

人文社会ビジネス科学学術院 ビジネス科学研究群 2023年度 春C

# テキストマイニング day 1 (後半)

# スケジュール

## day 1

- 講義 – テキストマイニング概説 (津田先生)
- 講義 – 自然言語処理の最新動向

## day 2

- 講義 – テキストマイニングの手順
- 演習 – テキスト解析 (1)
- 演習 – データ理解

## day 3

- 演習 – テキスト解析 (2)
- 講義&演習 – データ分析 (使い方編)

## day 4

- TextMining Studio の紹介
- 講義&講義 – データ分析 (実践編)

## day 5

- 講義&講義 – データ分析 (実践編)

～LLM時代(ChatGPT登場後)のテキストマイニングのカタチ～

# 自然言語処理の最新動向

## ● ChatGPT の登場

- 従来の自然言語処理は**タスクごとにモデルを学習**していた
- **ChatGPT** ひとつで様々な自然言語処理タスクが解ける → 大きな衝撃
- ChatGPT は、**大規模言語モデル**を人間が好む答えを出すよう追加学習したもの

## ● 大規模言語モデルの成り立ち

- 従来の**言語モデル**は**文章を生成**(次の単語を予測)していた
- **Transformer**で**文章生成する仕組みを超大規模化** → 知識と読解力を得た

## ● ChatGPT 登場後の動向

- ChatGPT に迫る性能を出す **OSS モデル**や学習手法が相次いで登場中
- **ChatGPT を利用してタスクを達成**する Agent の利用が広がる(予想)

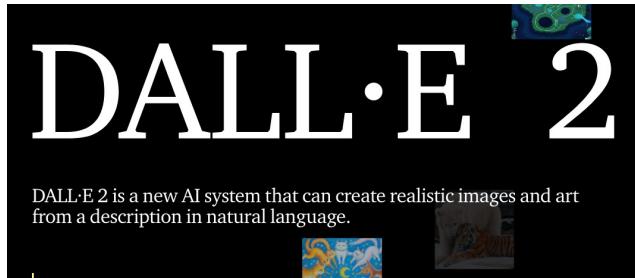
# 目次

- ChatGPT の登場
  - ChatGPT 概観
  - 大規模言語モデルの成り立ち
    - 言語モデル
    - ニューラルネットワークの導入
    - Transformer・BERT・GPT-3
    - プロンプトエンジニアリング
- ChatGPT 登場後の動向
  - OSS モデル
  - 自律駆動型AI
- テキストマイニングの未来予想図

## ChatGPT の登場

# その前に — 2022年の振り返り

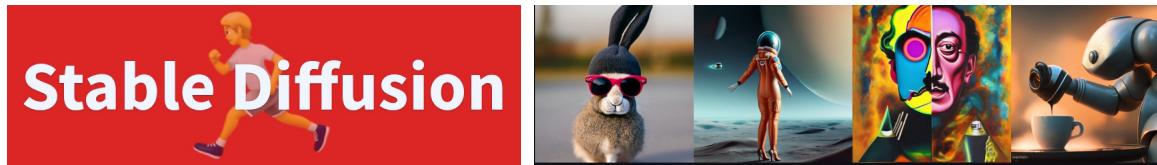
大規模データで学習(自己教師あり学習)した**基盤モデル**とその応用としての**生成系AI**が話題にとりわけ、画像生成AI「Stable Diffusion」や対話型チャットAI「ChatGPT」が登場し、注目を集めた



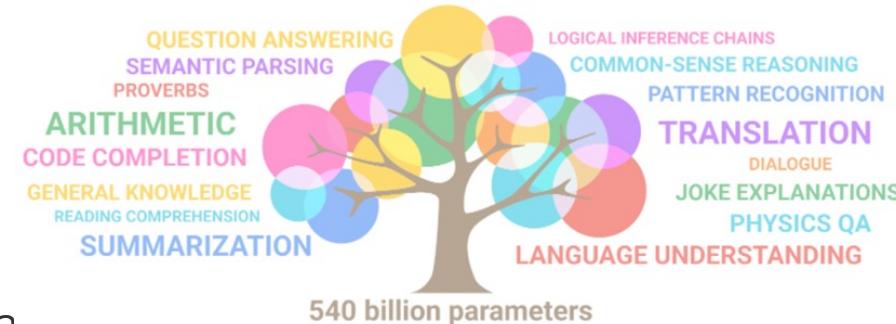
2022/4 に登場した OpenAI による画像生成(Text-to-Image)モデルで、2021年に登場し話題になった DALL·E に比べてより高品質な画像が生成できるようになった



2022/5 に登場した Google による画像生成(Text-to-Image)モデルで、DALL·E に比べてシンプルな構造でありながら、人によるベンチマークでDALL·E を超えたと話題になった

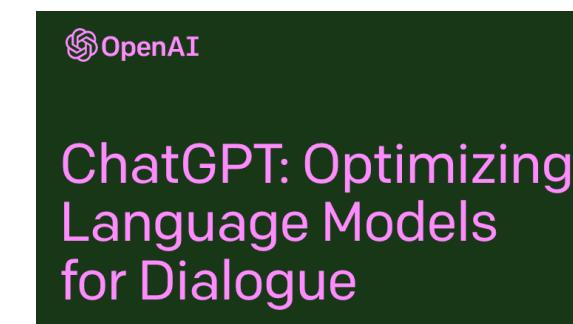


2022/8 にスタートアップの StabilityAi がオープンソース化した画像生成(Text-to-Image)モデルで、コードもモデルも一般公開されているため、モデルのカスタマイズやモデルを組み込んだ新たなサービスが次々と誕生している



540 billion parameters

2022/4 に登場した PaLM は Google によるパラメタ数 **5400億** の**巨大言語モデル**で、OpenAI の GPT-3 パラメタ数 **1750億**を超えて世界最大規模のAIモデルとして話題になった



User this code is not working like i expect — how do i fix it?

```
resultWorkerErr := make(chan error)
defer closer(resultWorkerErr)
go func() {
    defer cancel()
    resultWorkerErr <- b.resultWorker(ctx)
}()

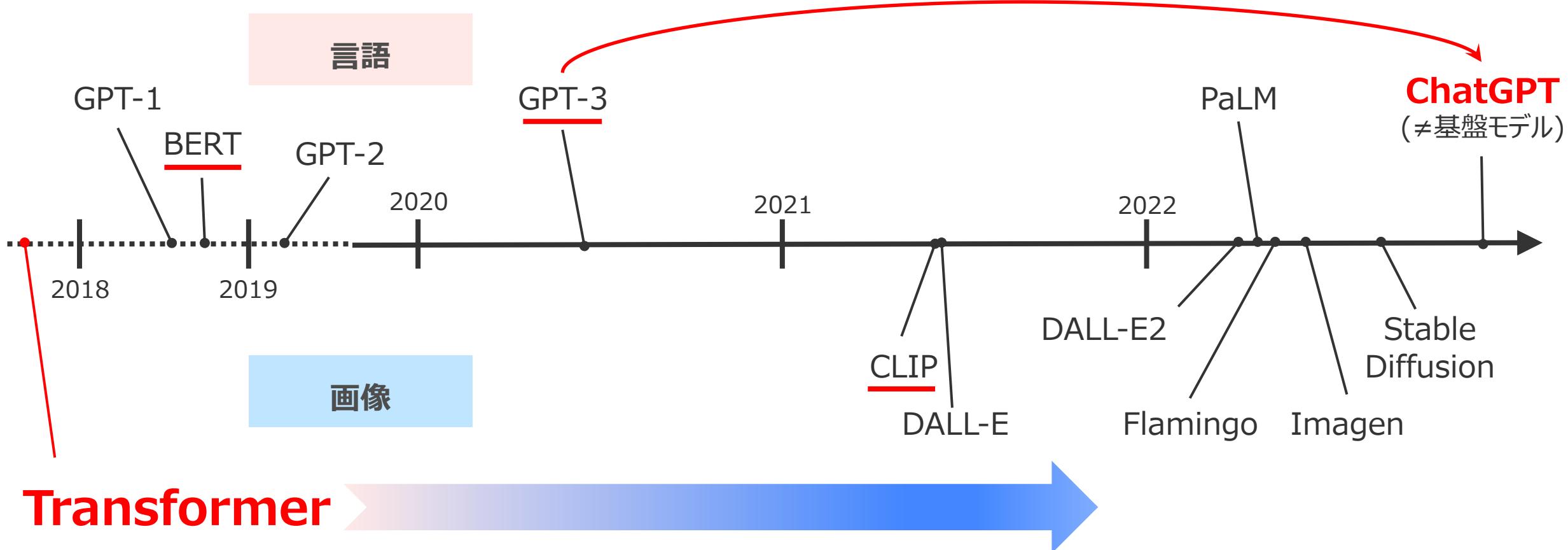
err := b.worker(ctx)
if err == nil {
    return <-resultWorkerErr
}
return multierror.Append(err, <-resultWorkerErr)
```

