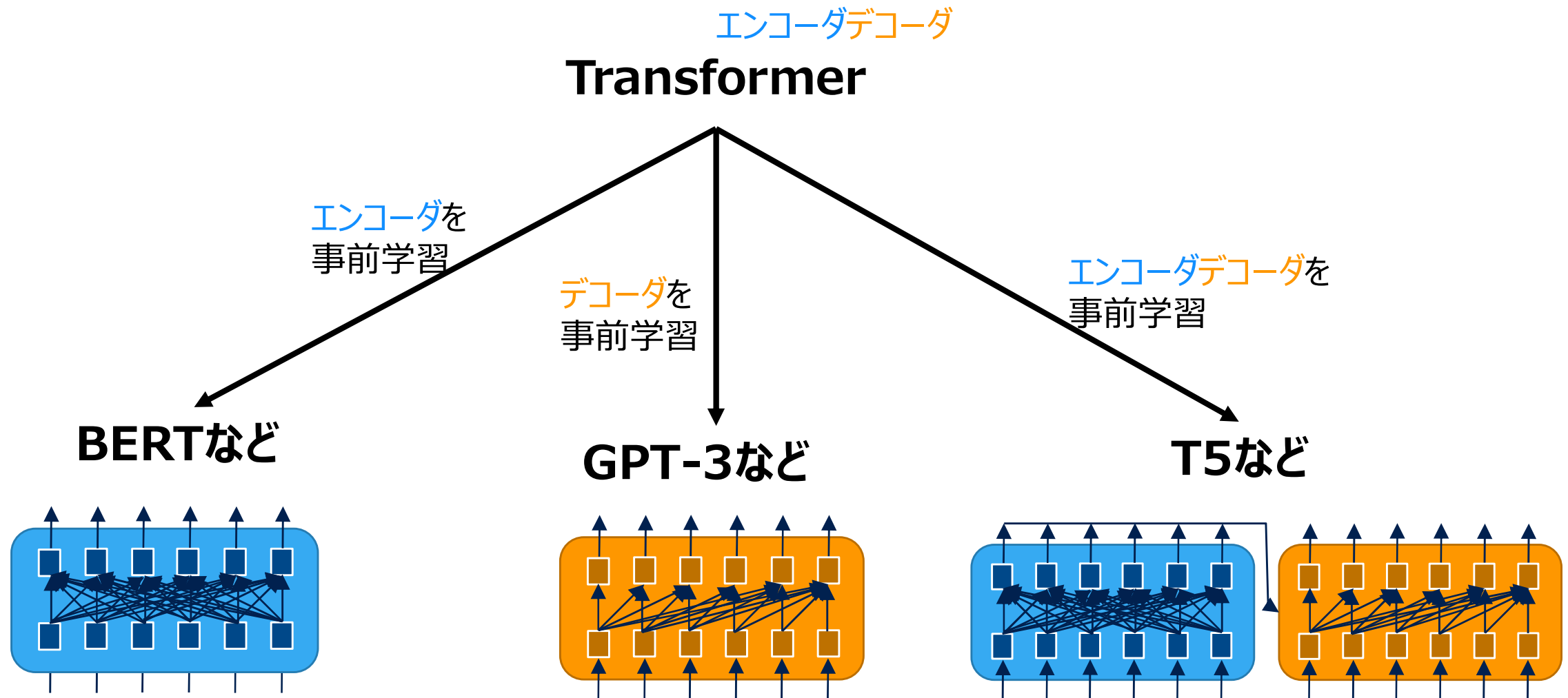
NLGタスクとモデルのイメージ (2/2)

- エンコーダデコーダ型のモデルを用いて, テキストAからテキストBに変換 (翻訳や要約など)
- 代表モデル: Transformer [Vaswani+, NIPS'17], T5 [Raffel, JMLR'21]



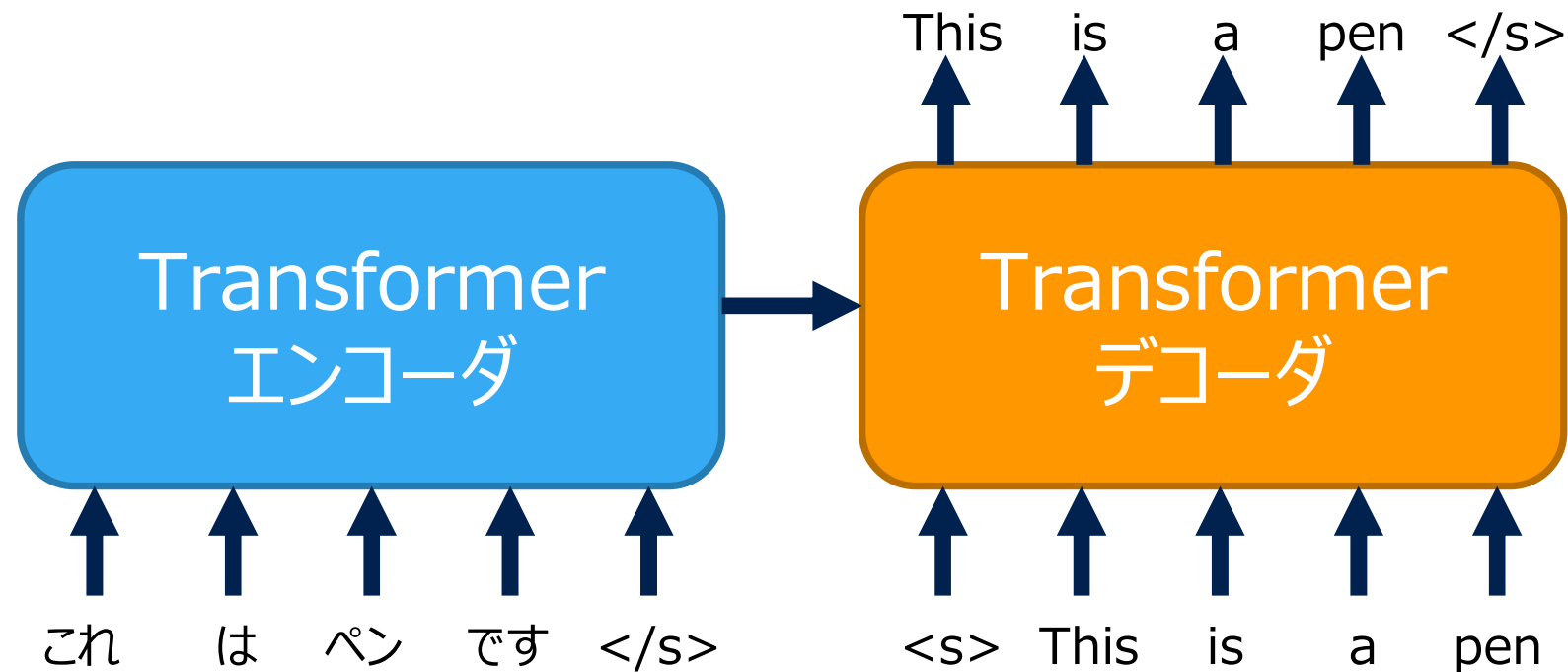
最近のベースモデル = Transformer

- 近年の事前学習済み言語モデルの多くが殆どが Transformer ベースとなっている



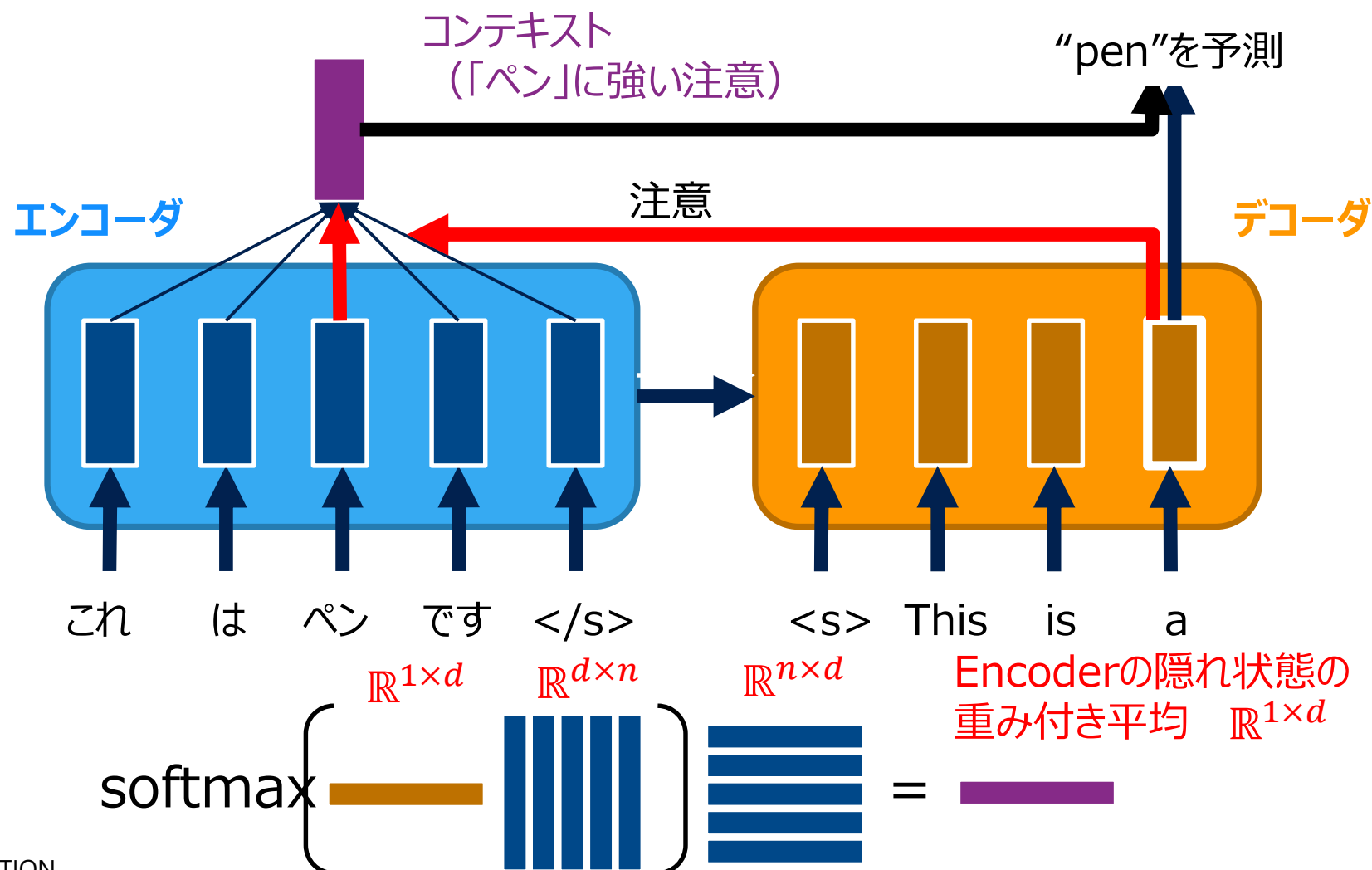
Transformer [Vaswani(Google)+, NIPS'17]

- 単語間の関係を理解（表現）するためにRNNやCNNを用いず，アテンションのみを用いたエンコーダデコーダ型モデルを提案
- 機械翻訳で評価実験し，当時の最高性能を更新



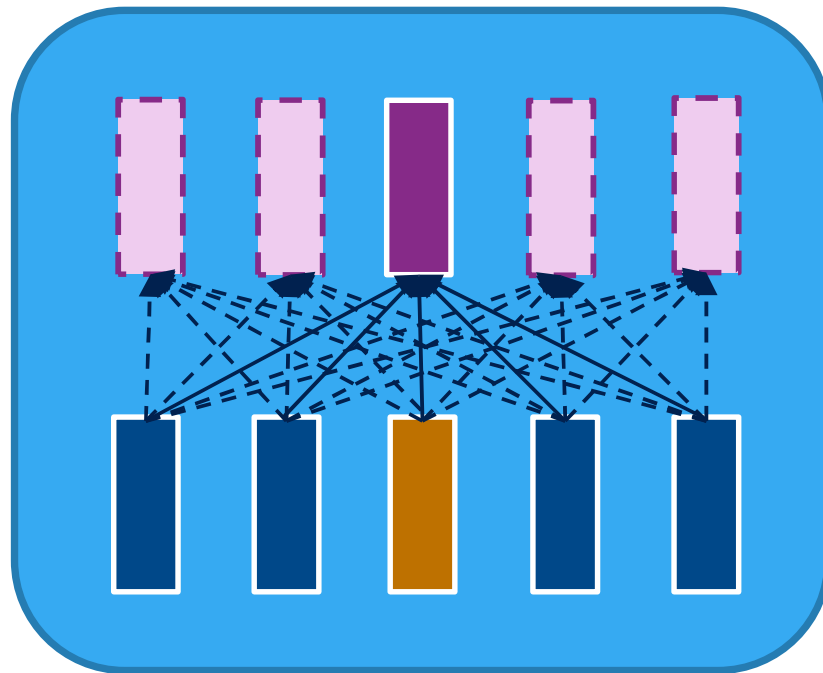
アテンション [Bahdanau(Jacobs U)+, ICLR'15]

- デコーダが1単語ずつ出力するときに、エンコードされた翻訳元テキストのどの部分に注意すれば良いかを与える機構（行列計算で実現）

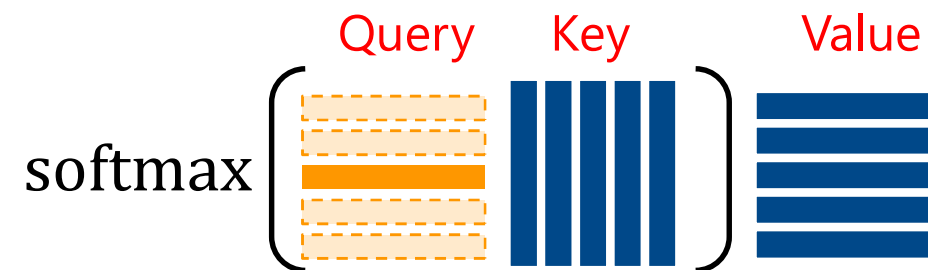


セルフアテンション

- エンコーダデコーダ間のクロスアテンションに加えて，系列単体でもアテンションを取る
- 系列の各位置をQuery，系列全体をKey， Valueとして考える



Key-Value辞書から，Queryに近いKeyのValueを取り出すイメージ

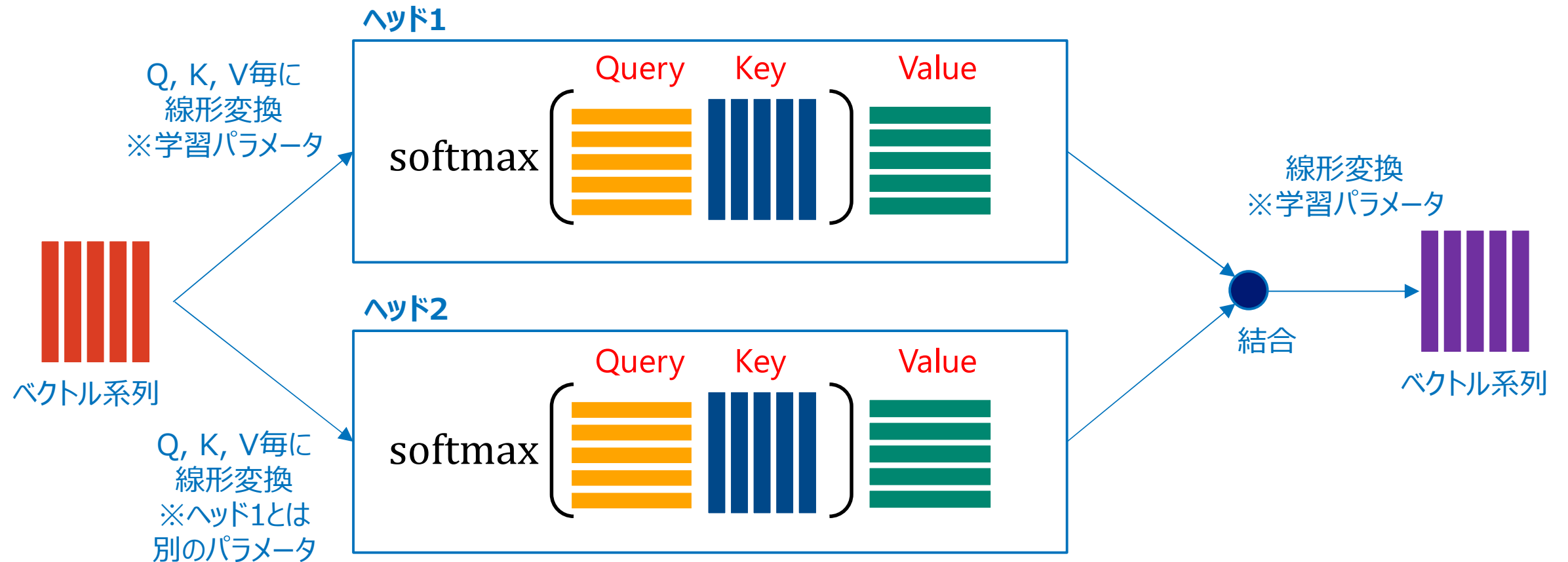


Valueの重み付き平均
 $\text{softmax}(QK^T)V$



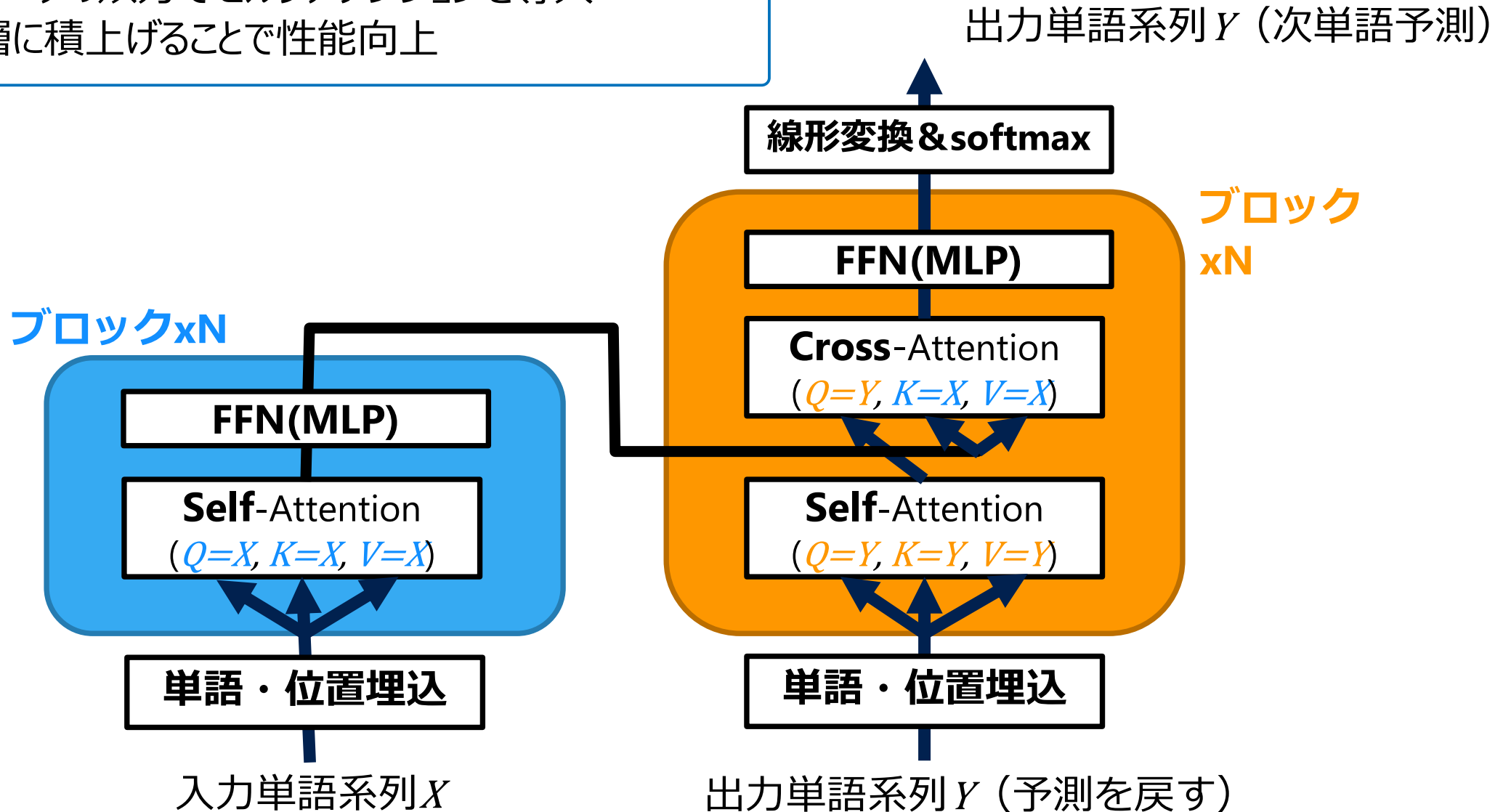
マルチヘッドアテンション

- アテンションを複数の“ヘッド”に分割して実施する
- ベクトルの次元方向に分割してアテンション処理を行い、最後に結合して元の次元数に戻すイメージ



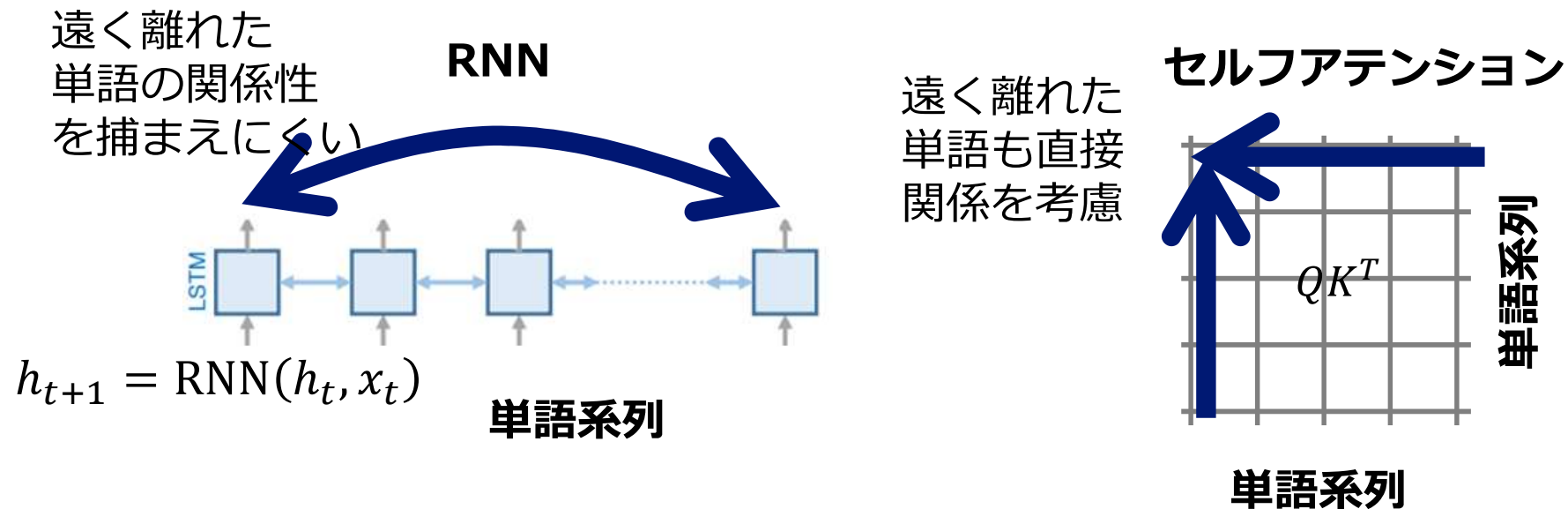
Transformerの簡略化イメージ

- エンコーダデコーダの双方でセルフアテンションを導入
- ブロックを多層に積上げることで性能向上



なぜセルフアテンションは有効か

- 単語系列の文脈の理解は主にRNN（LSTMやGRU）で行われてきたが、長期依存性の理解には限界があった
- セルフアテンションでは系列内で遠く離れた単語の関係性も直接考慮できるため性能向上に大きく寄与