ChatGPT It's difficult to say what's wrong with the code without more context. Can you provide more information about what the code is supposed to do and what isn't working as expected? Also, is this the entire code or just a part of it?

2022/11 に登場した OpenAI の**対話に特化した言語モデル**で、質問に対する正確な回答や、自然言語で指示することでさまざまな文章やプログラムを生成できることが話題となり、大きく取り上げられた

## その前に — 基盤モデルとは

BERT や GPT-3、CLIP など大量のデータから学習することで**多様なタスクに適応できる高い汎化性能を獲得した AI モデル** ← 2022年、Stanford 大の Bommasani らによって、基盤モデル (Foundation Model) と命名



引用して加筆: [https://blog.recruit.co.jp/data/articles/foundation\\_models/](https://blog.recruit.co.jp/data/articles/foundation_models/)

# ChatGPT とは

- ご存知のように … 自然言語の入力で、何でもできるモデル  
プログラミングができる

からあげ @karaage0703

妻のお願い「複数ページのPDFを1つずつ別ファイルで保存する作業の自動化」  
できそうだけど、疲れ切っていたので考えられず、ダメ元でChatGPTに聞いたらなんとかしてくれました。すごすぎ

このPDFを1つずつ別ファイルで保存する方法

このPDFファイルを1つずつ別ファイルで保存する方法は、以下の手順

アイルを開く  
たいページを選択する  
レ→名前を付けて保存を選択する  
レ名を入力し、保存場所を指定する  
ジ範囲」を選択する  
ジ範囲」欄に、保存したいページの範囲を入力する。例えば、2ページまでを保存したい場合は、「2-5」と入力します。  
」をクリックする  
頁を繰り返して、全てのページを別ファイルで保存する。

実行することで、1つのPDFファイルを複数のファイルに分割する  
、ページ範囲を指定しない場合は、全ページを別ファイルで保存する

このスクリプトを実行すると、指定されたPDFファイルが分割され、各ページが別ファイルとして出力されます。

ねこじすた @necogista · 3月9日  
単純作業が辛いから、アレをあーしてこーしてくれるプログラム書いててと思いつきちゃったので、いつもなら @karaage0703 先生に言うんだけど、絶賛社畜中で毎日遅くて可哀想なので、仕事増やしても良いかしらと一応気を遣っています。でも私も(確定申告しんどすぎて)発狂しそう笑。

出典: <https://twitter.com/karaage0703/status/1633857771304275968>

## 表作成 (データ整理) ができる

日本自動車販売協会連合会と全国軽自動車協会連合会が発表した2022年の国内新車販売台数の総計は420万1321台となり、前年比5.6%減と4年連続のマイナス。登録車は前年比8.3%減の256万3184台。軽自動車も0.9%減の163万8137台と振るわなかつた。これを表形式で整理してください。

もちろんです、以下にご提示のデータを表形式で整理しました。

	2022年販売台数	前年比
総計	420万1321台	-5.6%
登録車	256万3184台	-8.3%
軽自動車	163万8137台	-0.9%

このデータから前年度の国内新車販売台数を計算し、表形式で整理してください。

前年度の販売台数は、現年度の販売台数を前年比の減少率で割った値で計算することができます。以下にその計算結果を表形式でまとめました。

	2022年販売台数	前年比	2021年販売台数 (計算値)
総計	420万1321台	-5.6%	約445万台
登録車	256万3184台	-8.3%	約280万台
軽自動車	163万8137台	-0.9%	約165万台

計算には以下の公式を用いました：  
$$\text{2021年の販売台数} = \text{2022年の販売台数} / (1 - \text{前年比減少率})$$

- 基礎タスク: 言語を応用タスクで利用しやすい形式に変換する

形態素解析

意味解析

固有表現抽出

談話解析

構文解析

など

照応解析

ChatGPT で  
いいんじゃない?

- 応用タスク: 自然言語処理を応用したアプリケーション

テキスト検索

機械翻訳

テキスト分類

質問応答

テキスト要約

対話

情報抽出

など

# ChatGPT — マニュアルからのFAQ生成例 (GPT3.5)



「---」以降の文章からFAQを作成してください。

---

運転中は次のことを必ずお守りください。お守りいただかないと、不意にドアが開き車外に放り出されるなど、重大な傷害におよぶか、最悪の場合死亡につながるおそれがあります。

●すべてのドアを確実に閉め、施錠する。

●走行中はドア内側のドアハンドルを引かない。特に、運転席はドアロックボタンが施錠側になっていてもドアが開くため、注意してください。

●お子さまをリヤ席に乗せるときは、チャイルドプロテクターを使用して車内からドアが開かないようにする。

ドアを開閉するときの留意事項傾斜地・ドアと壁などのあいだが狭い場所・強風など、周囲の状況を確認し、予期せぬ動きにも対処できるよう、ドアハンドルを確実に保持してドアを開閉してください。

ワイヤレスリモコンを使ってドアガラスやムーンルーフを操作するときドアガラスやムーンルーフに人が挟まれるおそれがないことを確認してから操作してください。またお子さまには、ワイヤレスリモコンによる操作をさせないでください。お子さまや他の人がドアガラスやムーンルーフに挟まれたり巻き込まれたりするおそれがあります。

次の操作をすると車高が変わる場合がありますので、十分に周囲の安全を確かめてから操作してください。車高が変わったときに、車両に体が挟まれたりして思わぬ事故につながるおそれがあり危険です。



Q: 運転中に必ず守るべきことは何ですか？



A: すべてのドアを確実に閉め、施錠し、走行中はドア内側のドアハンドルを引かないことです。また、お子さまをリヤ席に乗せるときはチャイルドプロテクターを使用して車内からドアが開かないようにすることも重要です。

FAQ  
生成

Q: ドアを開閉するときに留意すべきことは何ですか？

A: 傾斜地、ドアと壁などのあいだが狭い場所、強風など周囲の状況を確認し、予期せぬ動きにも対処できるよう、ドアハンドルを確実に保持してドアを開閉することが大切です。

Q: ワイヤレスリモコンでドアガラスやムーンルーフを操作するときに留意すべきことは何ですか？

A: ドアガラスやムーンルーフに人が挟まれるおそれがないことを確認してから操作し、お子さまにはワイヤレスリモコンによる操作をさせないようにすることが重要です。

Q: ドアを開閉すると車高が変わる場合があるとありますが、それによって起こりうる事故の例は何ですか？

A: 車高が変わったときに車両に体が挟まれたりして思わぬ事故につながるおそれがあります。したがって、十分に周囲の安全を確かめてからドアを開閉することが重要です。



「---」以降の文章を注意すべき内容と対象者の2つの観点で整理し、表形式でまとめてください。

---

運転中は次のことを必ずお守りください。お守りいただかないと、不意にドアが開き車外に放り出されるなど、重大な傷害におよぶか、最悪の場合死亡につながるおそれがあります



要点  
整理



対象者

注意すべき内容

すべての運転者	・すべてのドアを確実に閉め、施錠する。 ・走行中はドア内側のドアハンドルを引かない。 ・留意事項を守ってドアを開閉する。
---------	--

子どもを乗せる運転者	・チャイルドプロテクターを使用して車内からドアが開かないようにする。
------------	------------------------------------

ドアやムーンルーフを操作する者	・操作前に周囲の安全を確かめる。 ・ワイヤレスリモコンによる操作をさせない。 ・ドアを開閉すると車高が変わるので、周囲の安全を確かめる。
-----------------	--

↑ 入力例には「トヨタ・レクサス取扱説明書」(P.99) を使用した  
[https://manual.lexus.jp/pdf/ls/LS500\\_OM\\_JP\\_M50J08\\_1\\_2201.pdf](https://manual.lexus.jp/pdf/ls/LS500_OM_JP_M50J08_1_2201.pdf)

# ChatGPT — システム要件定義の生成例 (GPT-4)

**プロンプト:** 新規アプリケーションを開発しようとしています。航空券の予約アプリケーションです。予約者はWebブラウザを使用してログインし、予約したい飛行機の便を検索します。予約したい便が見つかったらそれを選択して、予約情報を入力することで予約が完了します。空港の従業員はすべての予約情報を確認することができて、座席ごとに誰が登場する予定なのかを把握することができます。**このアプリケーションの要件定義をしてください。**

以下に、あなたのアプリケーションの基本的な要件定義を記述します。

1. ユーザー要件:

- ユーザーログイン: ユーザーは自分のアカウントでアプリにログインできる必要があります。ユーザーがまだアカウントを持っていない場合は、新たに作成できるようにする必要があります。
- フライト検索: ユーザーは目的地、出発地、出発日、帰着日などのパラメータを使用してフライトを検索できる必要があります。
- フライト選択: ユーザーは検索結果からフライトを選択し、予約することができる必要があります。
- 予約情報の入力: ユーザーは必要な個人情報（名前、連絡先、パスポート情報など）を入力して予約を完了できる必要があります。

2. 空港従業員要件:

- 予約情報の確認: 空港の従業員はすべての予約情報を確認できるようにする必要があります。
- 座席配分情報の確認: 空港の従業員は座席ごとの予約情報を確認し、誰が予定通り出現するかを把握できる必要があります。

3. システム要件:

- データベース: ユーザー情報、予約情報、フライト情報などを保存するためのデータベースが必要です。
- セキュリティ: ユーザーの個人情報や予約情報を安全に保管するためのセキュリティ対策が必要です。
- ユーザビリティ: アプリケーションは使いやすく、直感的なユーザーインターフェースを持つべきです。

**プロンプト:** ユーザーはログインしなくても予約ができるようにしたいです。その場合予約情報を入力するときにユーザー情報を追加で入力する必要があります。これをふまえて**もう一度要件定義を出力してください**。

あなたの要求に基づいて、要件定義を以下のように更新します。

1. ユーザー要件:

- フライト検索: ユーザーは目的地、出発地、出発日、帰着日などのパラメータを使用してフライトを検索できる必要があります。
- フライト選択: ユーザーは検索結果からフライトを選択し、予約することができる必要があります。
- 予約情報の入力: ユーザーは必要な個人情報（名前、連絡先、パスポート情報など）を入力して予約を完了できる必要があります。ログインしていないユーザーも予約を完了できるようにする必要があります。

2. 空港従業員要件:

- 予約情報の確認: 空港の従業員はすべての予約情報を確認できるようにする必要があります。
- 座席配分情報の確認: 空港の従業員は座席ごとの予約情報を確認し、誰が予定通り出現するかを把握できる必要があります。

3. システム要件:

- データベース: ユーザー情報、予約情報、フライト情報などを保存するためのデータベースが必要です。ユーザーがログインせずに予約を行った場合でも、その予約情報を保持する必要があります。

# ChatGPT — 何がすごいか

## ● 驚異的な体験

- 月間アクティブユーザー数1億達成: 2ヶ月 (TikTok 9ヶ月, Instagram 2年半) [[Reuters](#)]

## ● 驚異的な性能

- GRE(米国大学院の入学試験)やUBE(米 国統一司法試験)などで高スコアを収める [[GPT-4](#)]

## ● 驚異的なスピード

**2022年11月** • OpenAI が ChatGPT 発表 (11月30日)

**2022年12月** • ChatGPT が公開後5日で100万ユーザを獲得  
• Google が ChatGPT に関して「コードレッド」を宣言と報道

**2023年1月** • Microsoft が OpenAI に約1.3兆円を投資するとの報道

**2023年2月** • Microsoft が ChatGPT を搭載した検索エンジン Bing を発表  
• Google が対話型のサービス Bard を限定公開

**2023年3月** • OpenAI が ChatGPT と Whisper の API を公開  
• OpenAI が GPT-4 を発表

かなり

抜粹

# ChatGPT — 大規模基盤モデル GPT-3.5 をファインチューニングしたモデル

事前学習した**大規模言語モデル GPT-3.5**をベースに、人間の質問に答えるように「①ファインチューニング」と人間の好みに合った答えを出すように「②人間のフィードバックに基づく強化学習 (RLHF)」を加えた**AIモデル**

## Step1

人間の用意した望ましい回答で事前学習モデル(GPT-3)を fine-tuning

A prompt is sampled from our prompt dataset.

Explain reinforcement learning to a 6 year old.



We give treats and punishments to teach...

A labeler demonstrates the desired output behavior.



This data is used to fine-tune GPT-3.5 with supervised learning.

## Step2

Step1の出力に人間がランク付けし、報酬モデルを学習

A prompt and several model outputs are sampled.

Explain reinforcement learning to a 6 year old.

A B

## RLHF: 人間が好む出力をするように訓練する手法

- ・モデルの出力に対して人間がスコアを付与
- ・スコアを最大化するようにモデルを訓練

to worst.

D > C > A > B

RM

This data is used to train our reward model.

The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.



r\_k

## Step3

Step2で学習した報酬モデルを使ってPPOで強化学習

## 主な特徴

- ・前の対話内容に続く質問への回答が可能
- ・間違った回答をすることも多い(例: 肉じゃがのレシピを訊くと、ホワイトソースを入れようとした)
- ・間違いを認めることもできる
- ・正しくない前提に対する異議を唱えることもできる
- ・不適切なリクエストには応じない
- ・スクリプトやコード生成も可能

## 他のツールとの連携

- ・ChatGPT for Google: Google の検索結果に ChatGPTの出力を表示してくれる拡張機能
- ・ArxivGPT: Arxiv の検索結果に論文の要約やポイントを表示してくれるかく超機能

引用: <https://openai.com/blog/chatgpt/>

# ArxivGPT — Chrome拡張機能

プロンプト(画面右)に沿って、ChatGPT が対象論文のサマリーを作ってくれる (プロンプトはカスタマイズ可能)

The screenshot shows a research paper from arXiv.org. The main content is a summary of the paper "Attention Is All You Need" by Vaswani et al. A red dashed box highlights the summary text on the right side of the page. A red arrow points from this summary to a larger text box on the right, which contains the detailed prompt used by ArxivGPT to generate the summary.