ここまでのまとめ

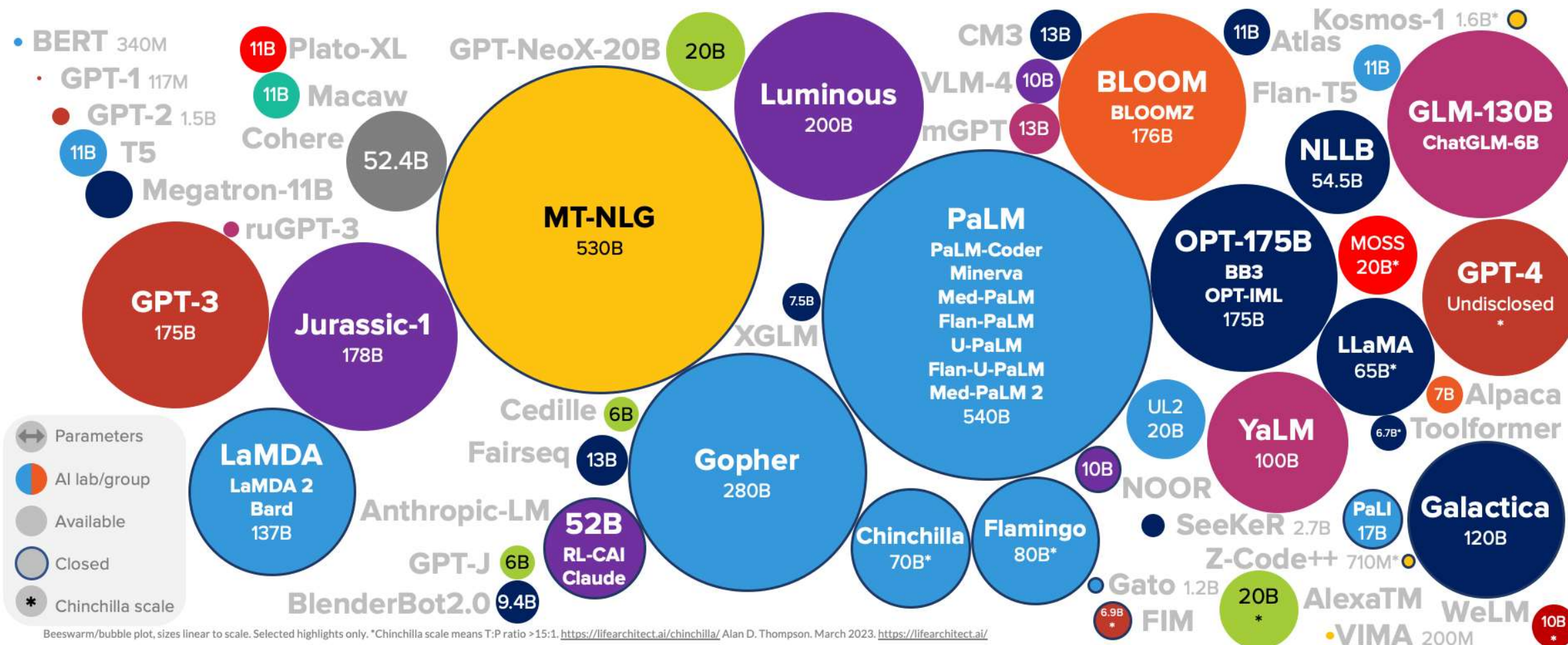
- ニューラルネットワークによる言語処理が主流に
- タスクはNLU（自然言語理解）とNLG（自然言語生成）に大別
- モデルはエンコーダ，デコーダ，エンコーダデコーダ型がある
- 最近のベースモデルはTransformer
- Transformerはテキストをはじめとした系列データのモデリングに強い

言語モデルの大規模化

BERTとGPT-3

言語モデルの大規模化

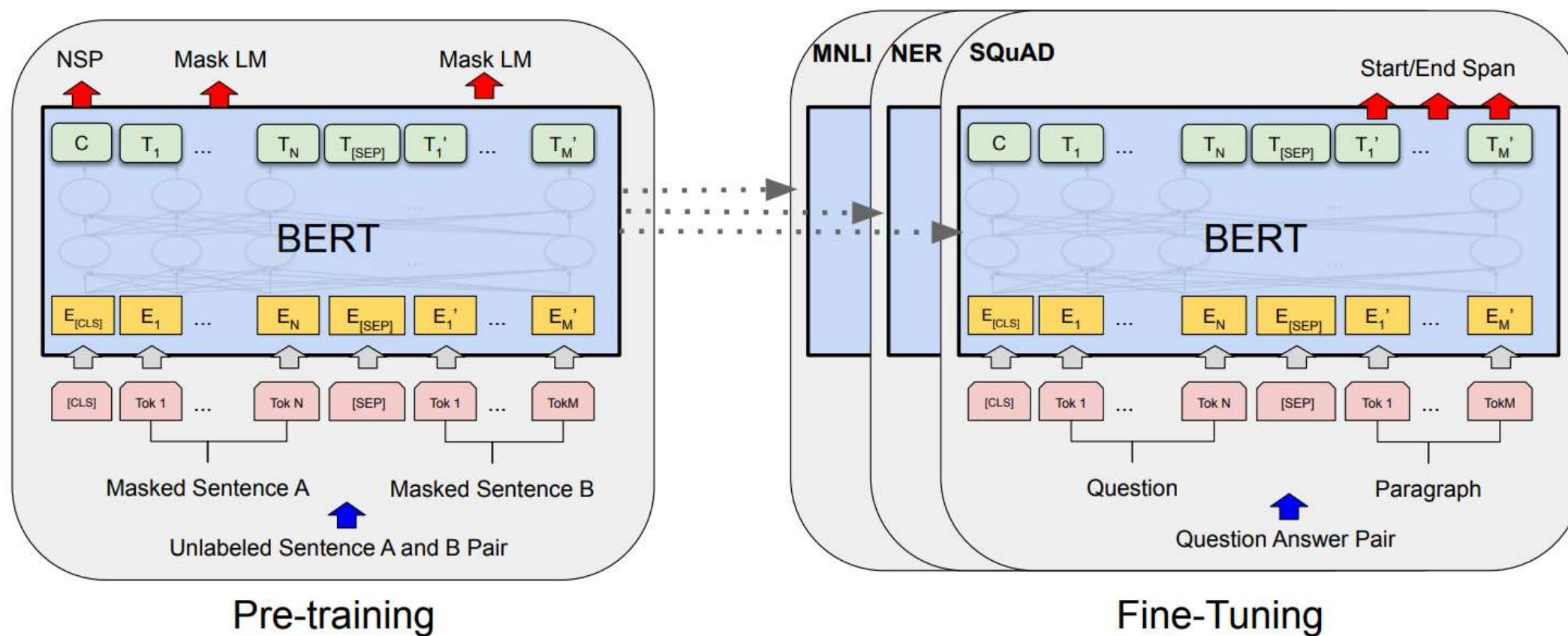
- モデルサイズ（パラメータ数）の大きな言語モデルが次々と作られている
- どこからが「大規模」というコンセンサスはない



Beeswarm/bubble plot, sizes linear to scale. Selected highlights only. *Chinchilla scale means T:P ratio > 15:1. <https://lifearchitect.ai/chinchilla/> Alan D. Thompson, March 2023. <https://lifearchitect.ai/>

BERT [Devlin(Google)+, 2018/10, NAACL'19]

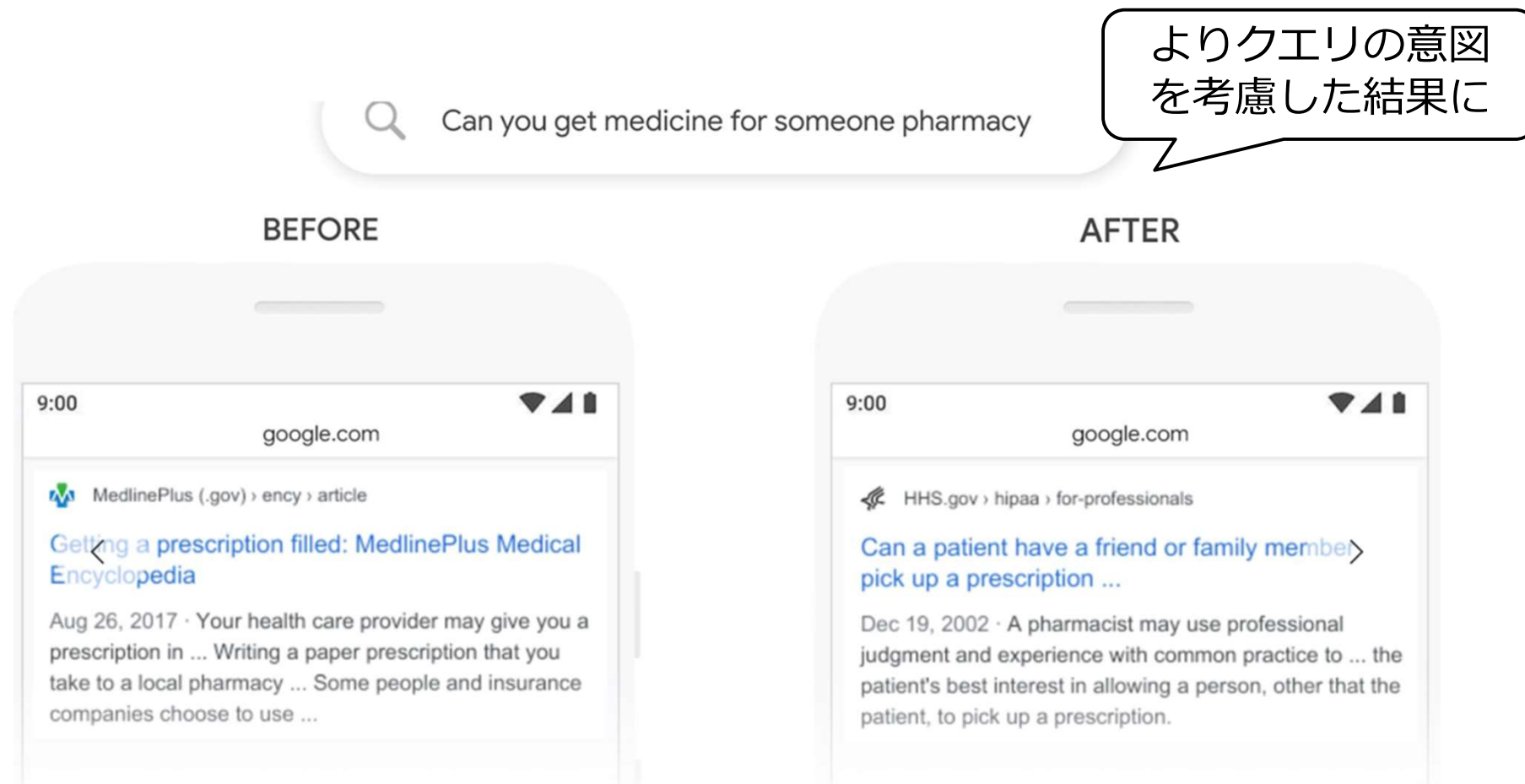
- 大量のテキストデータで事前学習した, 24層の巨大なTransformerエンコーダに基づく言語モデル (340M = 3.4億個のパラメータ)
- 多数の自然言語理解 (NLU) タスクへのファインチューニングで高い性能を実現して注目を浴びる



<https://www.aclweb.org/anthology/N19-1423/>

BERTの検索エンジンへの導入

- 2019年10月25日（BERT発表から1年後）、Googleは検索エンジンのアルゴリズムをBERTベースに更新
- 2019年12月10日には日本語含む72言語に拡張



BERTが優れた性能を達成したタスクの例：

機械読解 (SQuAD 1.1)

[Rajpurkar(Stanford)+, EMNLP'16]



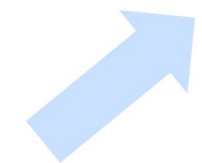
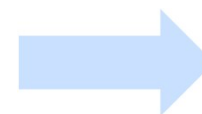
- 質問に対してテキストを読み解いて回答するタスク

入力：Wikipediaの段落

In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under **gravity**. The main forms of precipitation include drizzle, rain, sleet, snow, **grau-pel** and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals **within a cloud**. Short, intense periods of rain in scattered locations are called "showers".

入力：質問

Where do water droplets collide with ice crystals to form precipitation?



出力：回答

within a cloud

テキスト中の**任意の範囲**を抽出して回答

SQuAD 1.1で人間超えのスコアを達成



- 機械読解に特化したニューラルネット構造を持たずに、人間の質問応答スコアを大きく超える性能を達成

SQuAD1.1 Leaderboard

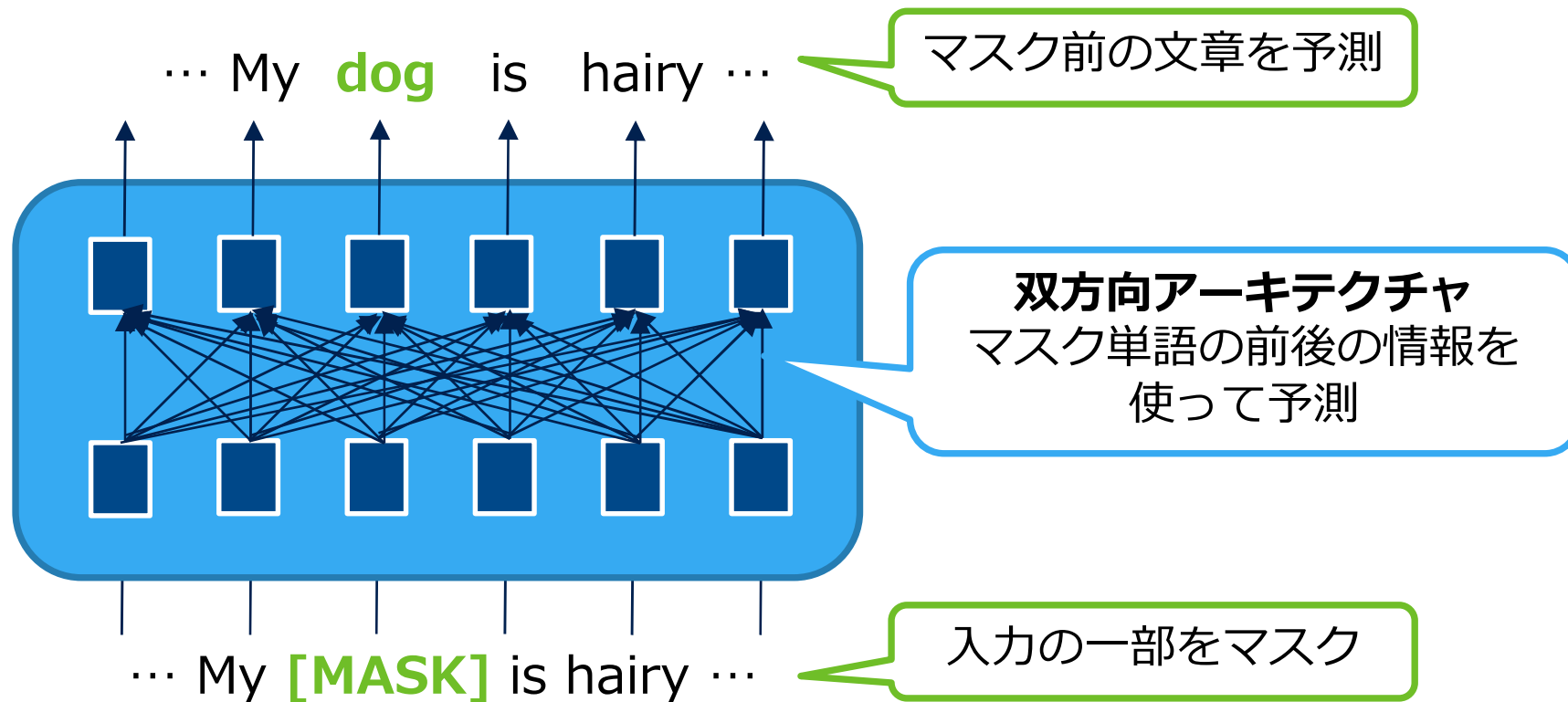
Since the release of SQuAD1.0, the community has made rapid progress, with the best models now rivaling human performance on the task. Here are the ExactMatch (EM) and F1 scores evaluated on the test set of v1.1.

Rank	Model	完全一致 EM	部分一致 F1	
	Human Performance Stanford University (Rajpurkar et al. '16)	82.304	91.221	人間の正答率 82.304%
1 Oct 05, 2018	BERT (ensemble) Google AI Language https://arxiv.org/abs/1810.04805	87.433	93.160	BERTの正答率 87.433%
2 Oct 05, 2018	BERT (single model) Google AI Language https://arxiv.org/abs/1810.04805	85.083	91.835	

<https://rajpurkar.github.io/SQuAD-explorer/>

BERTの特徴(1) マスク化言語モデリング

- 穴埋め問題により，文脈の中で単語が出現する確率を予測する
- 大量のテキストさえあれば，人間が追加で正解をアノテーションすることなく学習可能（自己教師あり学習）



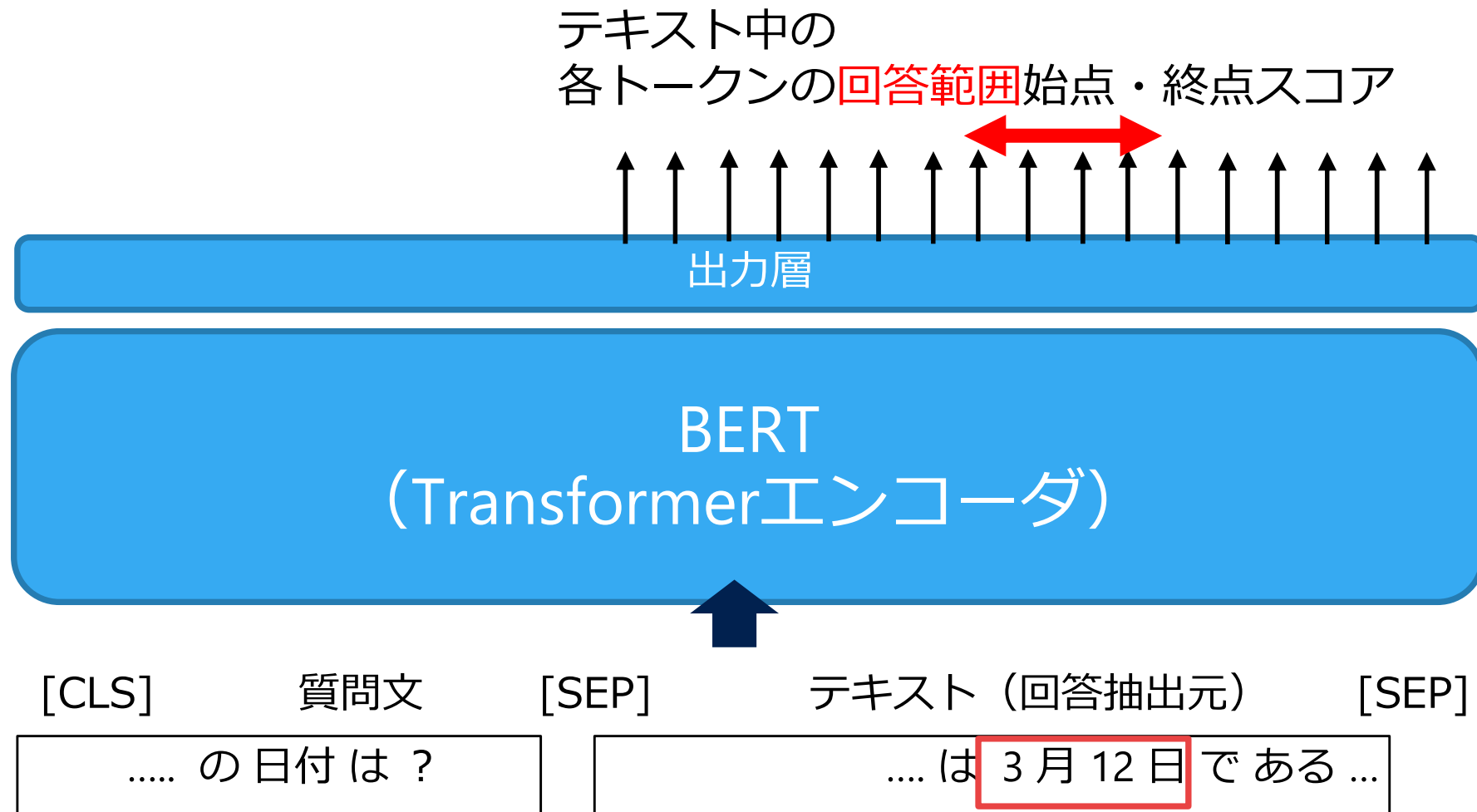
BERTの特徴(2) 次文章予測

- 自然言語理解 (NLU) のタスクは2つの文の関係性理解が多い
- そこで、文書中で隣接する文/ランダムな2文をセパレータで繋げて、隣接する文か否かの判定を穴埋め予測と同時に学習