**arXivGPT** "https://arxiv.org/" prompt is used

要約：従来の系列変換モデルはエンコーダー・デコーダー構成の複雑な再帰型や畳み込みニューラルネットワークに基づいている。性能の良いモデルは、エンコーダーとデコーダーをアテンションメカニズムで接続している。我々は、再帰性や畳み込みを完全に排除し、アテンションメカニズムだけに基づく新しいシンプルなネットワークアーキテクチャ、Transformerを提案する。2つの機械翻訳タスクでの実験結果は、これらのモデルがより優れた品質であり、並列化が可能で、訓練時間がかなり短くて済むことを示している。

洞察と学び：アテンションメカニズムだけに基づくTransformerモデルは、再帰型や畳み込みニューラルネットワークに基づくモデルよりも優れた品質であり、並列化が可能で、訓練時間がかなり短くて済むことを示している。このモデルは、他のタスクにも適用でき、汎用性が高いことも示されている。

関連研究の課題と著者の仕事との違い：

- 従来のモデルは再帰型や畳み込みニューラルネットワークに基づいており、Transformerはアテンションメカニズムだけに基づく。

下：設定画面におけるプロンプト設定例※

The screenshot shows the ArxivGPT settings interface. It displays a custom prompt for generating a summary of the arXiv homepage. The prompt text is:

著者による論文の内容を簡潔に一文で要約してください。次に、その論文から得られた重要な洞察と学びを挙げてください。次に、論文の内容に基づいて、関連研究の課題と著者の仕事との違いについて3~5つ挙げてください。次に、論文の内容に基づいて、残る課題と将来の研究の方向性について3~5つ挙げてください。最後に、論文の内容に基づいて、論文の研究分野から少なくとも5つの関連文献をリストアップしてください。日本語で。

Save Prompt

+ Add Prompt

**Trigger Mode**

**Always**  
ArxivGPT is queried on every search

**Manually**  
ArxivGPT is queried when you manually click a button

※ abstract 中に記載のない内容はハリシネーションに注意が必要 (例: 関連文献リストなど)

# GPT-4 – ChatGPT をさらに上回る人レベルの言語理解・生成能力

GPT-4	(参考) ChatGPT
<ul style="list-style-type: none"><li>• GPT-3.5 (2世代前) や ChatGPT (1世代前)の後継モデル</li><li>• 画像とテキストを入力し、テキストを出力するよう訓練されている</li><li>• モデルのパラメータ数や構造、学習データ・方法などの詳細は未公開</li></ul>	<ul style="list-style-type: none"><li>• GPT-3.5 (1世代前) をファインチューニングしたモデル ※ GPT-3.5 はテキストとコードで訓練されている</li><li>• 基本的に InstructGPT の方式を踏襲</li></ul>

Technical Report (<https://arxiv.org/pdf/2303.08774.pdf>) にある例

Example of GPT-4 visual input:  
User What is funny about this image? Describe it panel by panel.



Source: <https://www.reddit.com/r/hmmm/comments/ubab5v/hmmm/>

GPT-4 The image shows a package for a "Lightning Cable" adapter with three panels.

Panel 1: A smartphone with a VGA connector (a large, blue, 15-pin connector typically used for computer monitors) plugged into its charging port.

Panel 2: The package for the "Lightning Cable" adapter with a picture of a VGA connector on it.

Panel 3: A close-up of the VGA connector with a small Lightning connector (used for charging iPhones and other Apple devices) at the end.

The humor in this image comes from the absurdity of plugging a large, outdated VGA connector into a small, modern smartphone charging port.

Table 3. Example prompt demonstrating GPT-4's visual input capability. The prompt consists of a question about an image with multiple panels which GPT-4 is able to answer.

ユーザー: この画像のどこが面白いのか? パネルごとに描写してください。

GPT-4: この画像は「Lightning Cable」アダプターの3つのパネルが描かれたパッケージ示しています。パネル1: ...

この画像のユーモアは、**大きく時代遅れのVGAコネクタを小さな現代的なスマートフォンの充電ポートに接続する**という不合理さから来ています。

GPT-4 visual input example, École Polytechnique Exam Question:  
User Answer question I.1.a. Think step-by-step.

I. Principale de la détection de rayonnement avec un bolomètre  
Comme illustré sur la figure 1 un bolomètre est constitué d'un absorbeur qui reçoit le rayonnement que l'on désire détecter. Sa température  $T$ , supposée uniforme, est mesurée à l'aide d'un thermomètre incorporé, constitué d'un matériau conducteur dont la résistance  $R(T)$  varie avec la température  $T$ . Cette variation est caractérisée par le coefficient  $\alpha = \frac{\partial R}{\partial T}$ . L'ensemble possède la capacité thermique  $C_{th}$ .

Un barreau conducteur thermique, longueur  $L_c$ , de section  $S$  et de conductivité thermique  $\lambda$  et sans échange thermique latéraux, relie le bolomètre à un thermomètre de température  $T_b$ . Bas.

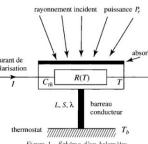


Figure 1 - Schéma d'un bolomètre  
I.1. Dans cette première partie, on étudie le comportement du bolomètre en régime stationnaire, à la température de fonctionnement  $T = T_0$   
I.1.a) Exprimer la température  $T(x)$  en tout point du barreau conducteur repéré par son abscisse  $x$ , si on prends  $P(0) = T_0$  et  $T(L) = T_b$ .

GPT-4 To answer question I.1.a, we need to express the temperature  $T(x)$  in every point of the conductive bar, which is characterized by the abscissa  $x$ . We are given that  $T(0) = T_0$  and  $T'(L) = T_b$ .

For a stationary regime, the heat equation for the one-dimensional heat conduction in the bar is:  
$$\frac{d^2T}{dx^2} = 0$$

Integrating once, we get:  
$$\frac{dT}{dx} = C_1$$

Integrating a second time:  
$$T(x) = C_1 x + C_2$$

To determine the constants  $C_1$  and  $C_2$ , we use the boundary conditions:

$T(0) = T_0$ , which gives  $C_2 = T_0$   
 $T(L) = T_b$ , which gives  $C_1 = (T_b - T_0)/L$

Finally, the expression for  $T(x)$  is:  
$$T(x) = (T_b - T_0) * (x/L) + T_0$$
  
This represents a linear temperature profile along the conductive bar, with a slope of  $(T_b - T_0)/L$ .

Table 15. Example prompt demonstrating GPT-4's visual input capability. The prompt consists of a question which requires understanding a physics problem with a diagram written in French and solving it using chain-of-thought prompting.

物理の問題を解いている

# 代表的なサービス

	Azure OpenAI Service	OpenAI
利用可能モデル	OpenAI GPT-4, GPT-3.5, GPT-3, Codex, Embeddings, DALL-E	OpenAI GPT-4, GPT-3.5, GPT-3, Codex, Embeddings, DALL-E, Whisper(音声認識)
サービス提供速度	OpenAI がサービス提供した後、遅れてサービス提供開始	最新モデルが先に提供される場合が多い
セキュリティ	<ul style="list-style-type: none"> <li>• <a href="#">Azure のセキュリティ基準に準拠</a></li> <li>• <a href="#">APIキーによる認証とAzure AD認証に対応</a></li> <li>• Azure 仮想ネットワークによる保護</li> </ul>	<ul style="list-style-type: none"> <li>• OpenAI のセキュリティポリシーに準拠 (<a href="#">脆弱性開示ポリシー</a>)</li> <li>• APIキーによる認証</li> </ul>
データ	<ul style="list-style-type: none"> <li>• お客様から提供されたトレーニングデータは、お客様のモデルの fine-tuning (微調整) にのみ使用される</li> <li>• マイクロソフトのモデルをトレーニング/改善するために使用することはない (<a href="#">参考</a>)。</li> <li>• データは悪用/誤用の監視目的で30日間保持され、承認されたマイクロソフト社員がレビューする可能性がある (<a href="#">参考</a>)。保持されないよう要求可能。</li> </ul>	<ul style="list-style-type: none"> <li>• API経由のデータは OpenAI のモデルをトレーニング/改善するために使用することはない (<a href="#">参考</a>)</li> <li>• データは悪用/誤用の監視目的で30日間保持され、OpenAI社員/サードパーティ業者がレビューする可能性がある。保持されないよう要求可能。</li> </ul>
プライバシー	<ul style="list-style-type: none"> <li>• <a href="#">マイクロソフトの声明</a>および<a href="#">Azure OpenAI Serviceの製品ポリシー</a>に準拠</li> <li>• 日本の法律を準拠法とし、東京地裁裁判所を管轄裁判所として契約可能</li> </ul>	<ul style="list-style-type: none"> <li>• OpenAI の<a href="#">プライバシーポリシー</a>に準拠。カリフォルニア州法に準拠</li> </ul>
価格	モデル利用価格は同一 ( <a href="#">Azure の価格体系に基づく</a> )	モデル利用価格は同一 ( <a href="#">OpenAI の価格体系に基づく</a> )
SLA	<ul style="list-style-type: none"> <li>• 99.9%以上の稼働率を保証 (<a href="#">詳細</a>)</li> </ul>	SLAは提供されていない

出典: <https://zenn.dev/microsoft/articles/e0419765f7079a>

# ChatGPT — 知つておくべき代表的なリスク

項目	概要	考えられる事例
Hallucinations	幻覚、でっち上げ。特定のソースとの関係で無意味な内容や真実でない内容を作り出す傾向がある	でっち上げの情報を元に、顧客への提案資料を作成
Harmful content	ポリシーに反するコンテンツや、個人・集団・社会に害を及ぼす可能性のあるコンテンツを生成する可能性がある	LLMを組み込んだサービスを提供している場合、顧客に対して、有害なコンテンツを出力
Disinformation and influence operations	ニュース記事、ツイート、対話、メールなど、もっともらしく現実的で的を射たコンテンツを生成することが可能である	社員が業務中にフェイクニュースを作成し、SNS 投稿
Privacy	一般に利用可能なデータソースから学習しており、これは、個人情報を含む場合がある。その結果、個人の特定を試みるために使用される可能性がある	LLMを組み込んだサービスを提供している場合、顧客が訓練データ等の情報をLLMから抽出
Cybersecurity	ソーシャルエンジニアリングのいくつかのサブタスク(フィッシングメールの作成など)や、ソースコード内等の脆弱性を説明が可能	プログラム作成の際に、セキュアなコードへの変換を依頼し、プログラム内に埋め込まれた機微な情報(クレデンシャル等)が流出

引用: [中島,2023] AI-SCHOLAR主催「What is GPT」講演資料 (中島氏が [GPT-4 Technical Report](#) から抜粋して加筆修正)