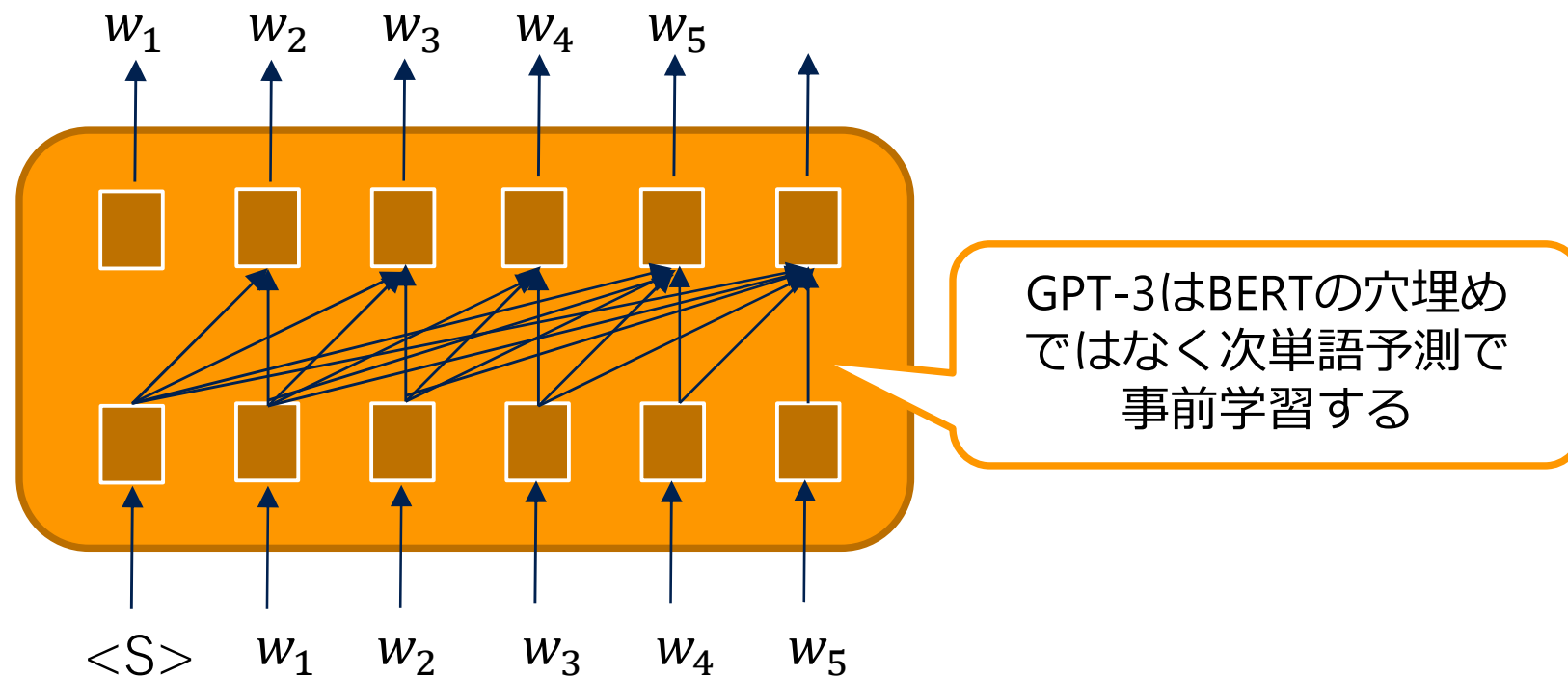
BERTの特徴(3) ファインチューニング

- シンプルな出力層を1層だけ基盤モデルの最上部に追加する
- 例：機械読解では回答範囲の始点・終点予測用に線形変換を追加



GPT-3 [Brown(OpenAI)+, 2020/07, NeurIPS'20]

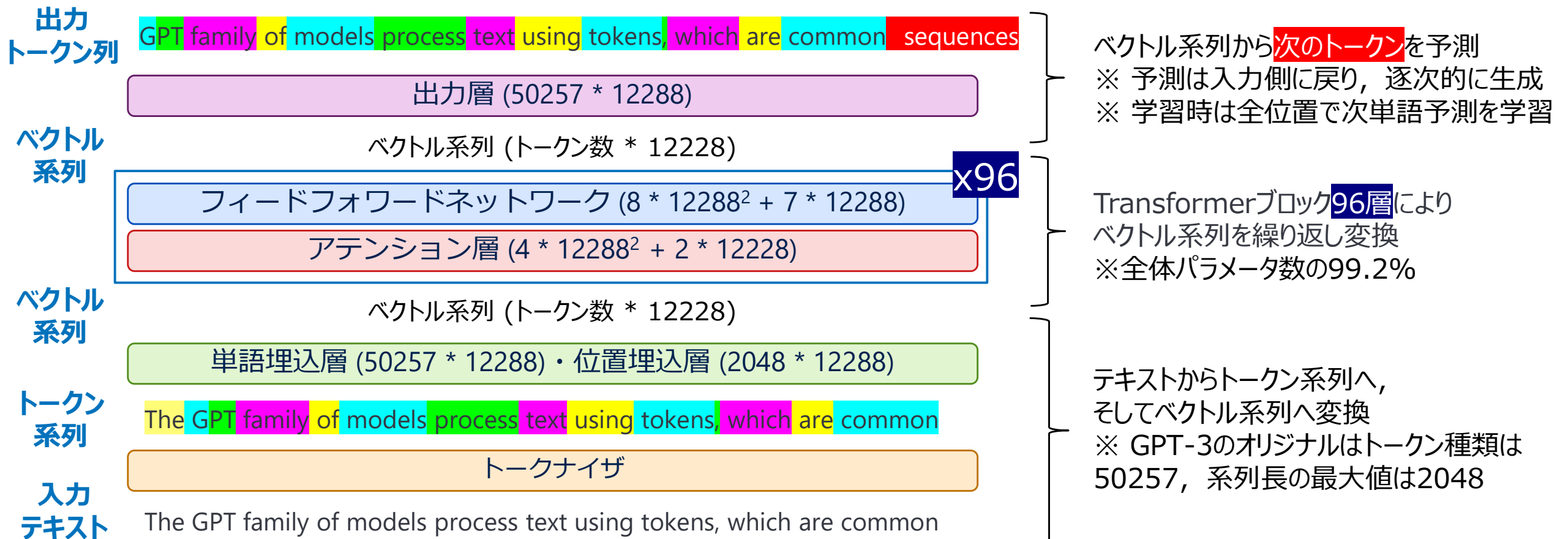
- 超大量のテキストデータ（3000億トークン）で事前学習した、96層の超巨大なTransformerデコーダに基づく言語モデル
- パラメータ数は175B = 1750億個（ファイルサイズで言うと700GB程度）
- 参考：BERTは 24層， 3.4億個のパラメータ， 約3.3億トークン



<https://papers.nips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html>

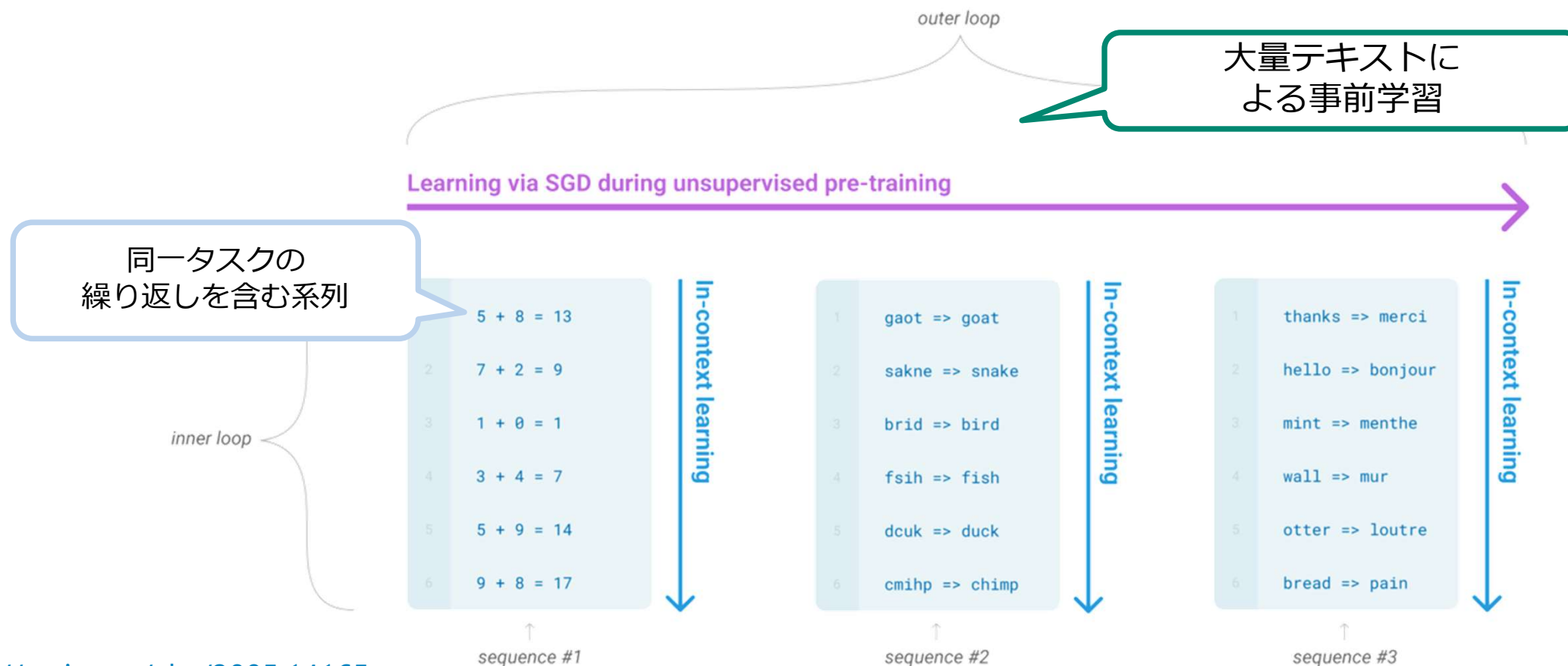
GPT-3の構造

- GPTは生成タスクに適したデコーダ（自己回帰）タイプ。BERTのエンコーダタイプとは異なる。
- GPT-3（下図）は96層で合計175Bのパラメータを持つ。ChatGPTやGPT-4の構造・パラメータ数は不明（Newton報「GPT-3.5は355B」は取り下げへ [\[Okumura, 2023/06\]](#)）



GPT-3の膨大な事前学習が可能にすること

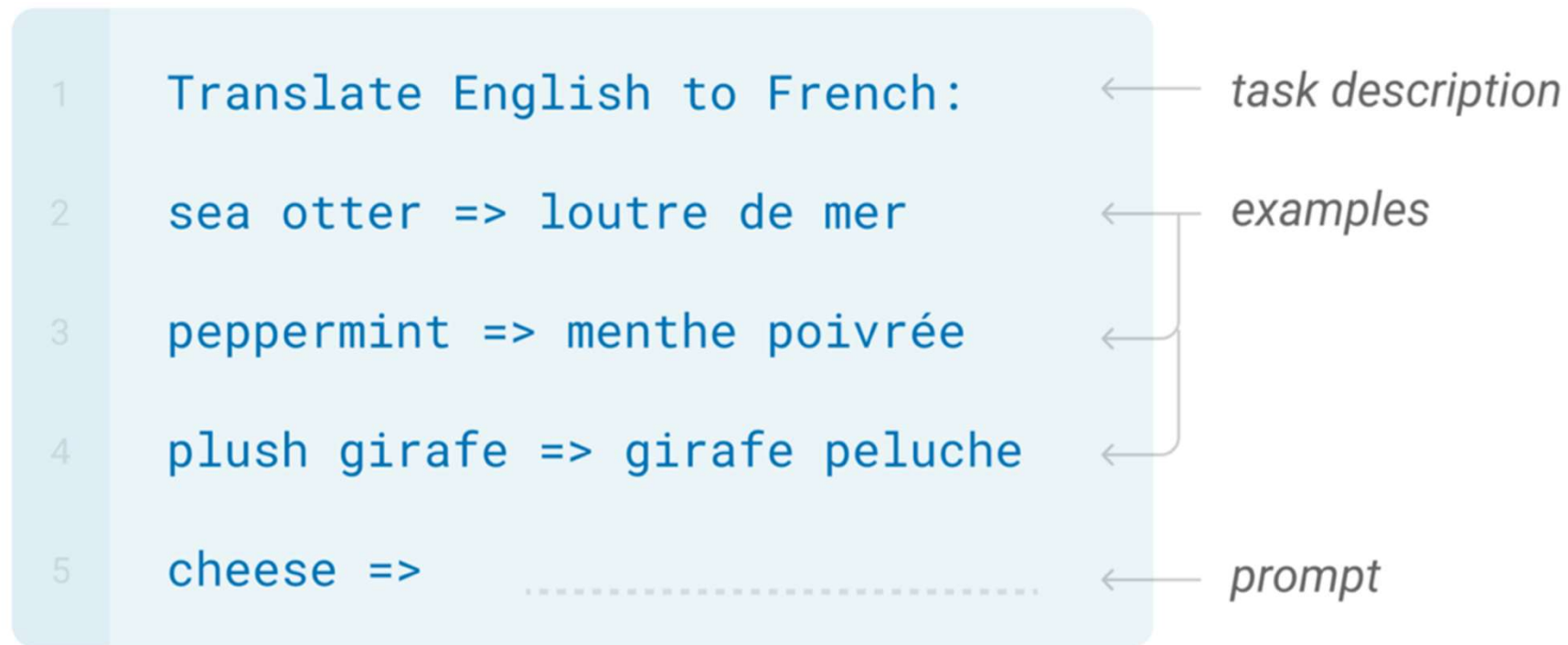
- 人間の様な「少数の例示による学習」が可能になってきた
- 大量のテキストの中には同一タスクの例が繰り返し現れるコンテキストも含まれており，こうした学習が例示による学習能力を向上させる



<https://arxiv.org/abs/2005.14165>

GPT-3のfew-shot学習

- タスク説明と少数の例を与えるのみでモデルの更新無しに問題を解く



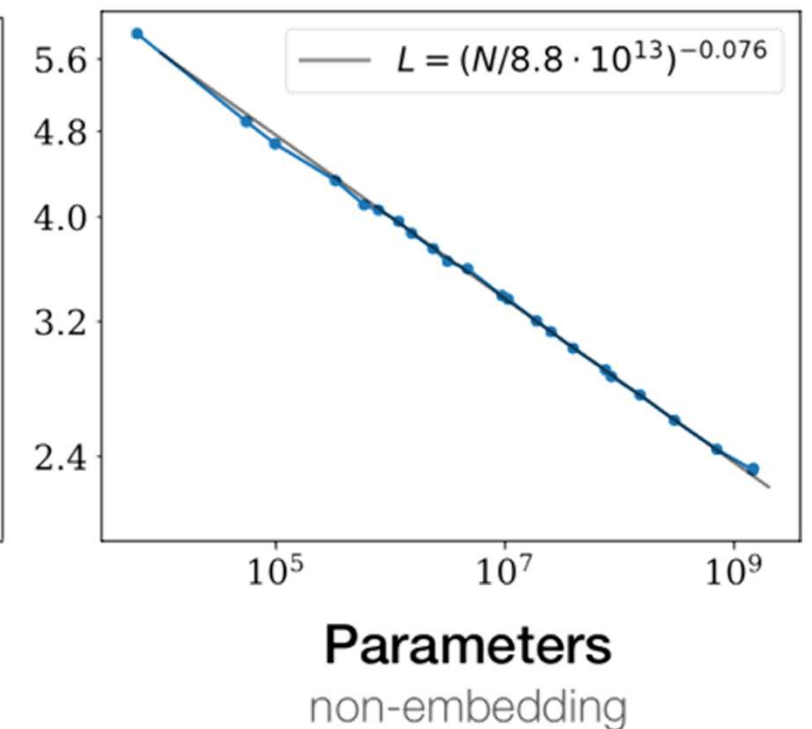
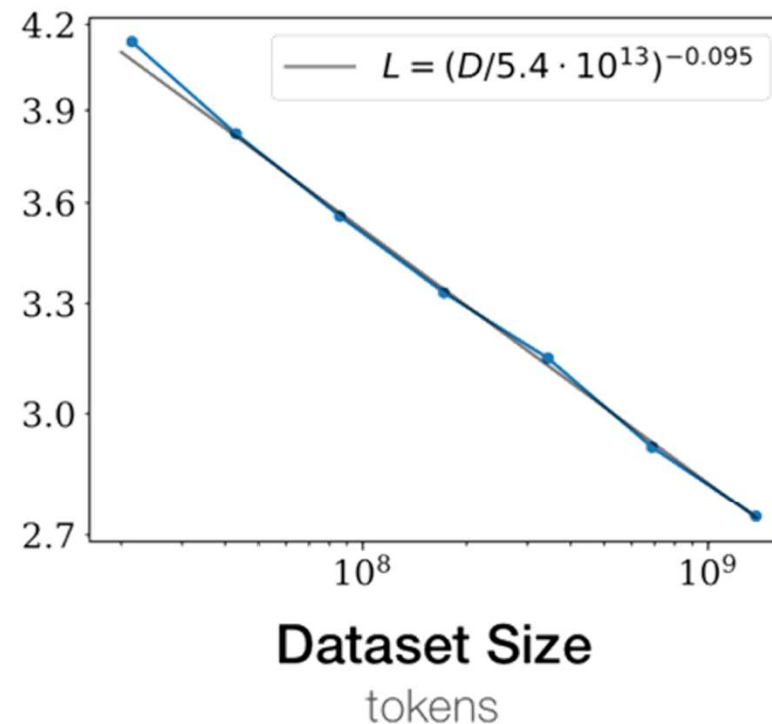
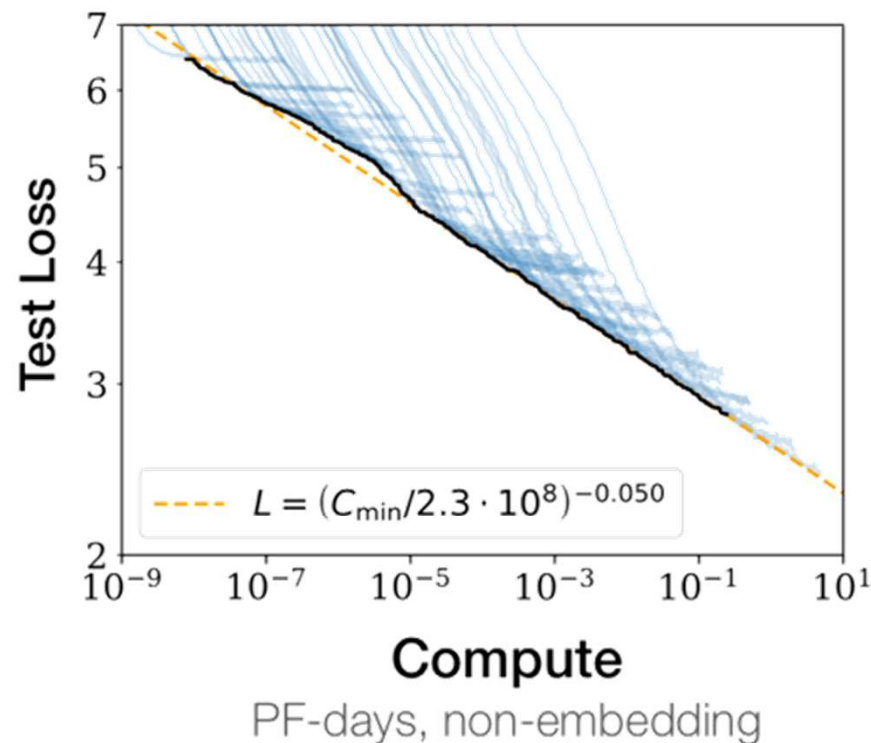
ここまでを一つの系列としてGPT-3に入力
→ 回答を続きの文章として出力する

<https://arxiv.org/abs/2005.14165>

Scaling laws [Kaplan (OpenAI)+, 2020/01]



- 計算量（学習回数），学習トークン数，モデルパラメータ数，に言語モデリング（次単語予測）の性能は強く依存することを実験的に示す
- 大量の計算機資源を用いて，大量のデータを，大きいモデルで学習すれば良いモデルができる！



<https://arxiv.org/abs/2001.08361>

Chain-of-Thought [Wei(Google)+, 2022/01]



- 思考の途中過程もプロンプトとして与えることによって、算術や常識の推論能力が大きく向上する現象
- Chain-of-Thoughtの能力を発現するにはある程度のモデルサイズが必要（8Bだと不十分であることが多い）

Standard Prompting

Model 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

Model 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. ✅

プロンプト
に思考方法
を追記

思考過程も
含めて正解
できる

“Let’s think step by step” [Kojima+, 2022/05]



- いわゆる「おまじない」として“Let’s think step by step”をプロンプトに与えると、Chain-of-thoughtのように推論仮定をモデルが出力するようになる
- 「プロンプトチューニング」の大事さを顕著に表す例と言える

(c) Zero-shot

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A: The answer (arabic numerals) is

(Output) 8 X

(d) Zero-shot-CoT (Ours)

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

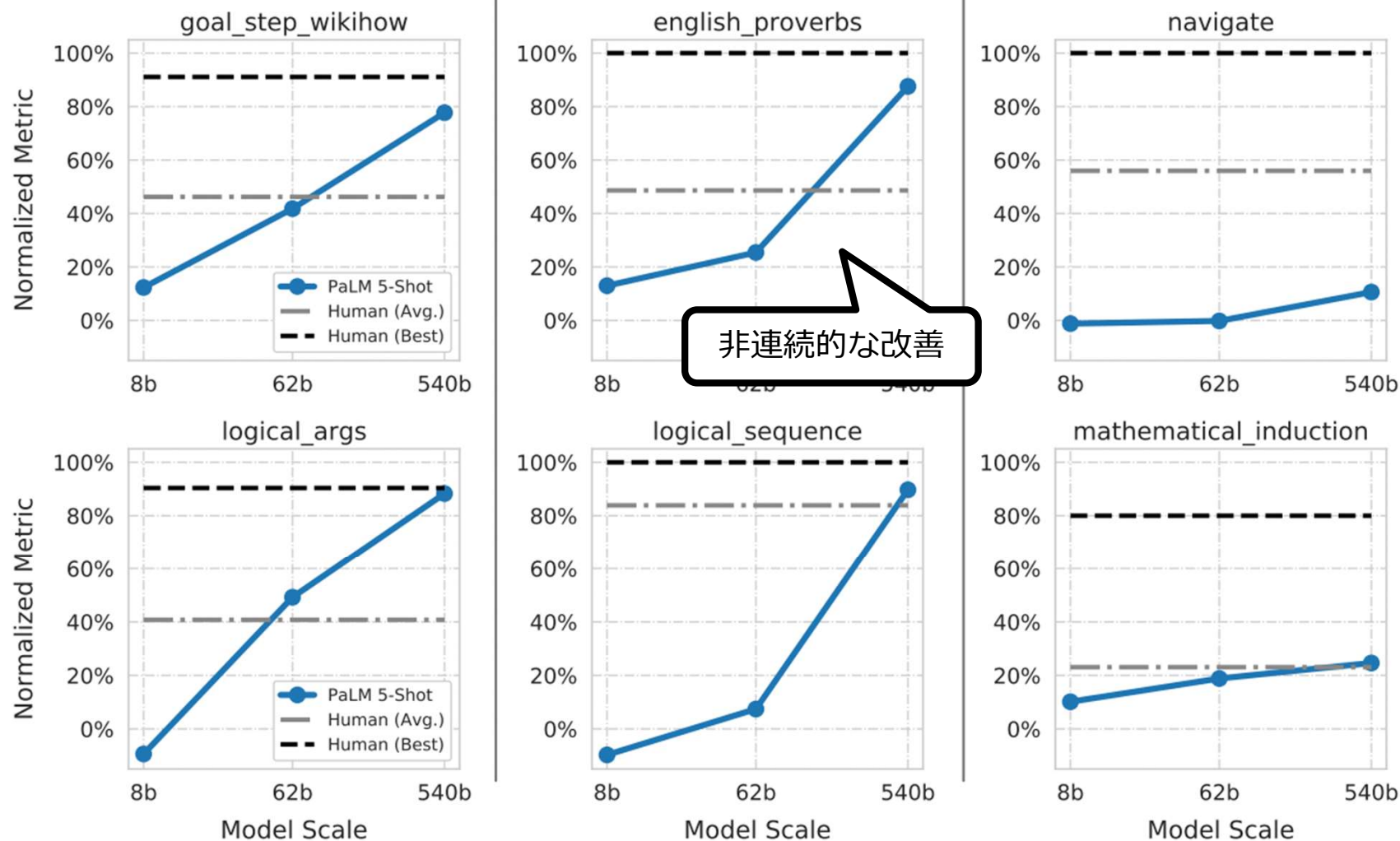
A: **Let’s think step by step.**

(Output) *There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls. ✓*

<https://arxiv.org/abs/2205.11916>

PaLM [Chowdhery (Google)+, 2022/04/19]