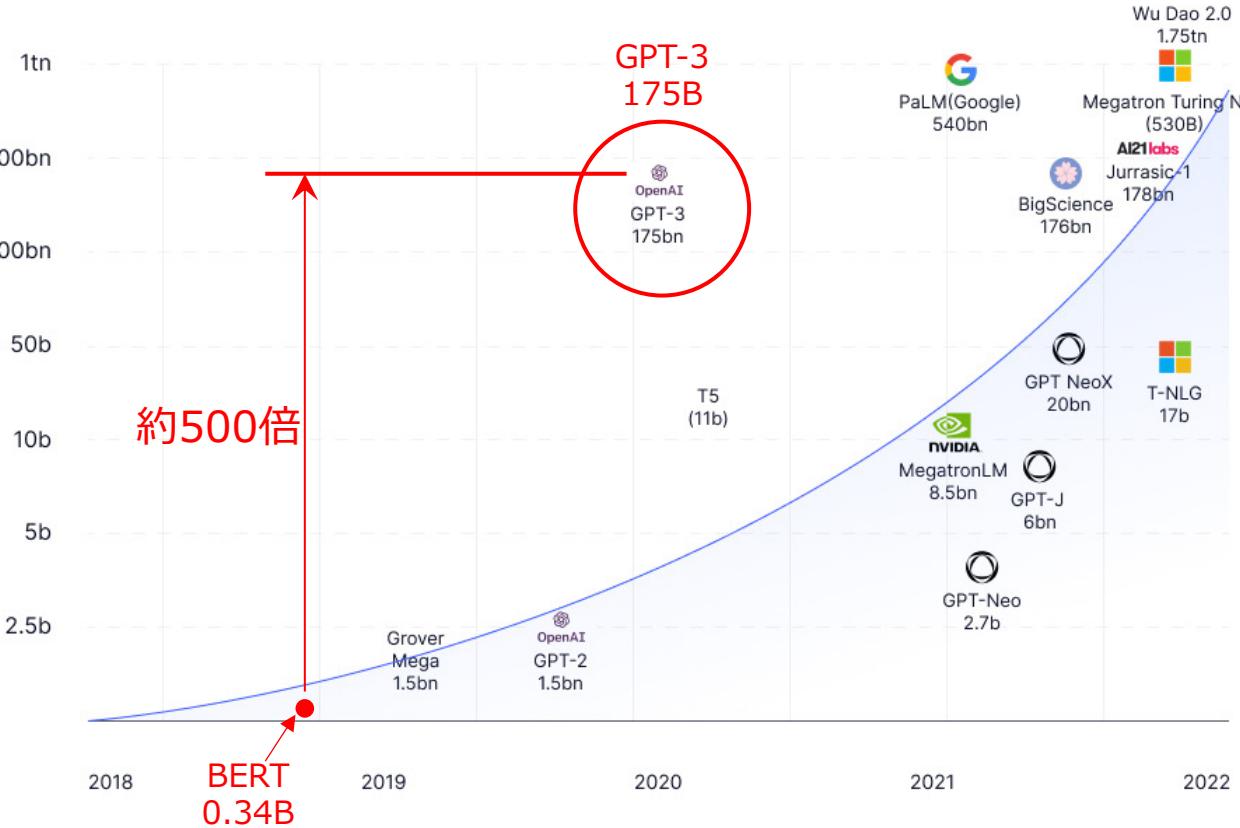
## 大規模言語モデル (LLM) の成り立ち

# 大規模言語モデル (Large Language Model: LLM) とは

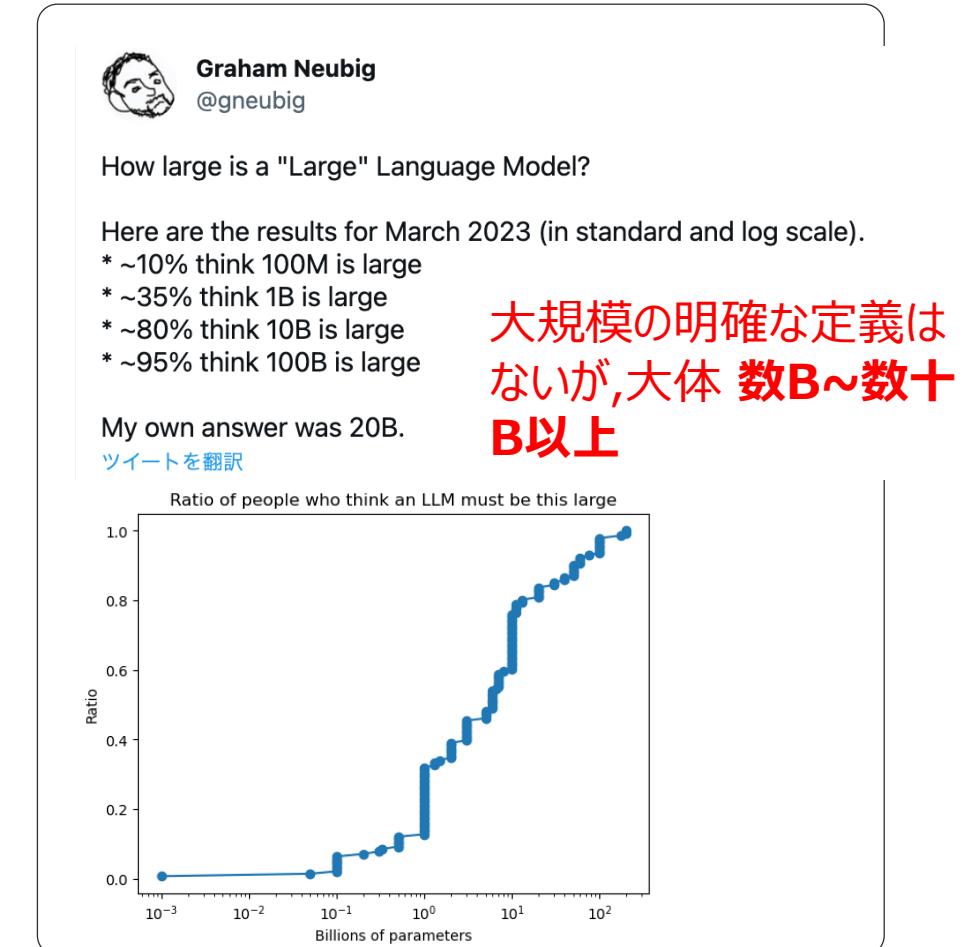
- 大量の計算機資源で、大量データを大きなモデルで学習すれば良いモデルができる

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

出典: <https://twitter.com/gneubig/status/1631386071228358658>



出典: <https://textcortex.com/ja/post/how-gpt-3-writing-tools-work>



# そもそも、言語モデル (Language Model: LM) とは

- 単語の並びの生成確率をモデル化したもの

単語の並び  $\Rightarrow y_1, y_2, \dots, y_T$  生成確率  $\Rightarrow P(y_1, y_2, \dots, y_T)$

- 特定の単語の次に来る単語を予測できる

$$y^* = \operatorname{argmax}_{y \in V} P(\text{日本}, \text{の}, \text{首都}, \text{は}, y)$$

$P(\text{日本}, \text{の}, \text{首都}, \text{は}, \text{ロンドン})$	= 0.00000043
$P(\text{日本}, \text{の}, \text{首都}, \text{は}, \text{パリ})$	= 0.00000082
$P(\text{日本}, \text{の}, \text{首都}, \text{は}, \text{東京})$	= <b>0.00000103</b>
$P(\text{日本}, \text{の}, \text{首都}, \text{は}, \dots)$	= ...

計算された確率が最大値を  
取る単語を選択する  
↓  
**東京**

- 単語の並びに従ってモデル化することで文脈も考慮  $\rightarrow$  N-gram言語モデル

$$y^* = \operatorname{argmax}_{y \in V} P(\text{日本}, \text{の}, \text{首都}, \text{は}, y) = \operatorname{argmax}_{y \in V} P(y | \text{日本}, \text{の}, \text{首都}, \text{は})$$

yの前方N-1個の単語(トークン)

- 分布仮説 [[Harris+, 1954](#)]

- 単語の意味はその周囲の単語から形成されるという仮説  
→ 似た文脈で出現する単語は意味が似ている

文1: 昨日、りんごを食べた。りんごジュースを飲んだ。りんごの皮をむいた。

文2: 昨日、りんごを食べた。ぶどうジュースを買った。ぶどうの皮をむいた。

文3: 昨日、自転車に乗った。自転車を修理した。自転車を買った。

共起による  
ベクトル表現  
[[Lin, 2002](#)]

	…	食べる	…	飲む	…	修理	…
りんご	0	1	0	1	0	0	0
ぶどう	0	1	0	1	0	0	0
自転車	0	0	0	0	0	1	0
:							

← 語彙数(数万～数十万)分の疎なベクトルになる →

似てる  
似てない

## ● 分布仮説 [Harris+, 1954]

- 単語の意味はその周囲の単語から形成されるという仮説

→ 似た文脈で出現する単語は意味が似ている

- 各意味を複数の次元で分散して表現する (=分散表現)

→ 次元は低次元(例えば100次元)で、値は実数値

单語埋め込み  
(word embedding)  
とも呼ばれる

これらの実数値をニューラルネットワークで求める

共起による  
ベクトル表現  
[Lin, 2002]

次元→	0	1	…	50	…	98	99
りんご	1.07	-1.08		1.48		0.46	0.48
ぶどう	1.95	-1.53		0.36		-0.61	-0.44
自転車	0.67	1.44		-1.50		0.10	0.67
:							

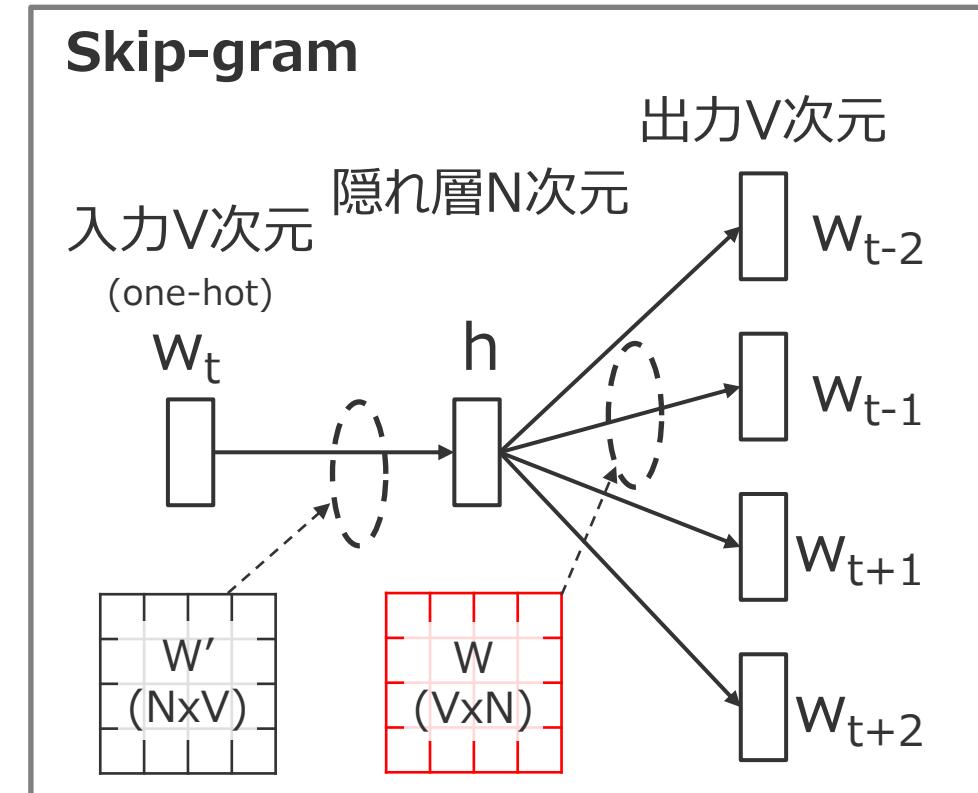
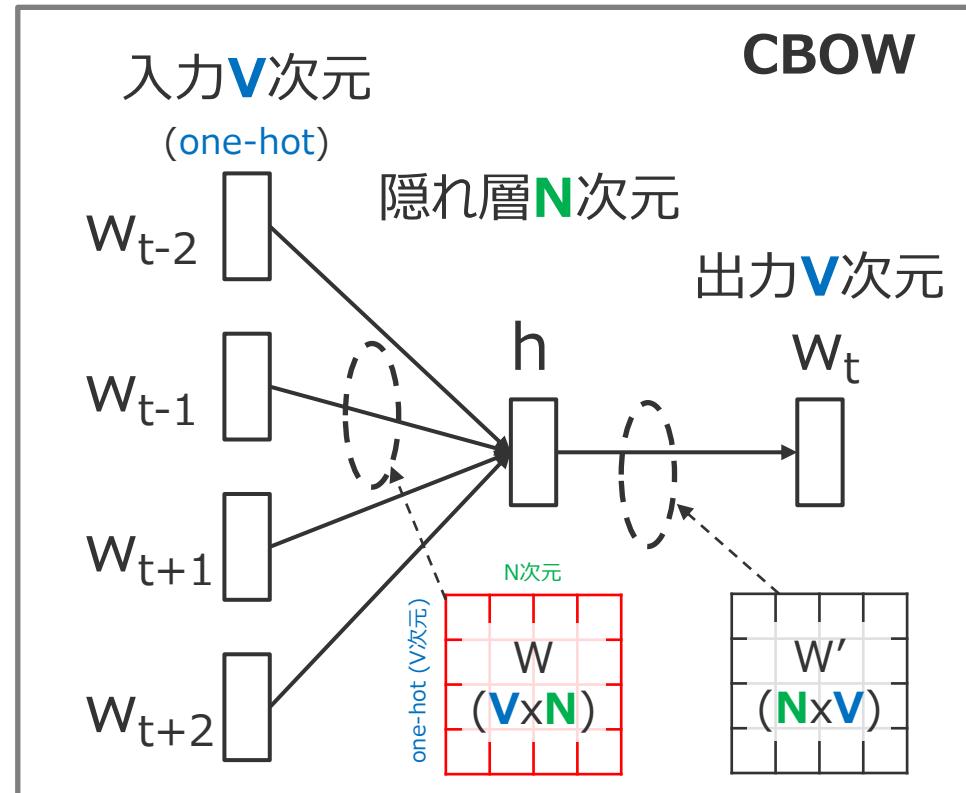
← 高々数百次元の密なベクトル →

似てる  
似てない

# ニューラルネットワークの導入

- 代表格: word2vec [[Mikolov+, 2013](#)]

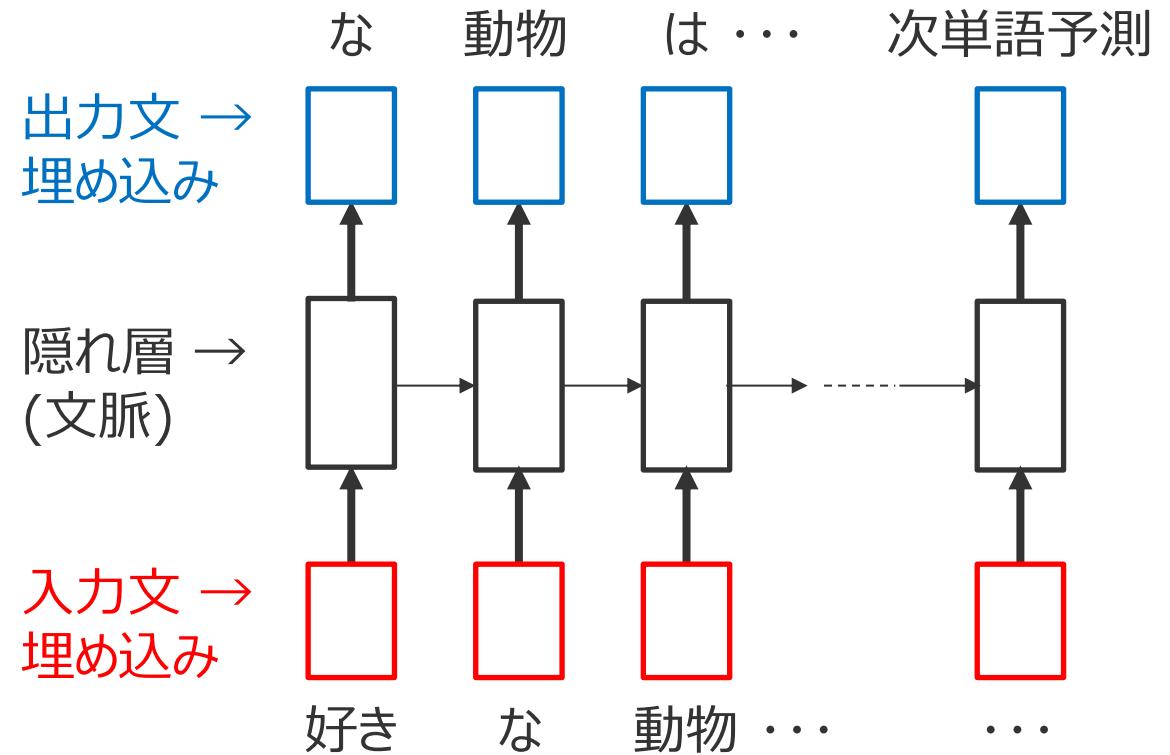
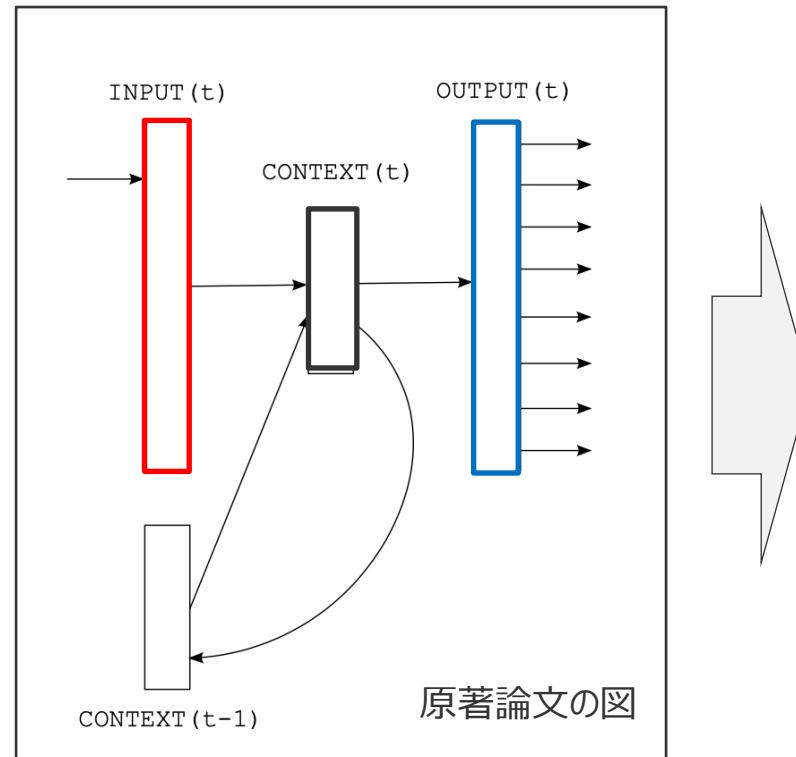
- 従来の単語ベクトルも類似度の比較はできていたが、足したり引いたりできなかった
  - $\text{king} - \text{man} + \text{woman} = \text{queen}$  が有名