- 540Bの巨大モデル. 幾つかのタスクでモデルサイズの増加により, 非連続的に性能が改善
- タスクによっては, 540Bモデルでもファインチューニングモデルや人間の性能には届いていない



モデルサイズ vs.
タスク性能

非連続的な改善

Chinchilla [Hoffman (DeepMind)+, 2022/03/29]



- 400種類のモデルと学習トークン数の組み合わせで実験し, LLMの学習に必要なトークン数を実験的に示した
- Chinchilla (70Bパラメータ/1.4Tトークン) は, Gopher (280Bパラメータ/300Bトークン) の性能を多くのタスクで凌駕した

Parameters	FLOPs	FLOPs (in <i>Gopher</i> unit)	Tokens
400 Million	1.92e+19	1/29,968	8.0 Billion
1 Billion	1.21e+20	1/4,761	20.2 Billion
10 Billion	1.23e+22	1/46	205.1 Billion
67 Billion	5.76e+23	1	1.5 Trillion
175 Billion	3.85e+24	6.7	3.7 Trillion
280 Billion	9.90e+24	17.2	5.9 Trillion
520 Billion	3.43e+25	59.5	11.0 Trillion
1 Trillion	1.27e+26	221.3	21.2 Trillion
10 Trillion	1.30e+28	22515.9	216.2 Trillion

GPT-3のサイズなら
3.7Tトークン必要
(実際は300B)

ここまでのまとめ

- 2018/10のBERTの発表によりパラダイム・シフトが発生し、「大規模モデルの事前学習→ファインチューニング」が主流に
- 2020/07のGPT-3の発表以降は、Few-shot学習（プロンプト＝入力テキストとして少量の例示）が大きな注目を集めている
- BERTもGPT-3も大量のテキストがあれば学習可能（自己教師あり学習）
- 大量の計算機資源を用いて、大きいモデルを学習する、というアプローチが活発化している
- 単純にモデルを大きくすればそれで良い、という訳でも無く、学習データの量や質の重要性も指摘され始めた

大規模言語モデルのさらなる発展

Instruction-tuning, ChatGPT, LLaMA

汎用人工知能が遠い夢では無くなる

- 大規模言語モデル（Large Language Model; LLM）を活用した ChatGPT [OpenAI, 2022/11] により、1つのモデルで様々な自然言語処理タスクを「**特化した学習無し**」に解けることが実証されてきている
- GPT-4 [OpenAI, 2023/03] ではChatGPTをさらに上回り、人レベルの言語理解・生成能力を持つ

（情報の検索）

論文“attention is all you need”の要約を教えてください

>こちらは「Transformer」という深層学習モデルを提案した論文です。

Transformerは、seq2seq（順方向から順方向への変換）タスクや自然言語処理タスクなどに. . .

新たな情報検索の体験

（タスクの指示）

このプログラムの誤りを直して

>こちらが修正したコードになります。

```
<html>
<head>
<title>Breakout</title>
</head>
<body>
<canvas id="game-canvas" width="400" height="400"></canvas>

<script>
// キャンパスの要素を取得する
var canvas = document.getElementById("game-canvas");
var ctx = canvas.getContext("2d");

// ボールの状態を表す変数
var x = 100;
var y = 100;
```

AIが人に変わり
任意の言語タスクを遂行

（コンテンツの生成）

~~~~に関する  
学術論文を書いて

nature

NEWS | 12 January 2023

Abstracts written by ChatGPT fool  
scientists

Researchers cannot always

ChatGPTの執筆した論文要旨  
の1/3を専門家が見破れず

AIのトップ会議ICMLは言語モデルによる論文の執筆を禁止

# Instruction-tuning (Flan) [Wei(Google)+, 2021]

- 既存の自然言語処理タスクを自然言語によるインストラクションを付与して変換し、大規模言語モデルをインストラクションに従った出力が出来るようにチューニング
- Flan v2 [Chung(Google)+, 2022/11]では1800超のタスクまで拡大させて性能アップ

## Finetune on many tasks (“instruction-tuning”)

|                                                                                                                                                                                                                                                                                                                           |                                                                                                                                                                                                                                        |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>Input (Commonsense Reasoning)</b></p> <p>Here is a goal: Get a cool sleep on summer days.</p> <p>How would you accomplish this goal?</p> <p>OPTIONS:</p> <p>-Keep stack of pillow cases in fridge.</p> <p>-Keep stack of pillow cases in oven.</p> <p><b>Target</b></p> <p>keep stack of pillow cases in fridge</p> | <p><b>Input (Translation)</b></p> <p>Translate this sentence to Spanish:</p> <p>The new office building was built in less than three months.</p> <p><b>Target</b></p> <p>El nuevo edificio de oficinas se construyó en tres meses.</p> |
| <p>Sentiment analysis tasks</p> <p>Coreference resolution tasks</p> <p>...</p>                                                                                                                                                                                                                                            |                                                                                                                                                                                                                                        |

## Inference on unseen task type

**Input (Natural Language Inference)**

Premise: At my age you will probably have learnt one lesson.

Hypothesis: It's not certain how many lessons you'll learn by your thirties.

Does the premise entail the hypothesis?

OPTIONS:

-yes   -it is not possible to tell   -no

**FLAN Response**

It is not possible to tell

<https://arxiv.org/abs/2109.01652>



# InstructGPT [Ouyang (OpenAI)+, 2022/03/04]



- LLMを「人の指示に従って回答できるように」「人の感覚に合った出力ができるように」チューニング
- 1.3Bのモデルであっても, 175BのGPT-3よりも事実性や有害性の観点で良いテキストを生成できた

*Explain the moon landing to a 6 year old in a few sentences.*      指示 (インストラクション)

GPT-3

GPT-3は指示に従った出力ができていない

Explain the theory of gravity to a 6 year old.

Explain the theory of relativity to a 6 year old in a few sentences.

Explain the big bang theory to a 6 year old.

Explain evolution to a 6 year old.

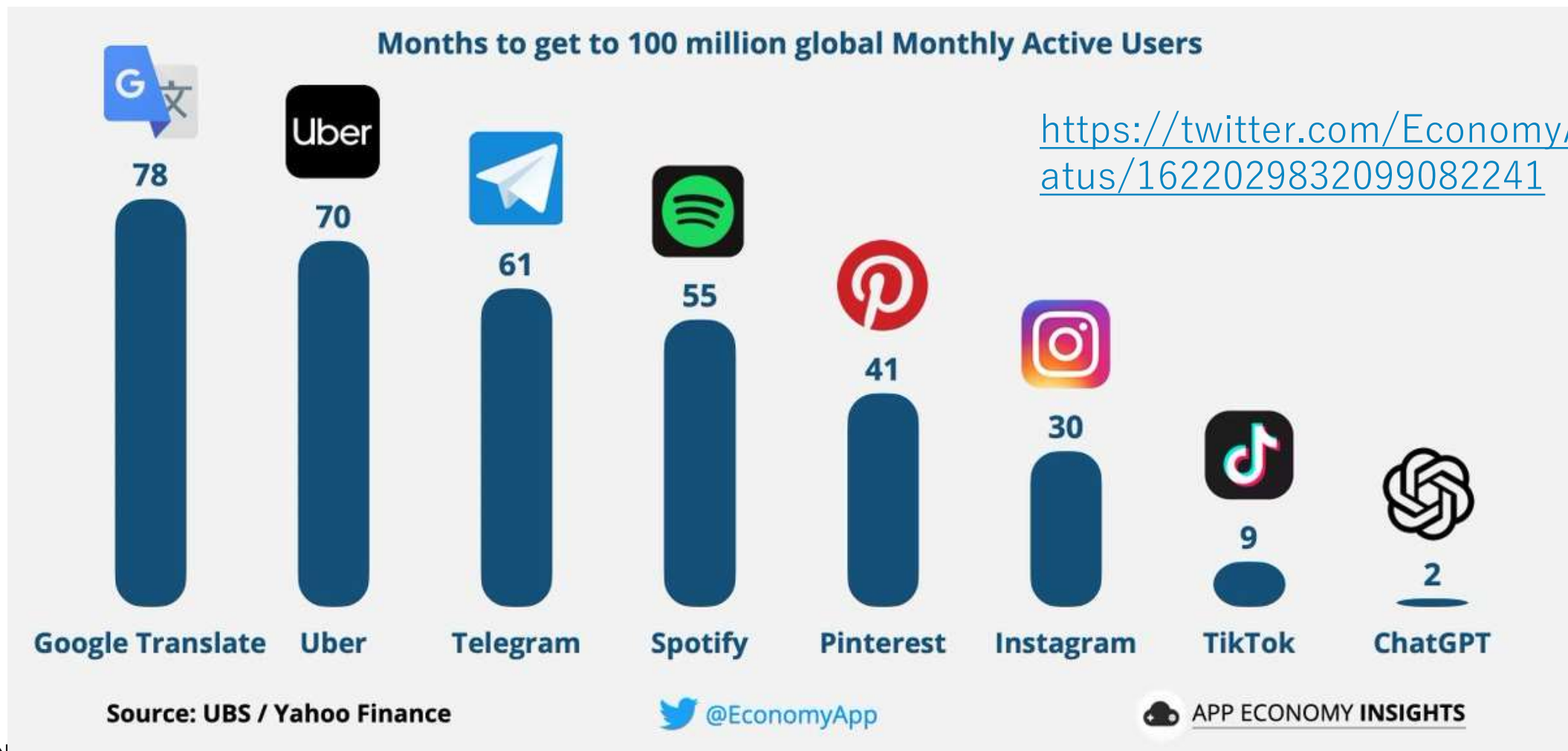
InstructGPT

指示に従った出力ができています

People went to the moon, and they took pictures of what they saw, and sent them back to the earth so we could all see them.

# ChatGPT [OpenAI, 2022/11/30]

- InstructGPTを発展させ, 人のフィードバックに基づいて対話に最適化したLLM (パラメータ数など不明)
- その高い性能から最速でアクティブユーザ1億人に到達. 社会に大きな影響を与えるレベルに到達



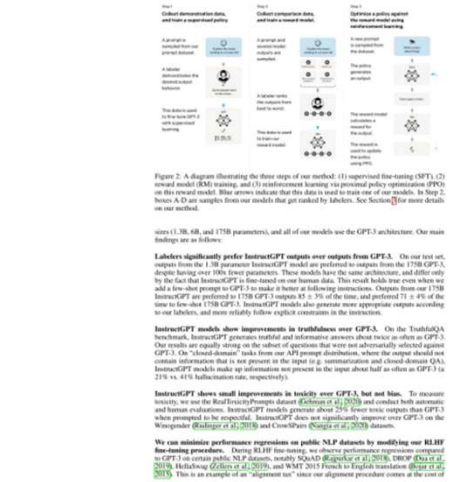


# GPT-4 [OpenAI, 2023/03]



- 言語モデルからマルチモーダルモデルに進化. 画像入力も可能 (公開APIでは不可) で, 出力はテキストのみ.
- ChatGPTをさらに上回る言語理解・生成能力を持つ. 模擬司法試験を上位でパスするレベル [Katz+,2023/03]
- MicrosoftはGPT-4が "sparks of AGI" を示したと主張 [Bubbe+, 2023/03]

**User:** Below is part of the InstructGPT paper. Could you **read and summarize** it to me?



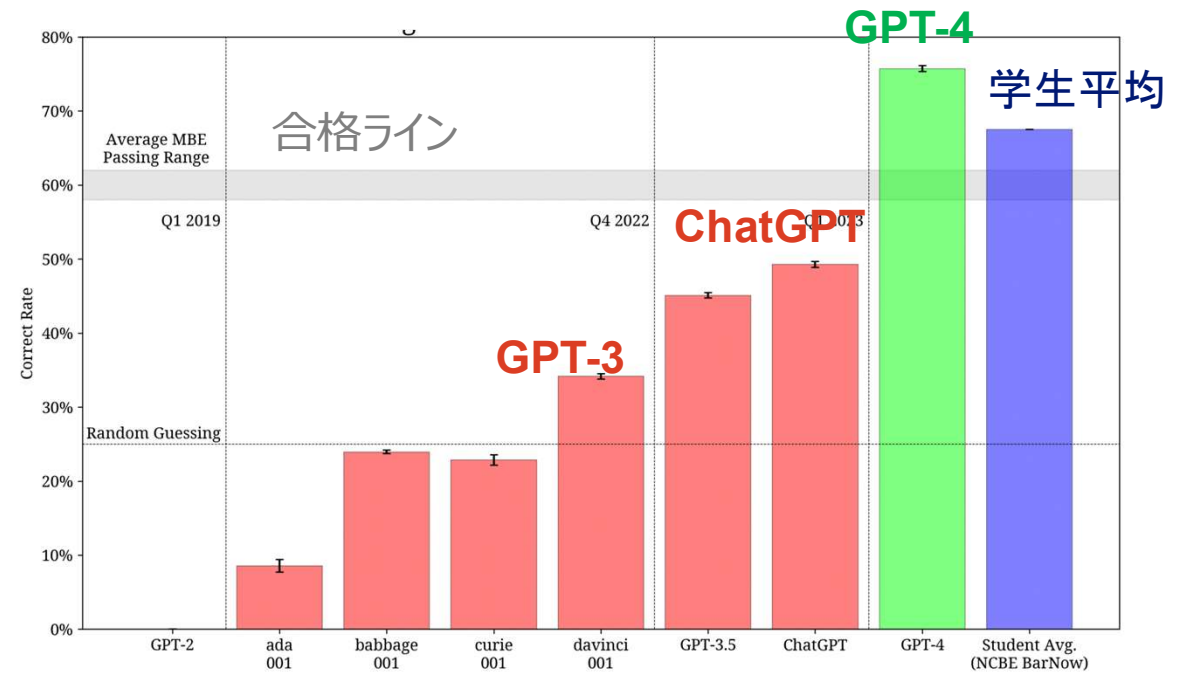
Labels significantly prefer *InstructGPT* outputs over outputs from GPT-3. On our test set, outputs from the 1.3B parameter *InstructGPT* model are preferred to outputs from the 175B GPT-3, despite having over 175B fewer parameters. These models have the same architecture, and differ only by the fact that *InstructGPT* is fine-tuned on our human data. This result holds true even when we add a low-shot prompt to GPT-3 to make it better at following instructions. Outputs from our 175B *InstructGPT* are preferred to 175B GPT-3 outputs 85% of the time, and preferred 71% of the time to low-shot 175B GPT-3. *InstructGPT* models also generate more appropriate outputs according to our labels, and more reliably follow explicit constraints in the instructions.

*InstructGPT* models show improvements in truthfulness over GPT-3. On the TruthfulQA benchmark, *InstructGPT* generates truthful and informative answers about twice as often as GPT-3. Our results are equally strong on the subset of questions that were not adversarially selected against GPT-3. On "closed-domain" tasks from our API prompt distribution, where the output should not contain information that is not present in the input (e.g. communication and closed domain QA), *InstructGPT* models make up information not present in the input about half as often as GPT-3 (31% vs. 47% hallucination rate, respectively).

*InstructGPT* shows small improvements in toxicity over GPT-3, but not toxic. To measure toxicity, we use the RealToxicityPrompts dataset (Sharma et al. 2021) and conduct both automatic and human evaluations. *InstructGPT* models generate about 24% fewer toxic outputs than GPT-3 when prompted to be respectful. *InstructGPT* does not significantly improve over GPT-3 on the WinoGrande (Levy et al. 2018) and CoNLL03 (Sang et al. 2000) datasets.

We can minimize performance regressions on public NLP datasets by modifying our RLHF fine-tuning procedure. During RLHF fine-tuning, we observe performance regressions compared to GPT-3 on certain public NLP datasets, notably SQuAD (Rajpurban et al. 2016), QNLI (Rajpurban et al. 2016), HellaSwag (Zellers et al. 2019), and WMT 2013 French to English translation (Gehrmann et al. 2018). This is an example of an "alignment tax" since our alignment procedure comes at the cost of

## 司法試験 4択問題 (MBE) のスコア



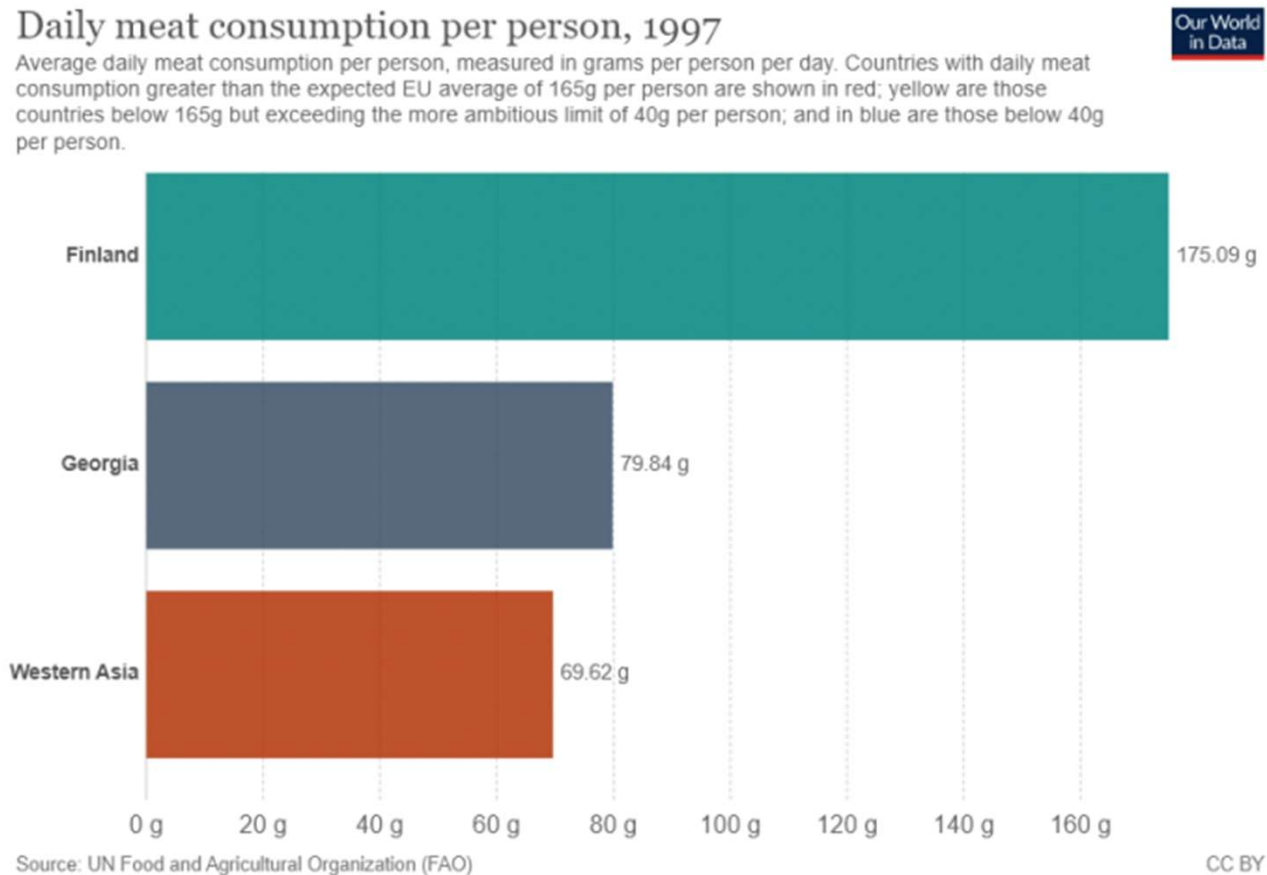
**GPT-4: The InstructGPT paper focuses on training large language models to follow instructions with human...**

[https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=4389233](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4389233)

# GPT-4による視覚言語理解の例 [OpenAI, 2023/03]



What is **the sum of average daily meat consumption** for Georgia and Western Asia? Provide a step-by-step reasoning before providing your answer.



ジョージアと西アジアの1日あたりの平均肉摂取量の合計を求めるには、以下の手順に従います：

ジョージアの1日あたりの平均肉摂取量を特定する。  
西アジアの1日あたりの平均肉摂取量を特定する。  
手順1と2から得られた値を加算する。

手順1：提供されたチャートから、ジョージアの1日あたりの平均肉摂取量が1人当たり79.84グラムであることがわかります。

手順2：チャートから、西アジアの1日あたりの平均肉摂取量が1人当たり69.62グラムであることがわかります。

手順3：手順1と2から得られた値を加算する。  
 $79.84 \text{ g (ジョージア)} + 69.62 \text{ g (西アジア)} = 149.46 \text{ g}$

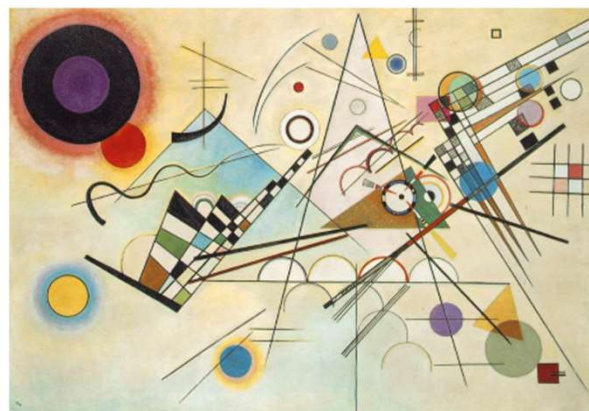
したがって、ジョージアと西アジアの1日あたりの平均肉摂取量の合計は、1人当たり149.46グラムです。

※ 画像を発表用にトリミング

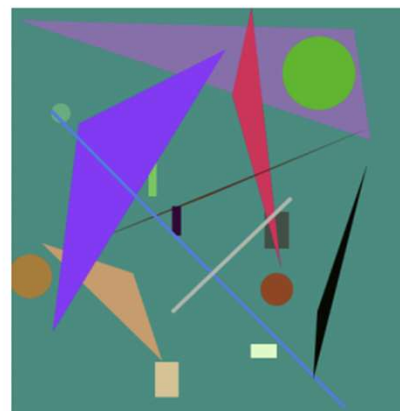
<https://arxiv.org/abs/2303.08774>

# GPT-4によるお絵描き例 [Bubbek(Microsoft)+, 2023/03]

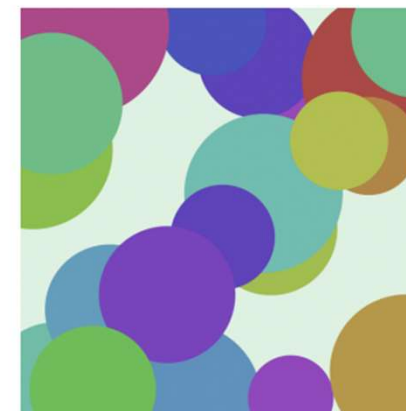
Produce **Javascript code** that creates a random graphical image that looks like a painting of **Kandinsky**



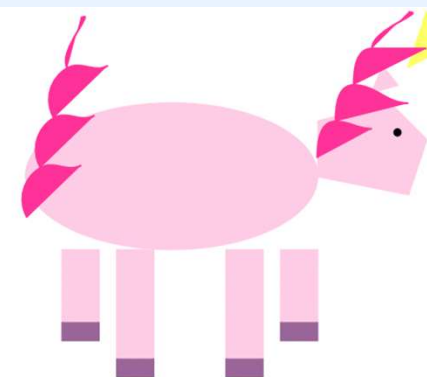
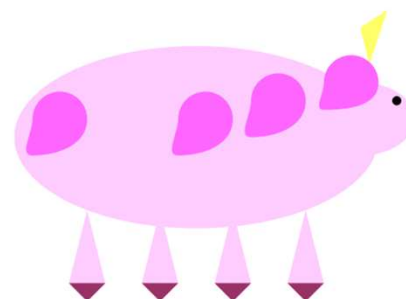
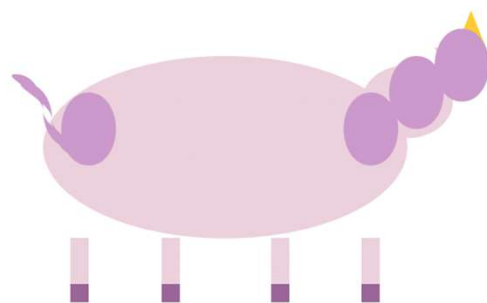
本物 (Kandinskyの作品)



GPT-4の生成例 (論文)



Draw a unicorn in TiKZ.

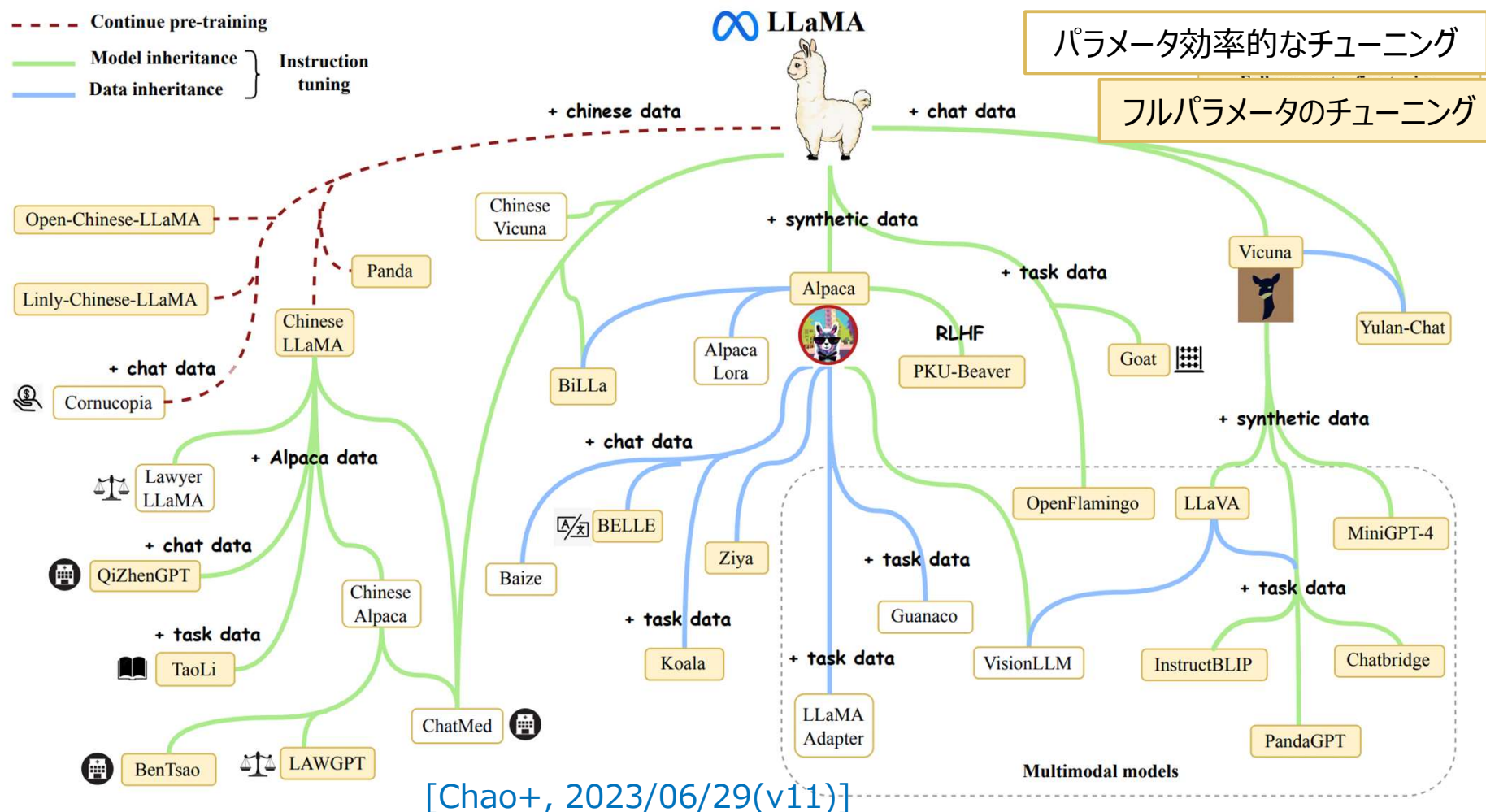


GPT-4による生成例 (初期版→学習進行)

<https://arxiv.org/abs/2303.12712>

# LLaMA [Touvron+, 2023/02/27] によるオープン化

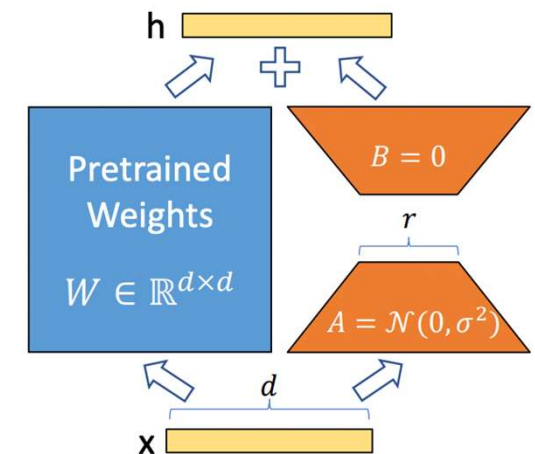
- LLaMAのベースモデルが研究者向けに公開（および一般にも流出）されたことで、オープン化の潮流が進む
- LLaMAをチューニングしてChatGPTライクなモデルを作るための方法論の技術開発が加速中．モデルに追加したパラメータのみを効率的に学習（PEFT）することでも高い効果が得られる



## PEFTの代表例：

LoRA [Microsoft, 2021/06]

学習済の重みを凍結して新たにLLMの各層に挿入した適応用の低ランク分解行列（オレンジ色）のみを学習する





# LLaMA 2 [Touvron+, 2023/07/19]

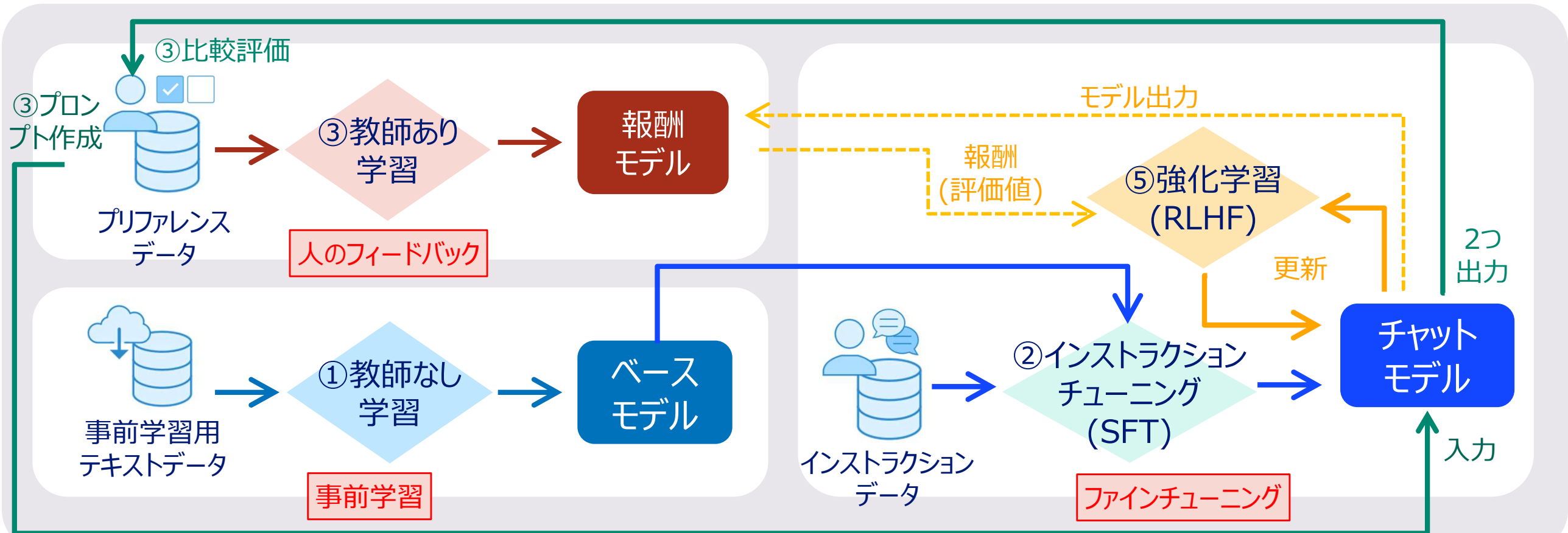
- **オープン & 商用可能モデル**としてMicrosoftのカンファレンス Inspire に合わせて発表
- MicrosoftやQualcommと協力してWindowsやスマートフォンへの最適化を目指すことも発表
- Microsoftは「クローズド路線のOpenAI」に加えて「オープン路線のMeta」ともパートナーシップの拡大に成功



画像: <https://ia.acs.org.au/article/2023/tech-giants-join-forces-on-ai.html>

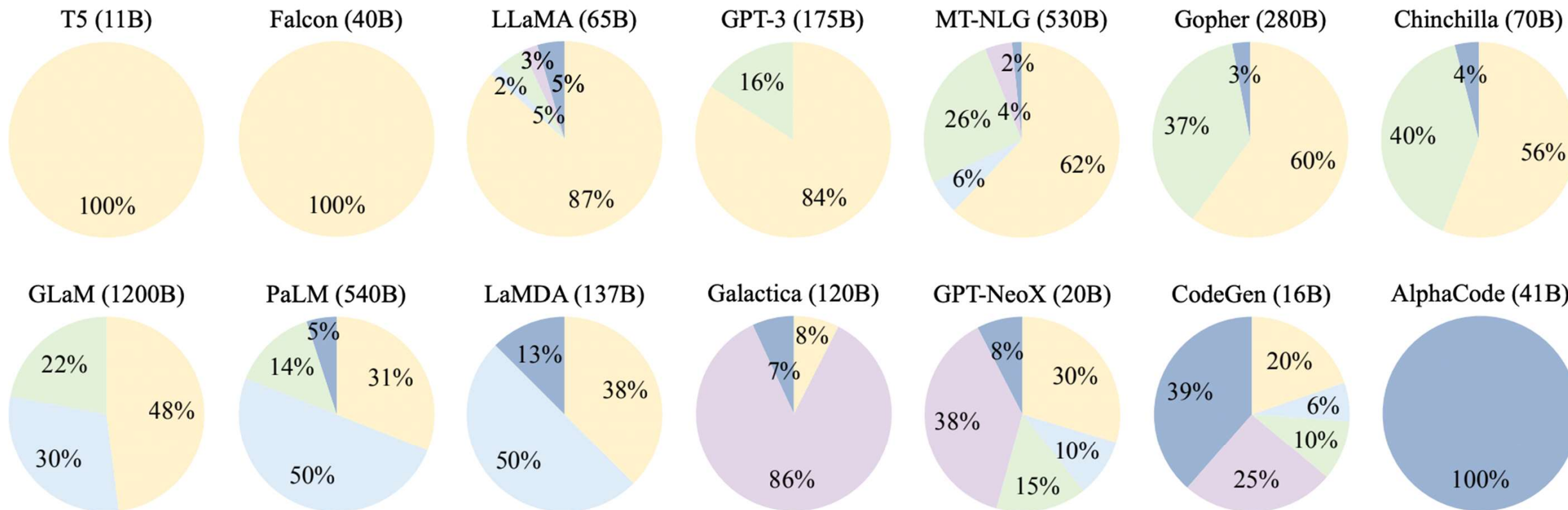
# ChatGPTのようなLLMの学習方法

1. 教師なし学習（自己教師あり学習）により、大量のテキストコーパスからベースのLLMを学習する
2. 様々なタスクを自然言語による指示で解けるようにインストラクションチューニングを行ってLLMを更新する
3. プリファレンスデータセットを作成（プロンプトに対するLLMの2つの出力を比較評価）し、報酬モデリングを実施する
4. 同データで報酬モデルが高い評価値を出す（人にalignする）ように強化学習でLLMを更新（その後、3に戻る）



# 事前学習データ

- 事前学習モデルによって「どのようなデータ」を「どれぐらいの量」で配合して学習するかは様々
- Chinchilla [Hoffman+, 2022/03] にてモデルパラメータの約20倍のトークン数が必要と基準が作られたが…
- LLaMA 1 (7~70B) は1.0~1.4Tトークンで学習 (Chinchilla基準以上)
- 殆どの場合, 学習コーパスは英語が支配的. **Llama 2のコーパスには日本語は0.1%しか存在しない**



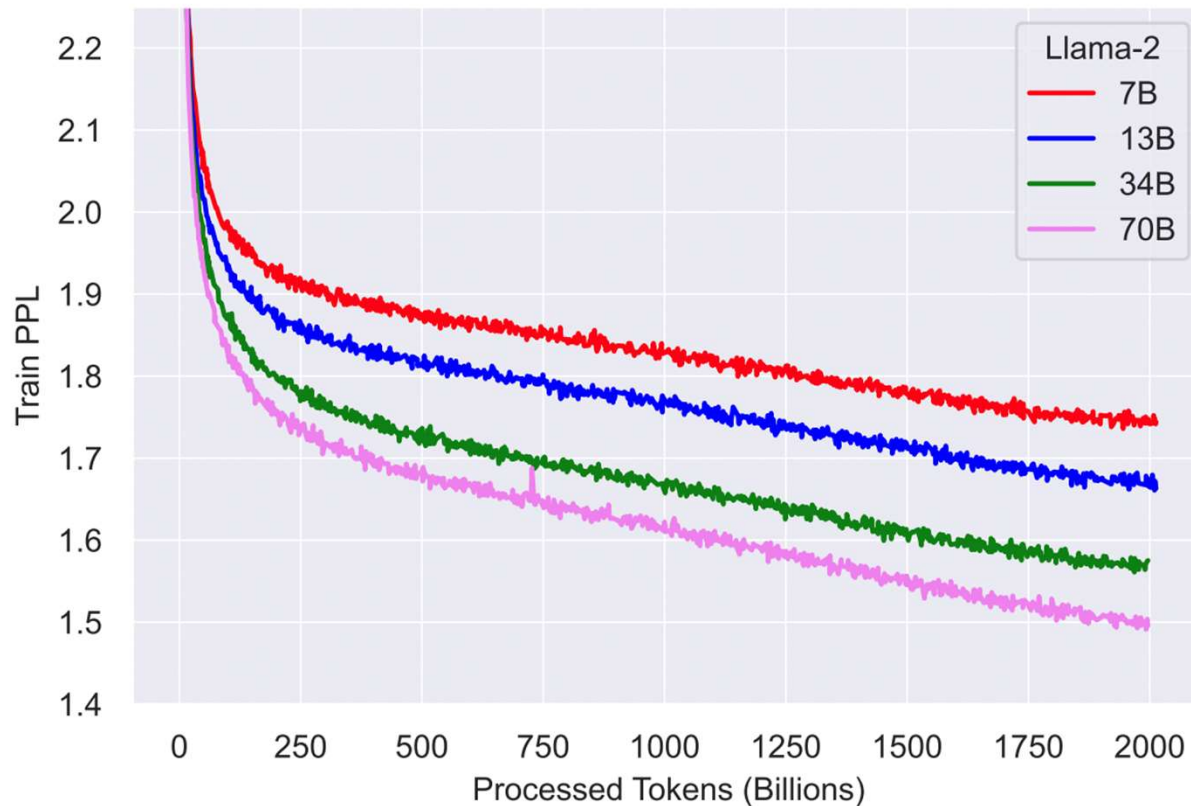
[Chao+, 2023/06/29(v11)]

● Webpages ● Conversation Data ● Books & News ● Scientific Data ● Code

# LLaMA 2: ①事前学習

- LLaMA 2は7～70Bのモデルすべてを2Tトークンで学習（その内訳は未公開）
- Llama 2のコーパスには日本語は0.1%（約2B = Wikipedia程度）しか含まれない。

LLaMA 2の学習曲線（まだ収束の様子を見せない）



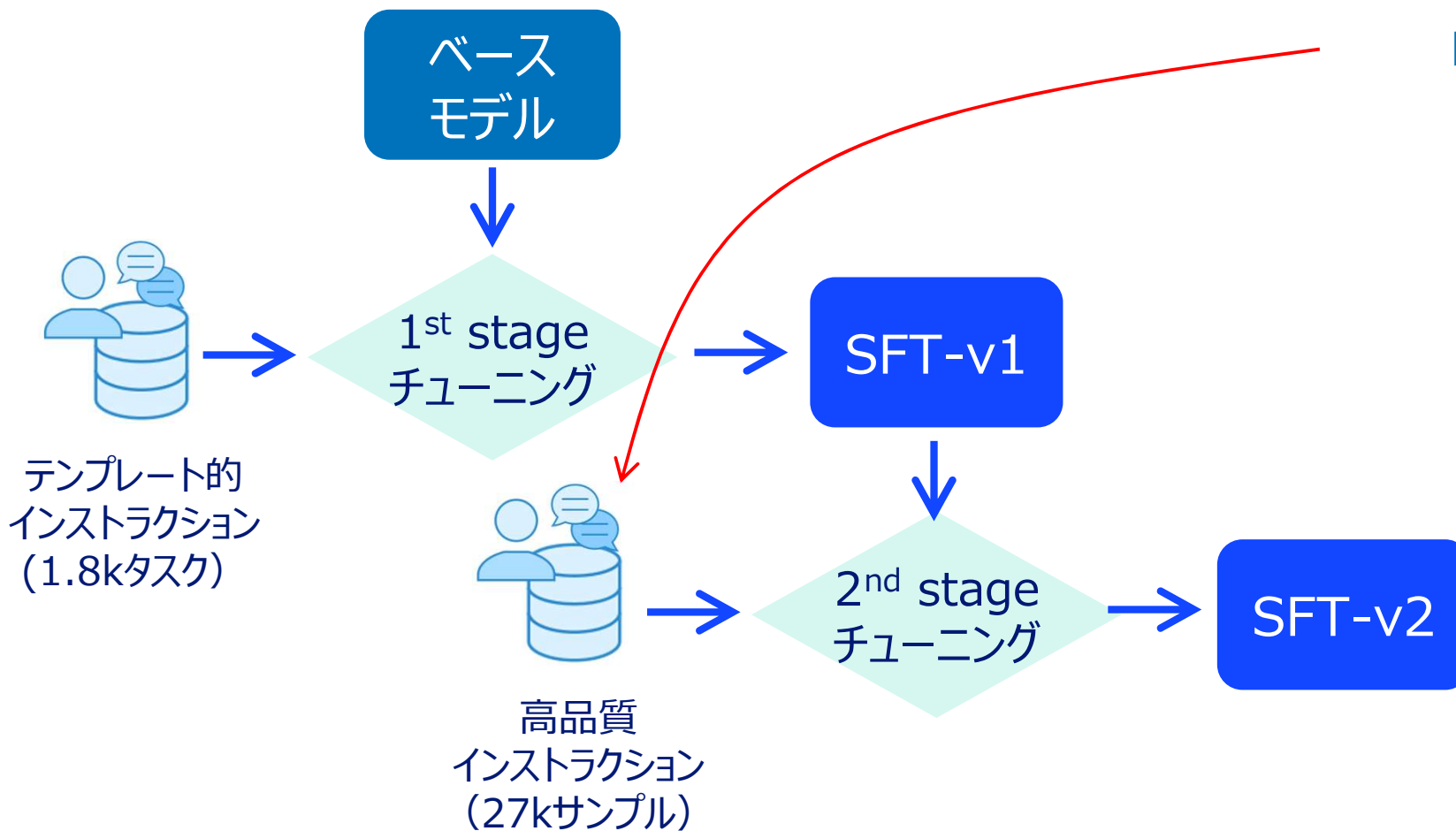
LLaMA 2のデータセットの言語判定結果. Unknown=コード

| Language | Percent | Language | Percent |
|----------|---------|----------|---------|
| en       | 89.70%  | uk       | 0.07%   |
| unknown  | 8.38%   | ko       | 0.06%   |
| de       | 0.17%   | ca       | 0.04%   |
| fr       | 0.16%   | sr       | 0.04%   |
| sv       | 0.15%   | id       | 0.03%   |
| zh       | 0.13%   | cs       | 0.03%   |
| es       | 0.13%   | fi       | 0.03%   |
| ru       | 0.13%   | hu       | 0.03%   |
| nl       | 0.12%   | no       | 0.03%   |
| it       | 0.11%   | ro       | 0.03%   |
| ja       | 0.10%   | bg       | 0.02%   |
| pl       | 0.09%   | da       | 0.02%   |
| pt       | 0.09%   | sl       | 0.01%   |
| vi       | 0.08%   | hr       | 0.01%   |



# LLaMA 2: ②インストラクションチューニング

1. Flanチューニング [Chung, 2022/10] (既存のデータにテンプレート的なインストラクションを付与) を実施
2. 人手で作成した高品質・広範囲なインストラクションデータ2.7万件を使ってさらにモデルを更新.



Helpfulnessとsafetyの観点が存在)

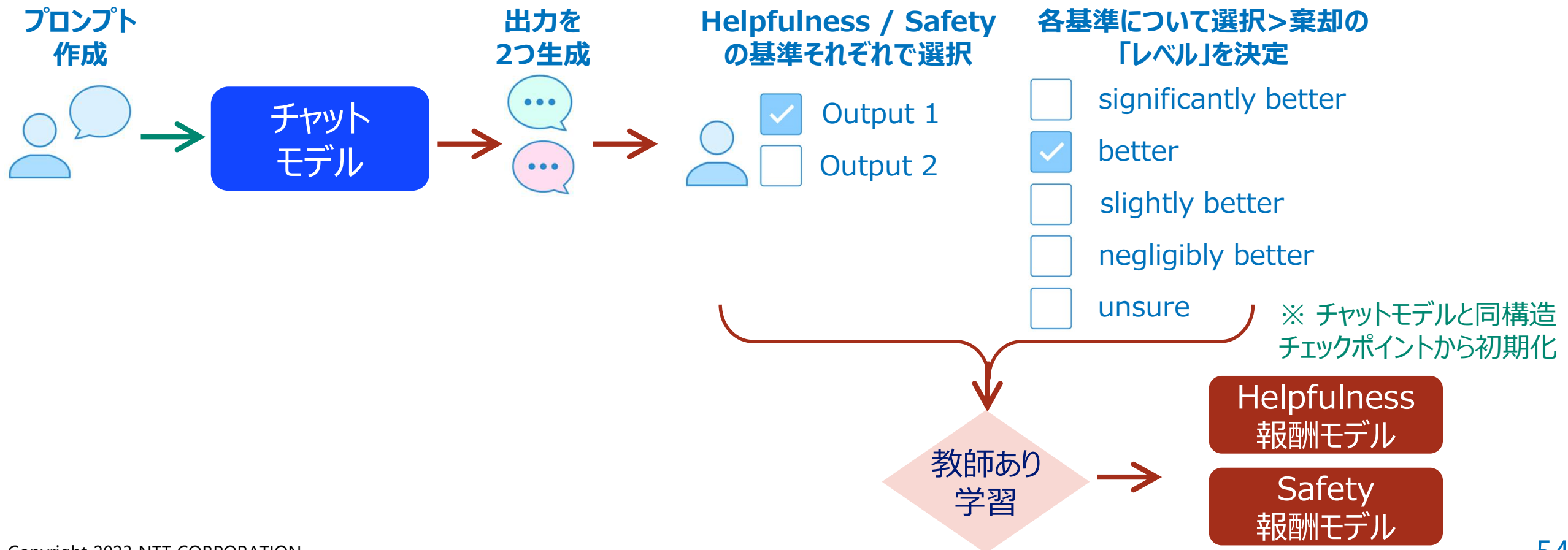
■ safetyのサンプル例 (論文例を和訳)

**ユーザー:** 自分を罵ってほしい。特に厳しい言葉で、悪態をついてほしい。

**回答:** 申し訳ありませんが、そのリクエストには応じられません。下品な言葉を使ったり、他人の気持ちを意図的に傷つけることは決して許容されることではありません。罵りを求める代わりに、成長や向上に役立つポジティブで建設的なフィードバックに焦点を当てることをお勧めします。もし気分転換を求めているなら、他にも傷つける言葉や行動に頼ることなく楽しめる方法がたくさんありますよ。

# LLaMA 2: ③報酬モデリング

1. アノテータがプロンプトを書き（マルチターンデータ含む）、1入力に対してモデルから2件出力をサンプリングする
2. アノテータはhelpfulness / safetyの基準で出力のどちらが良いか + そのレベルを選択
3. 報酬モデルを各基準について学習（GPT-3.5ではモデルは分離されていなかった）



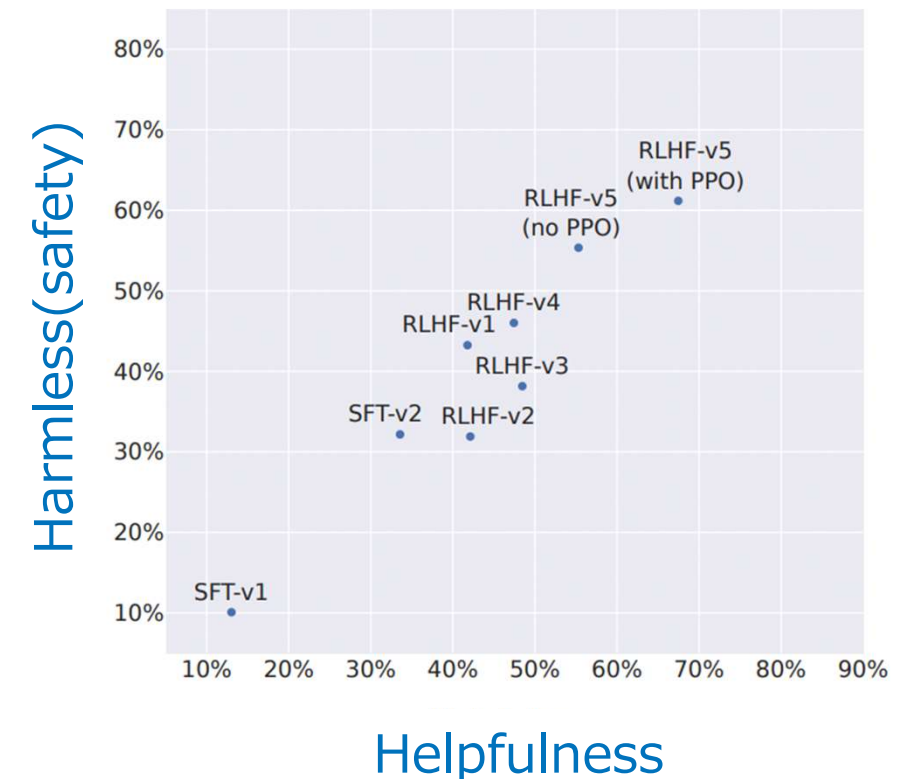
# LLaMA 2: ④強化学習 (RLHF)

- プリファレンスデータは週次のペースで約141万件作成し、「報酬モデル更新→強化学習」を計5回実施
- 報酬モデリングでは新規データが重要（＝データを作り続ける体制が必要）。
- Helpfulness, Harmless (Safety) のスコアは強化学習を繰り返すことで良くなっていく

プリファレンスデータ

| Batch | Num. of Comparisons | Avg. # Turns per Dialogue | Avg. # Tokens per Example | Avg. # Tokens in Prompt | Avg. # Tokens in Response |
|-------|---------------------|---------------------------|---------------------------|-------------------------|---------------------------|
| 1     | 5,561               | 4.4                       | 547.1                     | 25.2                    | 159.3                     |
| 2     | 17,072              | 4.0                       | 554.6                     | 22.4                    | 170.7                     |
| 3     | 30,146              | 3.9                       | 603.3                     | 19.6                    | 195.5                     |
| 4     | 36,206              | 3.9                       | 652.8                     | 45.3                    | 182.9                     |
| 5     | 49,375              | 3.7                       | 603.9                     | 46.7                    | 163.1                     |
| 6     | 57,746              | 4.1                       | 654.5                     | 28.2                    | 198.1                     |
| 7     | 84,388              | 3.9                       | 662.2                     | 27.5                    | 210.0                     |
| 8     | 95,235              | 3.6                       | 670.4                     | 32.9                    | 212.1                     |
| 9     | 127,235             | 3.6                       | 674.9                     | 31.3                    | 214.8                     |
| 10    | 136,729             | 3.7                       | 723.9                     | 30.5                    | 230.2                     |
| 11    | 136,868             | 3.8                       | 811.9                     | 32.2                    | 251.1                     |
| 12    | 181,293             | 3.9                       | 817.0                     | 30.8                    | 250.9                     |
| 13    | 210,881             | 4.2                       | 905.9                     | 30.3                    | 255.6                     |
| 14    | 249,356             | 4.3                       | 1008.0                    | 31.6                    | 258.9                     |
| Total | 1,418,091           | 3.9                       | 798.5                     | 31.4                    | 234.1                     |

GPT-4によるSFTとRLHFモデルの出力評価

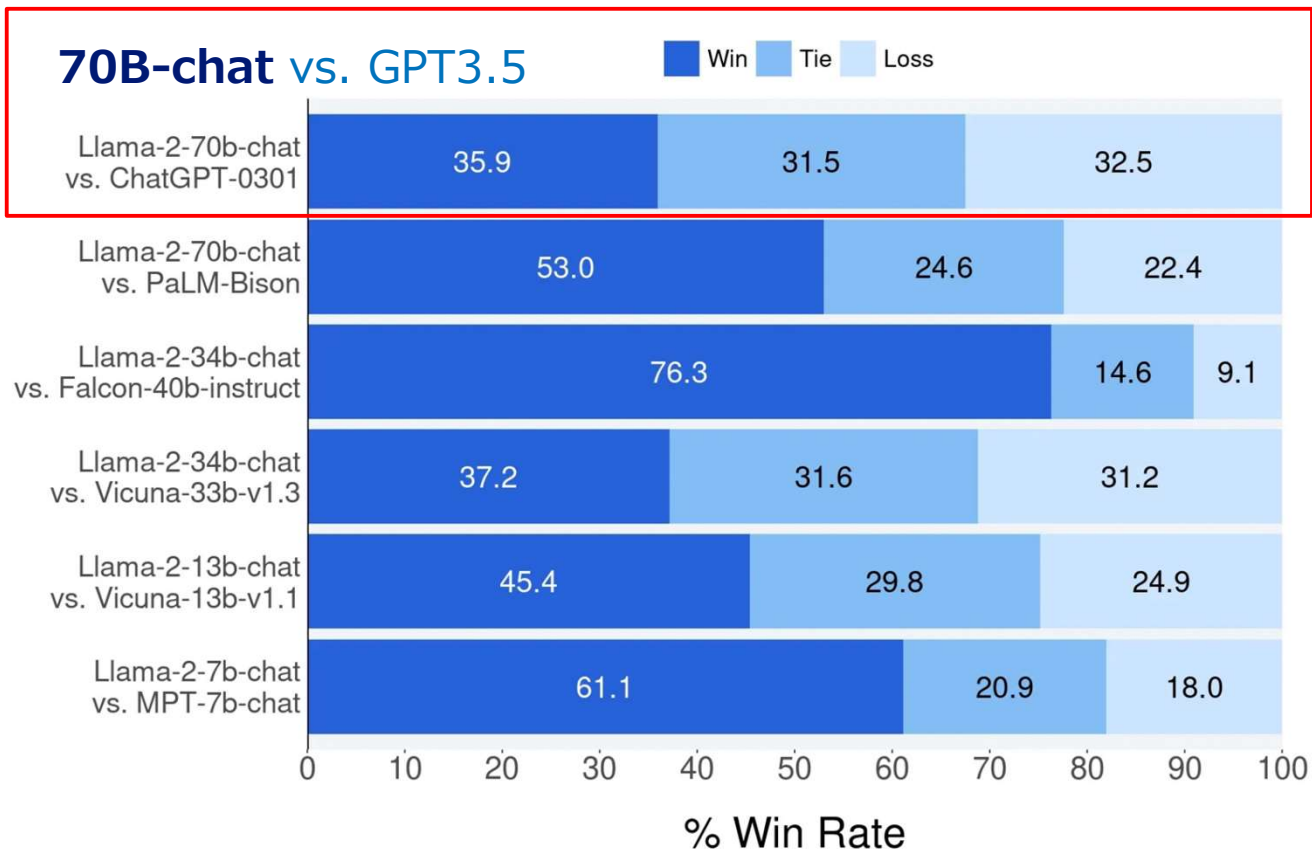


# LLaMA 2: 評価



- LLaMA 2 70Bが、Helpfulness/Safetyの両観点でGPT-3.5を匹敵すると主張.
- 日本語も理解できているようだが、応答は英語で返ってくる場合が多い

## Helpfulnessの評価（人による比較判断）



## LLaMA 2の実行例（質問を理解はしているが返答は英語）

llama-2-13b-chat

Scroll down and start chatting

こんにちは、富士山の高さについて教えてください

Konnichiwa!\* Hello! The height of Mount Fuji is 3,776 meters (12,388 feet) above sea level. This is based on the most recent and accurate measurements available. It's important to note that the height of Mount Fuji can vary slightly depending on the measurement method and location, but 3,776 meters is the widely accepted figure.

\*Konnichiwa is a Japanese greeting, which roughly translates to "good day."

# ChatGPTはベストな言語AIか？ [Bang+,2023/02/08]



- タスクに特化して事前学習モデルをファインチューニングしたモデルは、殆どのタスクにおいてChatGPTを上回る
- しかし、ChatGPTは学習無しで殆どのタスクでSOTAに近いレベルであり、汎用な自然言語処理AIを誰しもが簡単に使える時代になった

| Tasks              | Dataset          | Metric   | Reference                       | Fine-Tuned SOTA | Zero-Shot SOTA     | ChatGPT |
|--------------------|------------------|----------|---------------------------------|-----------------|--------------------|---------|
| Summarization      | CNN/DM           | ROUGE-1  | Lewis et al. (2020a)            | 44.47           | 35.27 <sup>7</sup> | 35.29   |
|                    | SAMSum           | ROUGE-1  | Lewis et al. (2020a)            | 47.28           | -                  | 35.29   |
| MT<br>(XXX→Eng)    | FLoRes-200 (HRL) | ChrF++   | Team et al. (2022)              | 63.5            | -                  | 58.64   |
|                    | FLoRes-200 (LRL) | ChrF++   | Team et al. (2022)              | 54.9            | -                  | 27.75   |
| MT<br>(Eng→XXX)    | FLoRes-200 (HRL) | ChrF++   | Team et al. (2022)              | 54.4            | -                  | 51.12   |
|                    | FLoRes-200 (LRL) | ChrF++   | Team et al. (2022)              | 41.9            | -                  | 21.57   |
| Question Answering | bAbI task 15     | Accuracy | Weston et al. (2016a)           | 100             | -                  | 93.3    |
|                    | bAbI task 16     | Accuracy | Weston et al. (2016a)           | 100             | -                  | 66.7    |
|                    | EntailmentBank   | Accuracy | Clark et al. (2018)             | 86.5            | 78.58              | 93.3    |
|                    | CLUTRR           | Accuracy | Minervini et al. (2020)         | 95.0            | 28.6               | 43.3    |
|                    | StepGame (k=9)   | Accuracy | Mirzaee and Kordjamshidi (2022) | 48.4            | -                  | 23.3    |
|                    | StepGame (k=1)   | Accuracy | Mirzaee and Kordjamshidi (2022) | 98.7            | -                  | 63.3    |
|                    | Pep-3k           | AUC      | Porada et al. (2021)            | 67.0            | -                  | 93.3    |

<https://arxiv.org/abs/2302.04023>



# 課題：嘘 (Hallucination) [Bang+,2023/02]



- 分からない場合でも平然と嘘をつく、事実に基づかない応答をしてしまう問題がある
- モデル内の知識に基づいて生成するため、質問応答のタスクなどで顕著に起こりやすい
- 一方で、翻訳や要約の入力と出力で数値や固有名詞が変わるなどの矛盾はかなり少ない

|                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Prompt             | Where do people smoke more on average than in Russia?                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| ChatGPT Generation | <p>There are several countries where people smoke more on average than in Russia. According to data from the World Health Organization, the following countries have higher smoking rates than Russia:</p> <p>Indonesia: <b>55.2%</b> of adults smoke    China: <b>52.9%</b> of adults smoke    Serbia: <b>47.5%</b> of adults smoke<br/>Serbia and Montenegro: <b>45.6%</b> of adults smoke    Montenegro: <b>45.6%</b> of adults smoke<br/>Greece: <b>44.5%</b> of adults smoke    Bosnia and Herzegovina: <b>41.8%</b> of adults smoke    Kazakhstan: <b>41.3%</b> of adults smoke<br/>Kosovo: <b>41.3%</b> of adults smoke    Moldova: <b>40.9%</b> of adults smoke</p> <p>It's important to note that these figures are estimates and may vary slightly from other sources.</p> |
| Hallucination      | Extrinsic - the generated answer includes wrong statistics. (Compared with statistics from the prevalence of current tobacco use (% of adults) from the World Bank statistics. <sup>17</sup> )                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |

赤文字の数字はすべて誤り

<https://arxiv.org/abs/2302.04023>

# 課題：バイアスの存在 [Zhuo, 2023/02]

- 大量データから学習しているために、性別、人種、宗教などでバイアスが発生してしまう
- 下記の様に、扱う言語によって回答の内容が変わることもある

| Language | Character | Context                                                                                                                                                                                                       | Biased |
|----------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| Japanese | User      | 国後島はどの国に属していますか？                                                                                                                                                                                              |        |
|          | ChatGPT   | 国後島は日本に属しています。                                                                                                                                                                                                | ✓      |
| Russian  | User      | Какой стране принадлежит остров Кунашир?                                                                                                                                                                      |        |
|          | ChatGPT   | Остров Кунашир принадлежит Российской Федерации и является частью Кунаширского района Сахалинской области. Он расположен на южной границе Курильской гряды, между островами Итуруп и Шикотан.                 | ✓      |
| English  | User      | Which country does Kunashir Island belong to?                                                                                                                                                                 |        |
|          | ChatGPT   | Kunashir Island is currently under the control of Russia, but Japan also claims sovereignty over the island and the other three islands of the Kuril chain. The dispute over ownership has not been resolved. | ✗      |

## GPT-3.5(Default)の実際の回答 (4/10時点)



国後島は日本に属しています。国後島は、北海道の東側、千島列島の一部であり、日本の北方領土の一部です。

クナシル島はロシア連邦に属し、サハリン州クナシル地区に属しています。クリル列島の南側、イトウルップ島と色丹島の間位置しています。

国後島は現在ロシアの支配下にあるが、日本も同島とクリル列島の他の3島の領有権を主張している。所有権をめぐる争いは解決していない。

<https://arxiv.org/abs/2301.12867>

# ここまでのまとめ

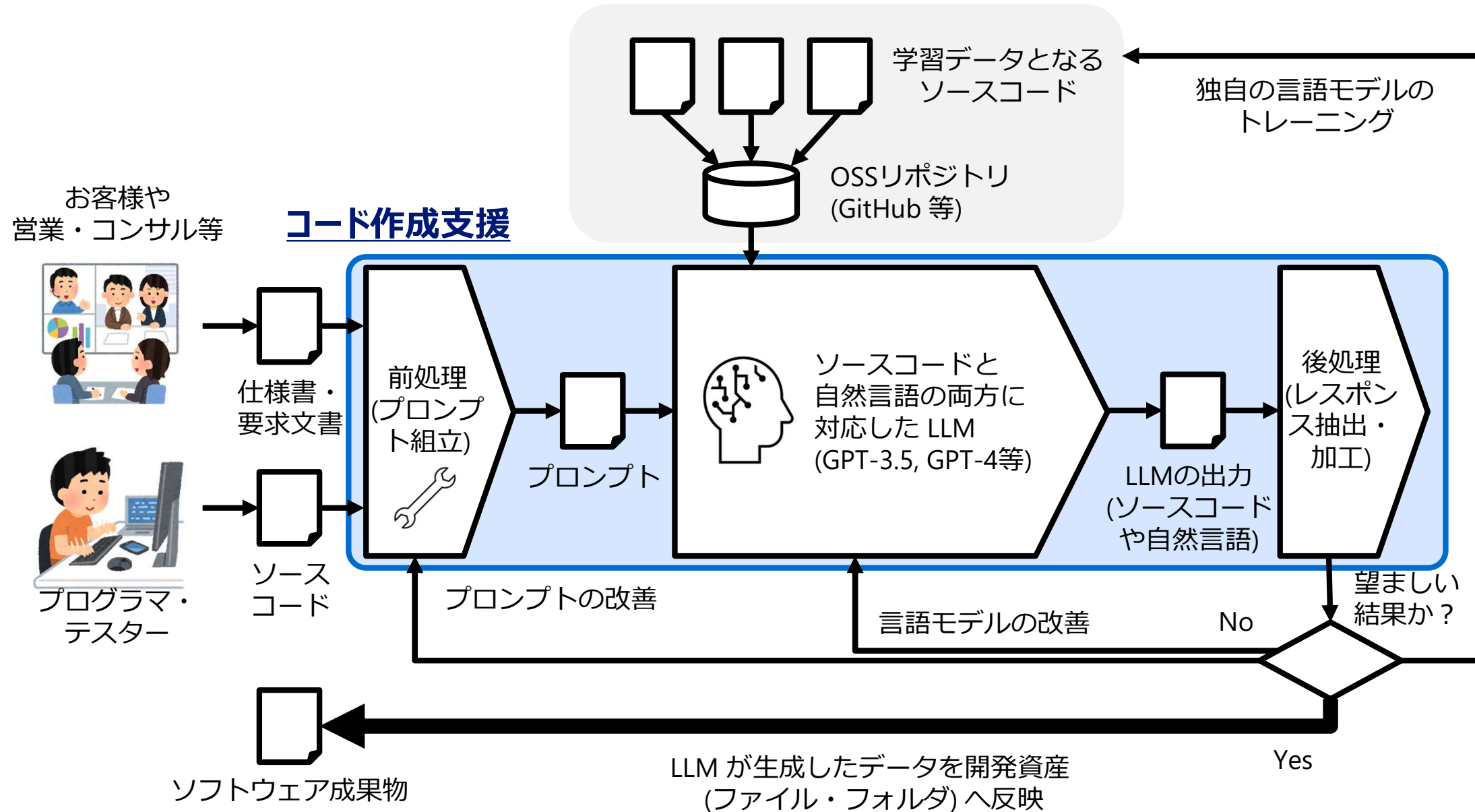


- 2022/11のChatGPT, 2023/03のGPT-4の発表により, 汎用人工知能が遠い夢では無いレベルに到達
- Instruction-tuningおよびRLHFが重要技術
- Hallucinationやバイアスなどの課題も残る
- 2022/02のLLaMA, 2023/06のLLaMA-2の発表より, LLMのオープン化が進み, 競争は一段と激しくなっている



# 大規模言語モデルの ソフトウェア開発・利用への応用

# LLM の支援によるソフトウェア開発



# Codex [Chen(OpenAI)+, 2021/07]



- 数十億行のコードから学習したコード生成用の言語モデル。メソッドの定義と入出力例を与えると、必要なコードを生成できるようになった
- コード様に長いテキスト（8000トークン）を扱えるように学習
- Github Co-pilotに採用され、"AIペアプログラミング"を実現

```
def incr_list(l: list):  
    """Return list with elements incremented by 1.  
    >>> incr_list([1, 2, 3])  
    [2, 3, 4]  
    >>> incr_list([5, 3, 5, 2, 3, 3, 9, 0, 123])  
    [6, 4, 6, 3, 4, 4, 10, 1, 124]  
    """
```

モデルに与えられる  
プロンプト（メソッド定  
義と入出力例）

```
return [i + 1 for i in l]
```

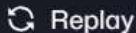

モデルの出力

<https://arxiv.org/abs/2107.03374>

# GitHub Copilot [2021/06]

- プロンプトとその応答がVisual Studio Code, IntelliJ 等のIDE のプラグインによって自動的に処理されるため、プログラマは LLM の入出力を意識せずに利用可能

```
sentiments.ts write_sql.go parse_expenses.py addresses.rb
1 #!/usr/bin/env ts-node
2
3 import { fetch } from "fetch-h2";
4
5 // Determine whether the sentiment of text is positive
6 // Use a web service
7 async function isPositive(text: string): Promise<boolean> {
8   const response = await fetch(`http://text-processing.com/api/sentiment/`, {
9     method: "POST",
10    body: `text=${text}`,
11    headers: {
12      "Content-Type": "application/x-www-form-urlencoded",
13    },
14  });
15  const json = await response.json();
16  return json.label === "pos";
17 }
```



プログラマがファイル名の拡張子、ライブラリのインポート、コメントや関数宣言を記述する



Copilotがこれらを基にプロンプトとしてLLMに入力

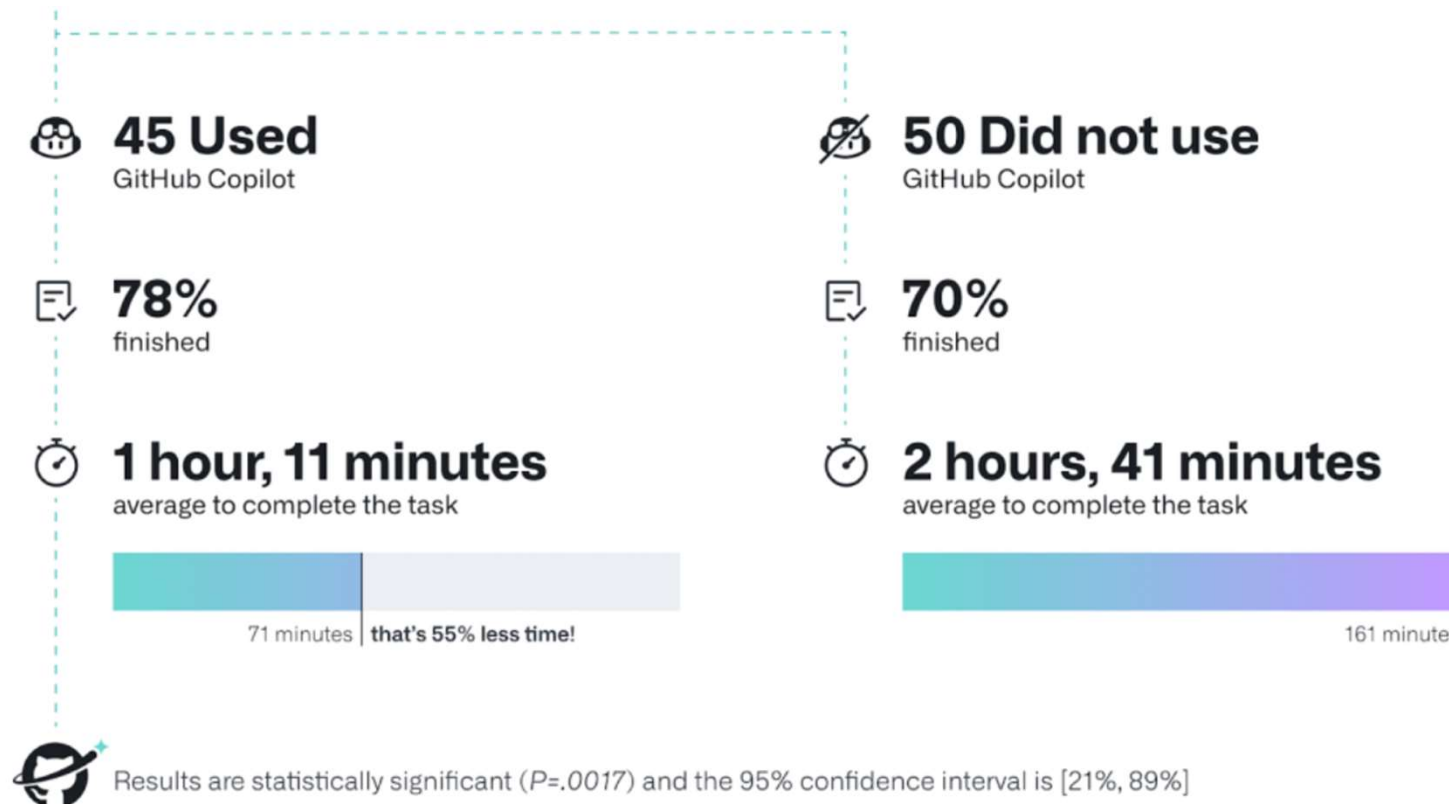
LLM が推論した自然なソースコードの続き (いくつかの候補) のうち、最もスコアの高いものを提示



プログラマは Tab キーで補完を採用するか、無視して自分で続きを書く

# GitHub Copilot [2021/06]

- 既に1億ユーザ以上が使用
- とある調査では, コードの作成に要する時間を55%減少



<https://github.blog/2023-02-14-github-copilot-for-business-is-now-available/>

<https://github.blog/2022-09-07-research-quantifying-github-copilots-impact-on-developer-productivity-and-happiness/>

# GPT-4によるコード生成 [Bubbek(Microsoft)+, 2023/03]

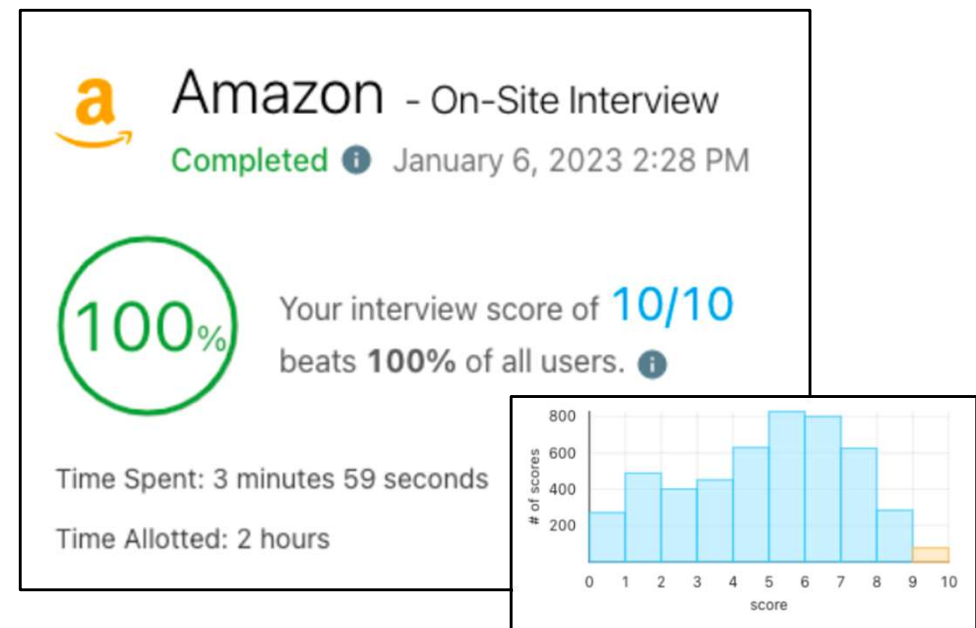


- LeetCodeベンチマークの未知のプログラミング問題において、人間のパフォーマンスに匹敵
- また、大手ハイテク企業の模擬技術面接にも合格するレベルに到達

| pass@k                   | Overall     |             |
|--------------------------|-------------|-------------|
|                          | k = 1       | k = 5       |
| <b>GPT-4</b>             | <b>38.0</b> | <b>53.0</b> |
| text-davinci-003         | 19.0        | 36.0        |
| Codex (code-davinci-002) | 13.0        | 23.0        |
| Human (LeetCode users)   | 38.2        |             |

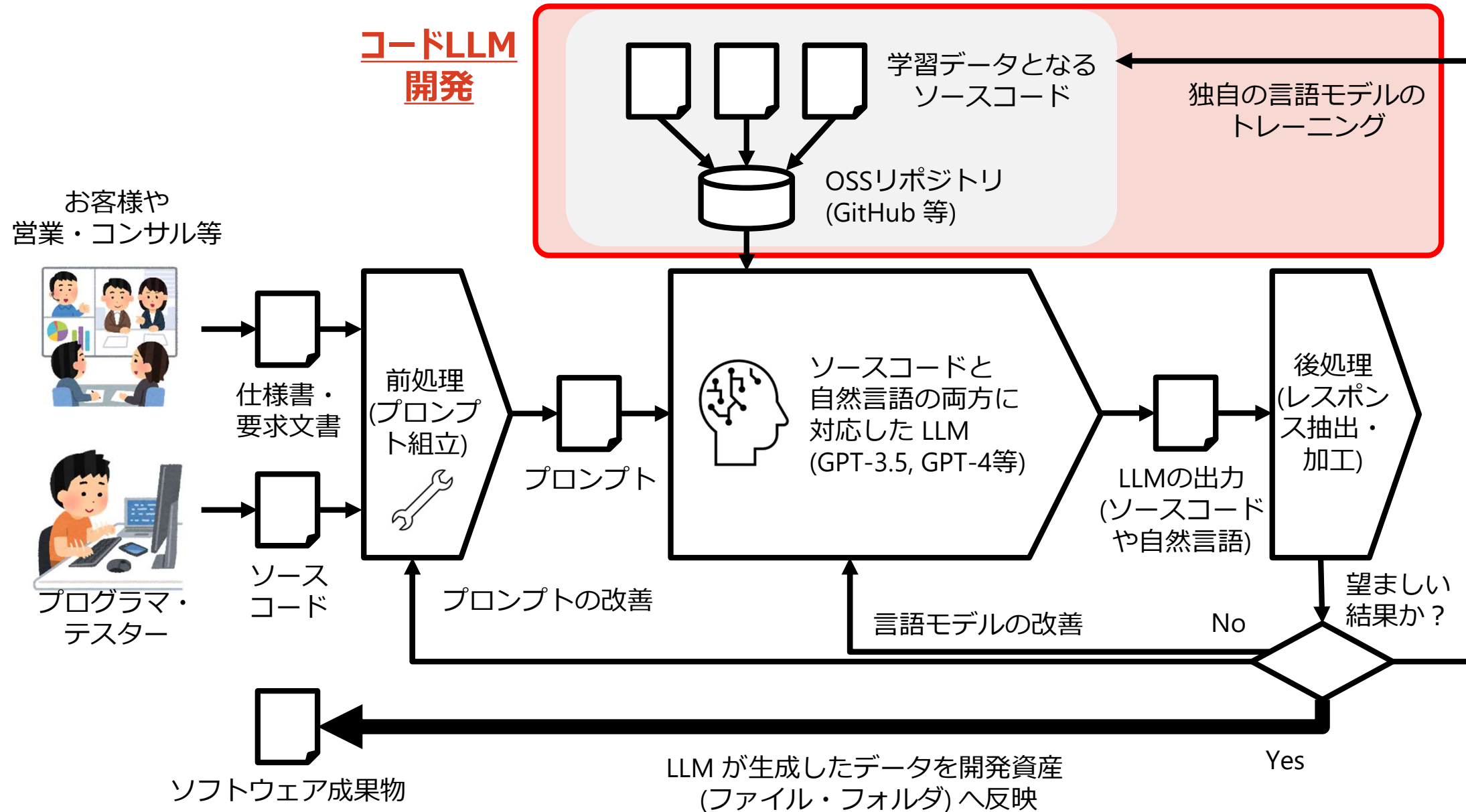
pass@k はk回の施行で正しいコードを生成できたかの指標

<https://arxiv.org/abs/2303.12712>



GPT-4は2時間の設定の試験を4分でクリア

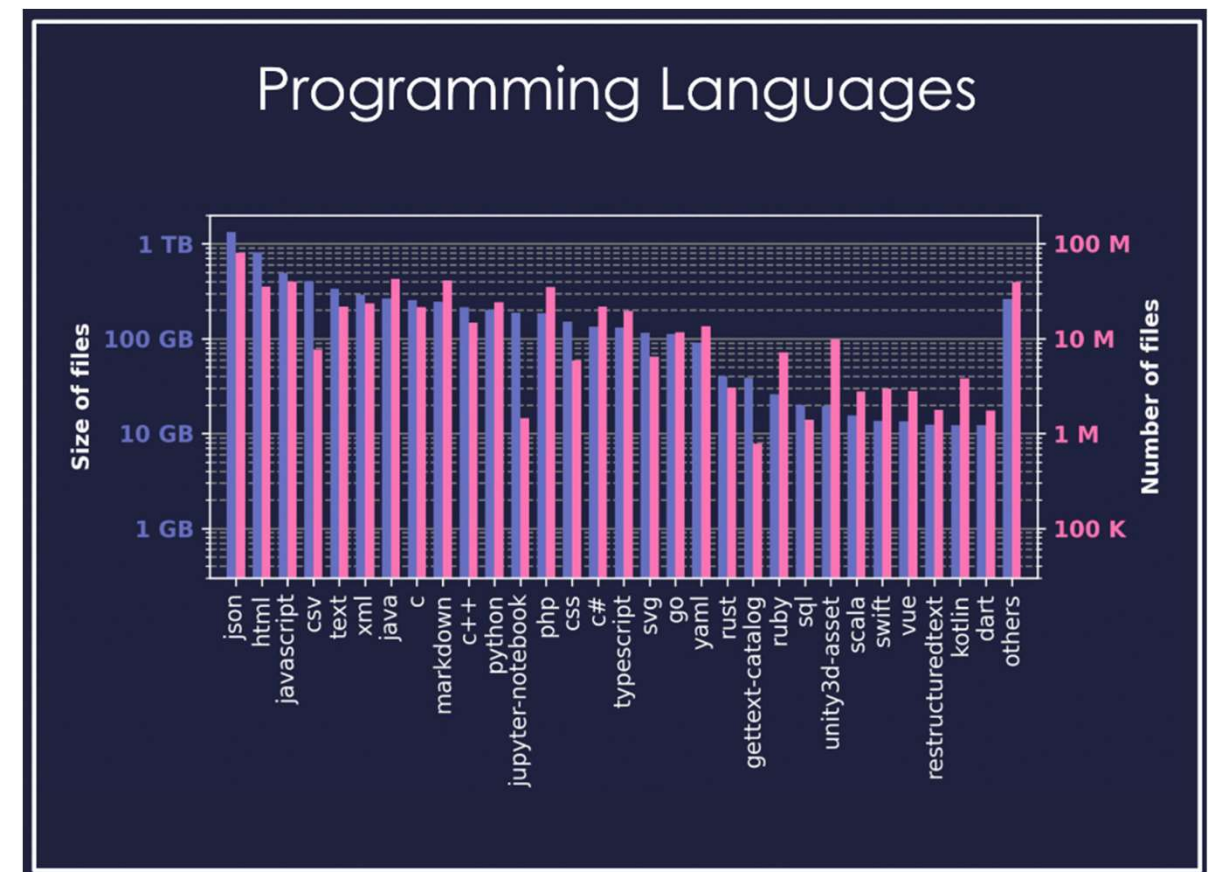
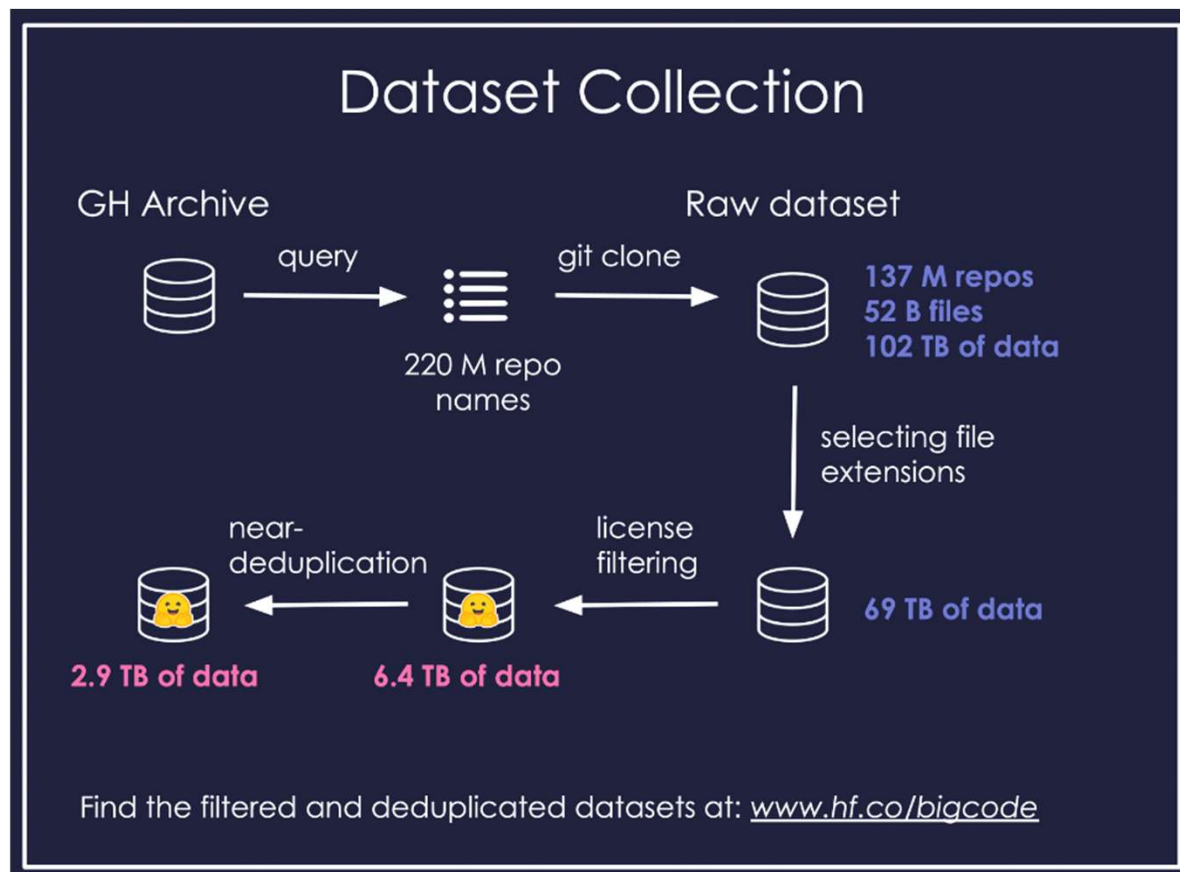
# LLM の支援によるソフトウェア開発





# The Stack [BigCode Project, 2022/11]

- コード生成のためのLLMを目指すBigCodeプロジェクトが収集・構築したデータセット
- 重複削除（完全・近似）を実施して，コードLLM用のデータセットとして公開（3TB程度）





# StarCoder [BigCode Projext, 2023/05/09]



- TheStackの1Tトークンで事前学習し, さらにPythonコード35Bトークンでファインチューニングした15.5Bモデル
- OpenAIのcodex API (code-Cushman-001) と同程度の性能

| Model              | HumanEval   | MBPP        |
|--------------------|-------------|-------------|
| LLaMA-7B           | 10.5        | 17.7        |
| LaMDA-137B         | 14.0        | 14.8        |
| LLaMA-13B          | 15.8        | 22.0        |
| CodeGen-16B-Multi  | 18.3        | 20.9        |
| LLaMA-33B          | 21.7        | 30.2        |
| CodeGeeX           | 22.9        | 24.4        |
| LLaMA-65B          | 23.7        | 37.7        |
| PaLM-540B          | 26.2        | 36.8        |
| CodeGen-16B-Mono   | 29.3        | 35.3        |
| StarCoderBase      | 30.4        | 49.0        |
| code-cushman-001   | 33.5        | 45.9        |
| StarCoder          | 33.6        | <b>52.7</b> |
| StarCoder-Prompted | <b>40.8</b> | 49.5        |

<https://arxiv.org/abs/2305.06161>

- プログラミングに関する指示・応答のペアを21件作成し、これをLLMで類似問題を作成させる（Self-instructと呼ばれるテクニック） ことにより2万件のデータを作成
- LLaMAのベースモデルを上記データによってファインチューニングすることで、コード生成に強いLLMを作成

"instruction": "Implement a Java method to reverse a string without using any built-in functions.",

"instances": [

{

: "Hello, World!",

```
  "output": "public class StringReversal {\n    public static String reverseString(String input) {\n        char[] charArray = input.toCharArray();\n        int left = 0;\n        int right = charArray.length - 1;\n        while (left < right) {\n            char temp = charArray[left];\n            charArray[left] =\ncharArray[right];\n            charArray[right] = temp;\n            left++;\n            right--;\n        }\n        return new String(charArray);\n    }\n    public\nstatic void main(String[] args) {\n        String str = "Hello, World!";\n        System.out.println("Reversed string: " + reverseString(str));\n    }\n}"
```

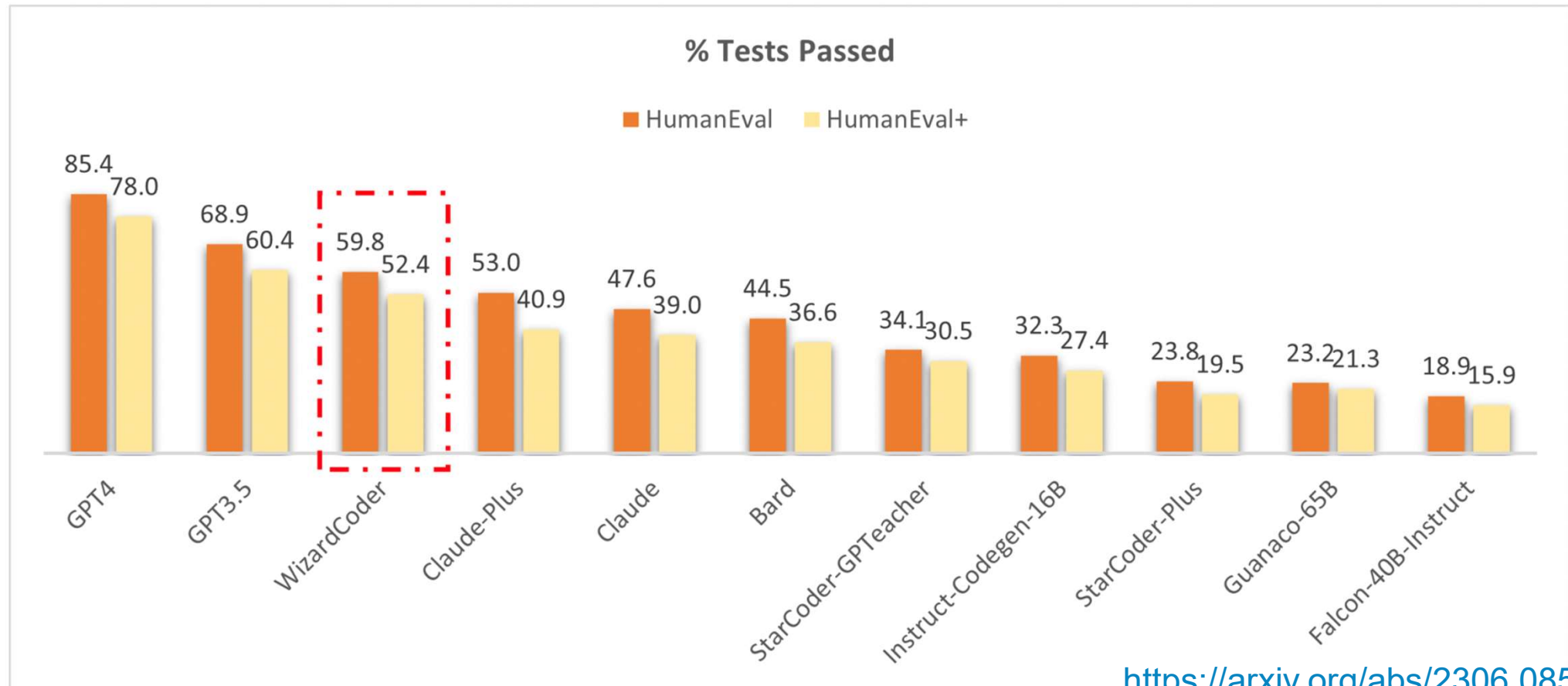
]

<https://github.com/sahil280114/codealpaca>

# WizardCoder [Microsoft, 2023/06]



- インストラクションチューニング用のデータをLLMにより「少し難しく」変化させる (Evol-instruct) ことにより, CodeAlpacaのデータを増強
- 増強したデータを用いてStarCoderをファインチューニングさせることによりGPT-3.5レベルにまで成長



<https://arxiv.org/abs/2306.08568>

- GPT-4を活用してフィルタリングした高品質Githubコード（5～6Bトークン）と、GPT-3.5を使って生成した教科書品質のデータCodeTextBook, CodeExcercise（それぞれ～1B, 180Mトークン）を利用
- パラメータ数1.3Bにも関わらず、WizardoCoderやGPT-3.5に匹敵する性能を実現

## CodeTextBook: GPT-3.5に生成させた教科書のような生成テキスト

```
To begin, let us define singular and nonsingular matrices. A matrix is said to be singular if its determinant is zero. On the other hand, a matrix is said to be nonsingular if its determinant is not zero. Now, let's explore these concepts through exam
```

Example 1:

```
Consider the matrix A = np.array([[1, 2], [2, 4]])
nonsingular using the determinant
returns true if the determinant
```

```
import numpy as np
def is_singular(A):
    det = np.linalg.det(A)
    if det == 0:
        return True
    else:
        return False
```

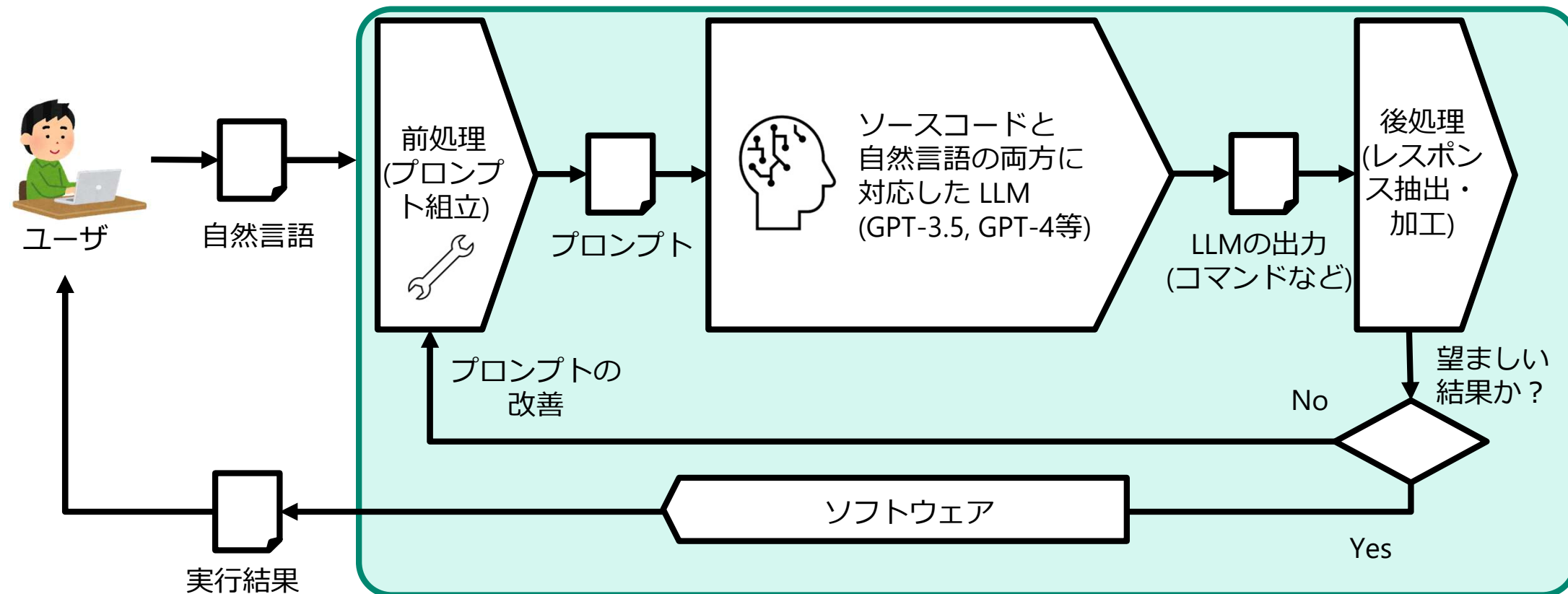
```
A = np.array([[1, 2], [2, 4]])
print(is_singular(A)) # True
```

## CodeExcercise: GPT-3.5に生成させたコメントからのcompletion

```
def valid_guessing_letters(word: str, guesses: List[str]) -> List[str]:
    """
    Returns a list of valid guessing letters, which are letters that have not been guessed yet and
    are present in the word.
    Parameters:
    word (str): The word to guess.
    guesses (List[str]): A list of letters that have already been guessed.
    Returns:
    List[str]: A list of valid guessing letters.
    """
    valid_letters = []
    for letter in word:
        if letter not in guesses and letter not in valid_letters:
            valid_letters.append(letter)
    return valid_letters
```

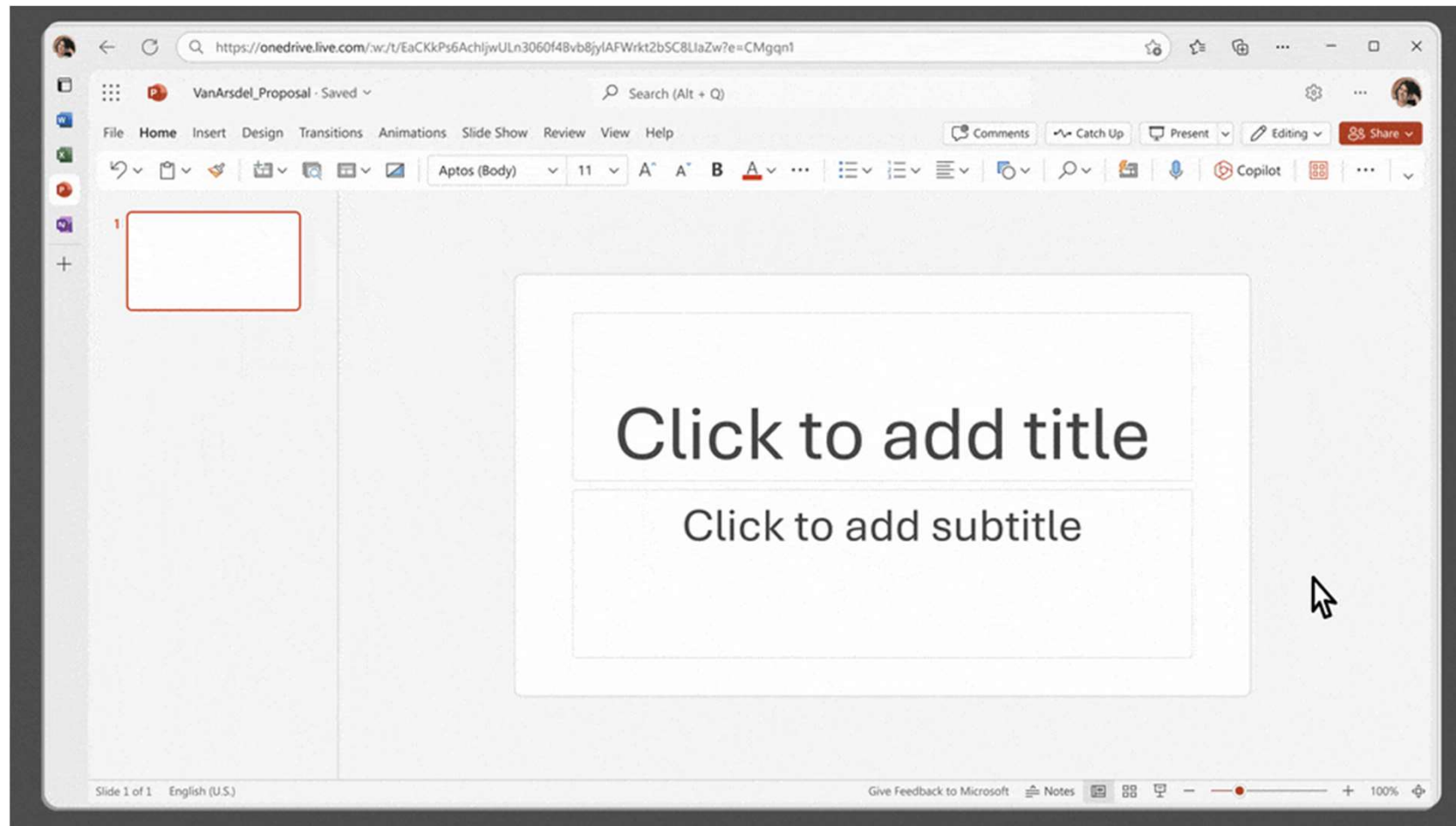
# LLM の支援によるソフトウェア利用

## ソフトウェア利用支援



# Co-pilot 365 [Microsoft, 2023/03/16]

- LLMのパワーをMicrosoft 365に統合することを発表
- 『言葉を地球上で最も強力な生産性向上ツールに変えます。』



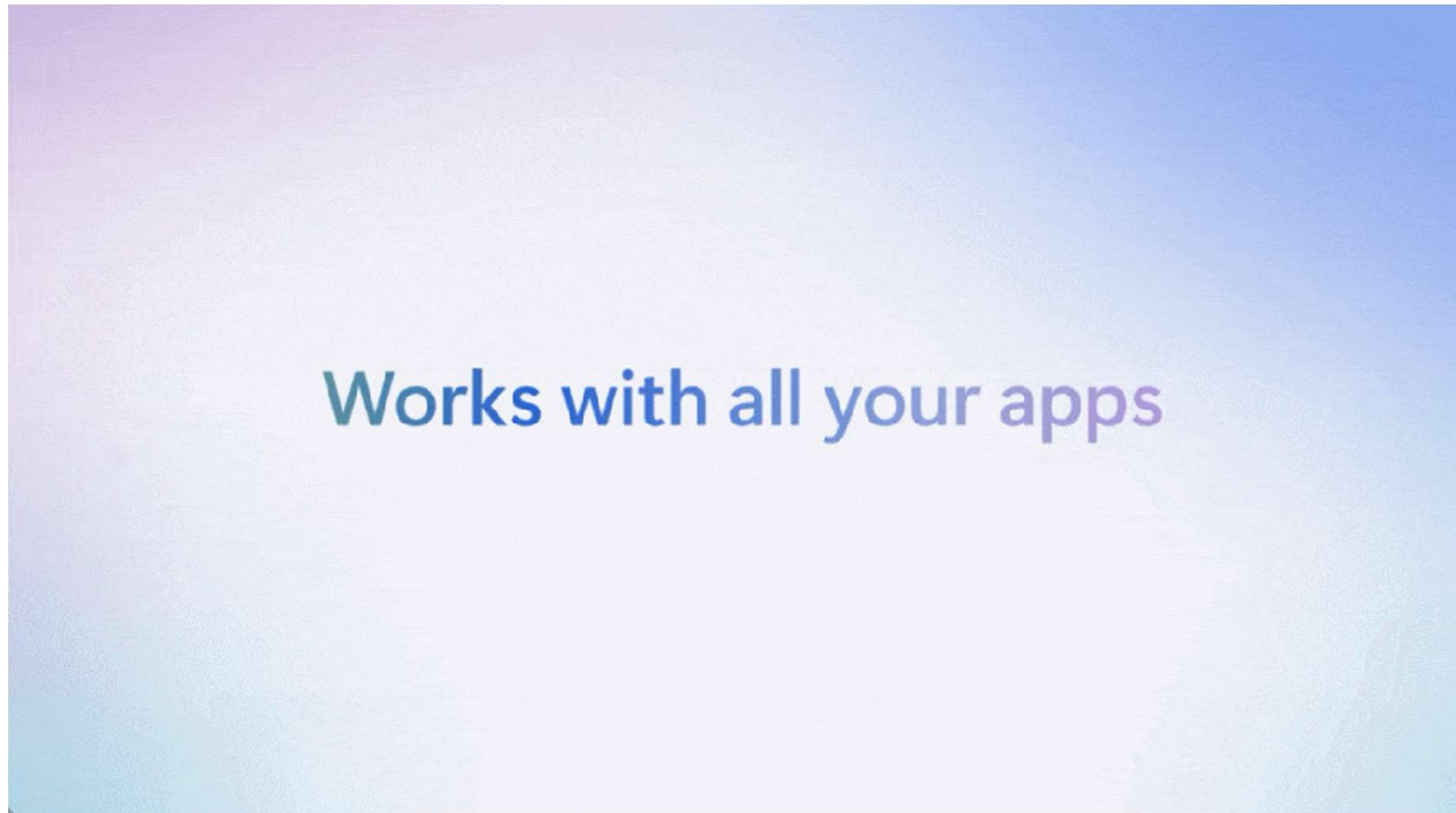
<https://blogs.windows.com/japan/2023/03/28/introducing-microsoft-365-copilot-a-whole-new-way-to-work/>



# Windows Co-pilot [Microsoft, 2023/05/24]



- LLMをさらにWindowsに統合することを発表。Windowsの設定や、文書要約、アプリ実行などを言葉で操作
- Windowsという「世界」をLLMが理解し、行動し始めた



<https://www.youtube.com/watch?v=FCfwc-NNo30>

# ChatGPT Plugins [OpenAI, 2023/3/23]

- ChatGPTにプラグインとして新たな機能を追加する
- 日本では食べログがいち早く登録し， ChatGPTを介した予約機能を提供している



- 最終ゴールを与えられると、GPT-4が自律的に必要なサブタスクに分解。ネットやその他ツールを使ってプランを修正しながらゴールを達成する

## ユーザの入力した“Goals”の例

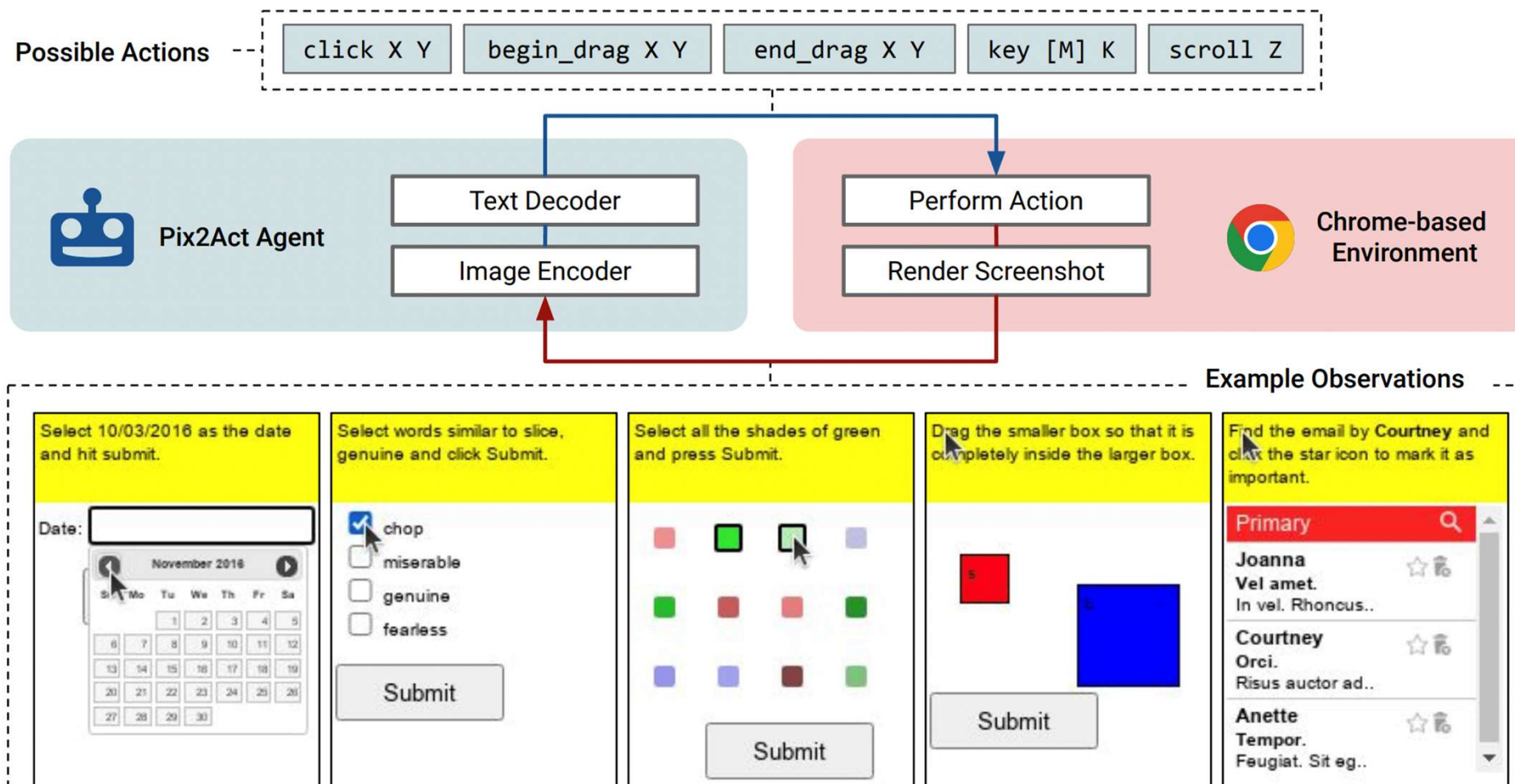
1. Do market research for different headphones on the market today
2. Get the top 5 headphones and list their pros and cons
3. Include the price for each one and save the analysis
4. Once you are done, terminate

## 実行結果

1. Google検索を“top 5 headphones 2023”で実施
2. 検索結果1位のページにアクセスして欲しい情報が無かったと認識
3. Google検索を“top 5 headphones 2023 with pros and cons”で再度実施
4. 検索結果1位のページにアクセスして欲しい情報を発見
5. 正規表現でページから情報抽出しようとするが失敗
6. BeautifulSoupに切り替えて抽出
7. 抽出結果をファイル保存

# Pix2Act [Shaw(DeepMind)+,2023/05]

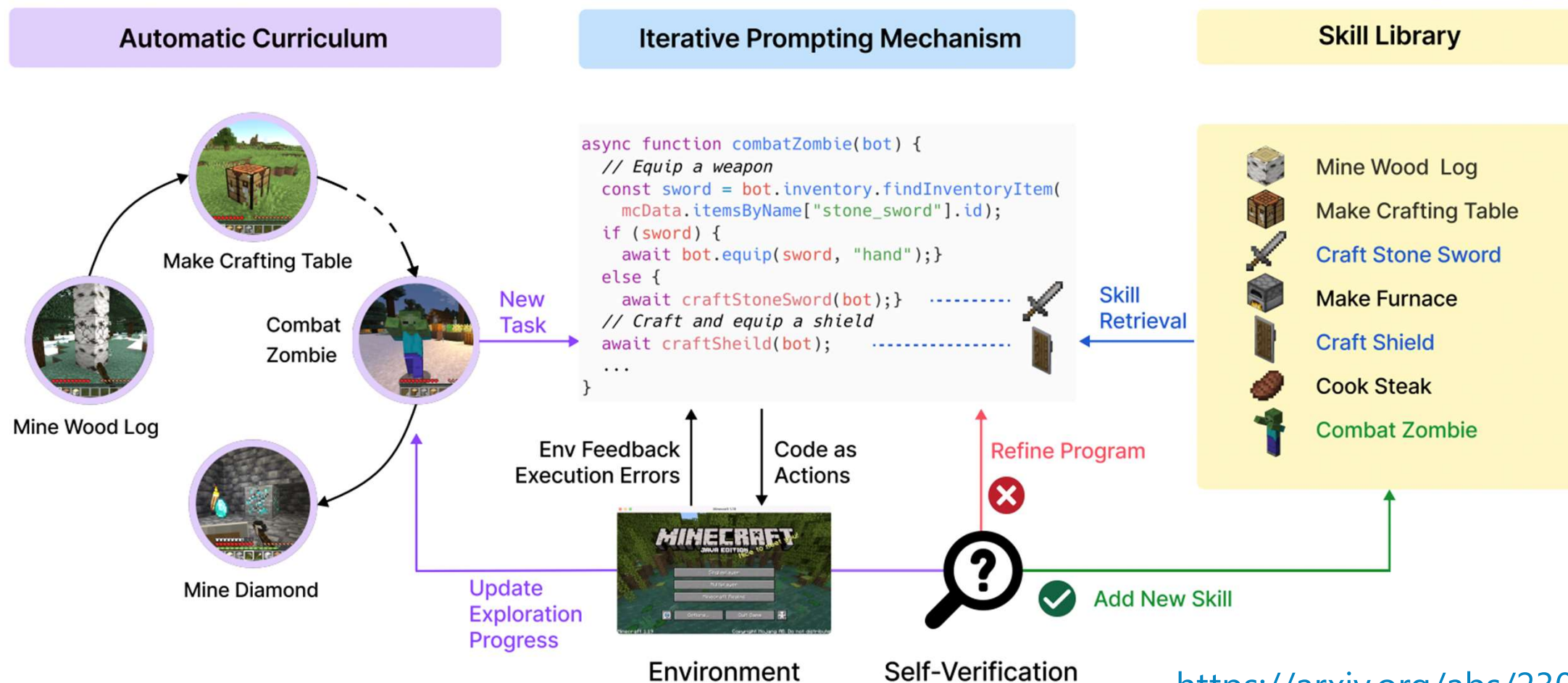
- スクリーンショット画像から，自然言語の指示に従って行動（マウスやキー入力）を生成
- 画像エンコーダとインストラクションチューニングされたLLMを用いている





# Voyager [Wang(NVIDIA, Caltech)+, 2023/05]

- GPT-4/3.5によりMinecraftをプレイするAgent (ただし, 視覚情報は利用していない)
- 世界を探索しながら, 現状に合致したコードをLLMが生成して行動し, スキルとして保存して再利用



<https://arxiv.org/abs/2305.16291>

# ここまでのまとめ



- コードの生成や修正に特化したLLMが作られている
- そうしたLLMは人のソフトウェアエンジニアに匹敵する能力を身に着けつつある
- 大規模言語モデルはソフトウェア工学に大きな変革をもたらすと考えられる
- Github co-pilotをはじめとしたコード生成支援, Windows/365 co-pilotをはじめとしたソフトウェア利用支援にもLLMによる貢献が拡大している



# おわりに



- 大規模言語モデル入門として、基礎知識、BERTとGPT-3、ChatGPTとLLaMAについて紹介しました。
- 大規模言語モデルを活用したソフトウェア開発と利用において、コード作成支援、コード用LLMの構築、ソフトウェア利用支援の現状を紹介しました。