# ニューラル言語モデル

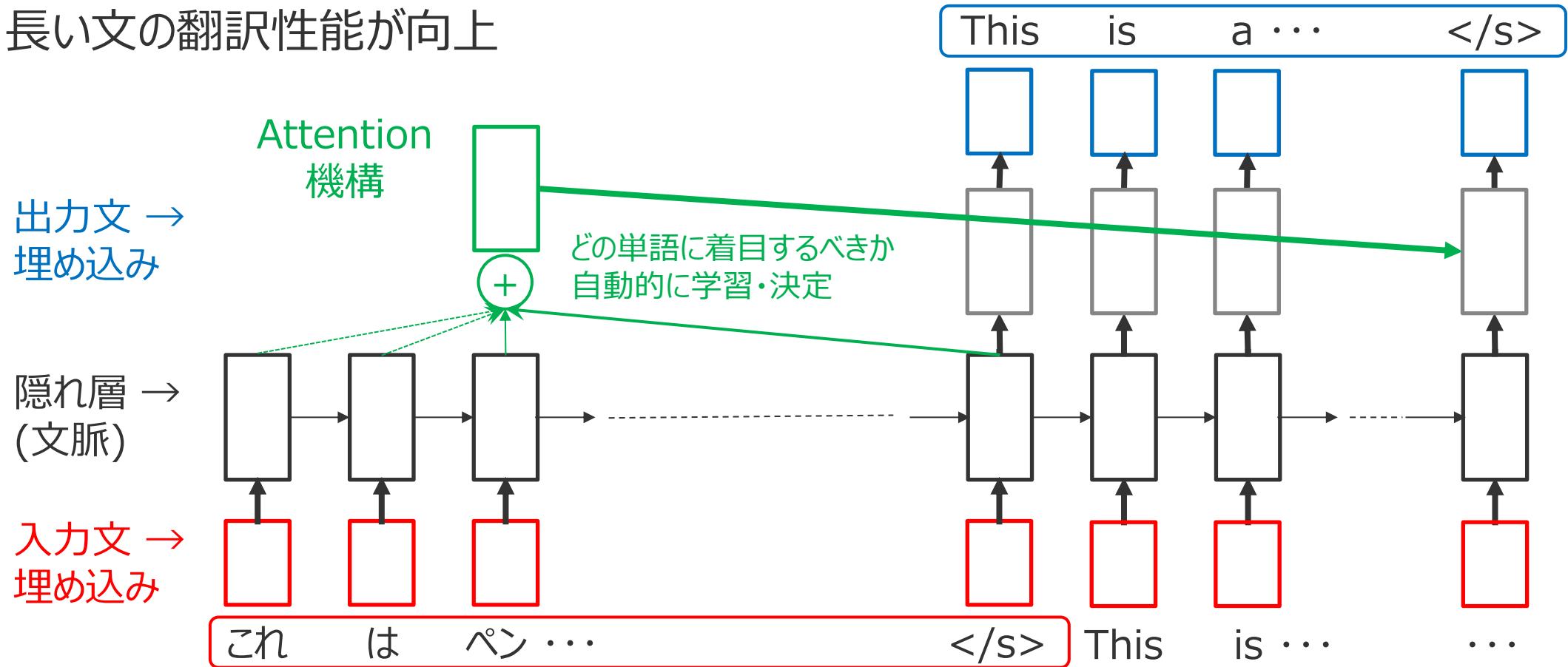
## ● RNN 言語モデル [Mikolov+, 2010]

- RNN(系列データを対象とするNN)を使った言語モデルで、次の単語を予測する
- 隠れ層に過去の履歴(文脈を考慮した情報)が埋め込まれていく



# エンコーダー-デコーダー型の機械翻訳モデル

- Seq2Seq [[Sutskever+, NIPS2014](#)]: ニューラル機械翻訳の基本となったモデル
- Attention 機構 [[Bahdanau+, 2015](#)] により、  
長い文の翻訳性能が向上

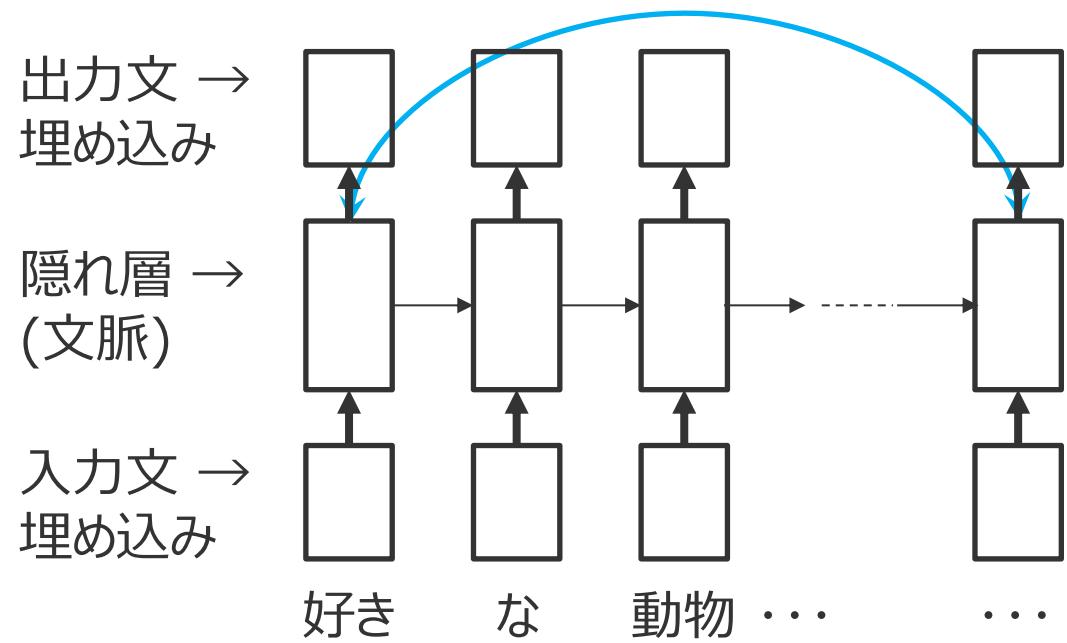


# Self-Attention の登場

- 従来の文脈理解は、長期依存性の理解に限界があった
- 離れた単語の関係性も直接考慮できる Self-Attention が性能向上に大きく寄与

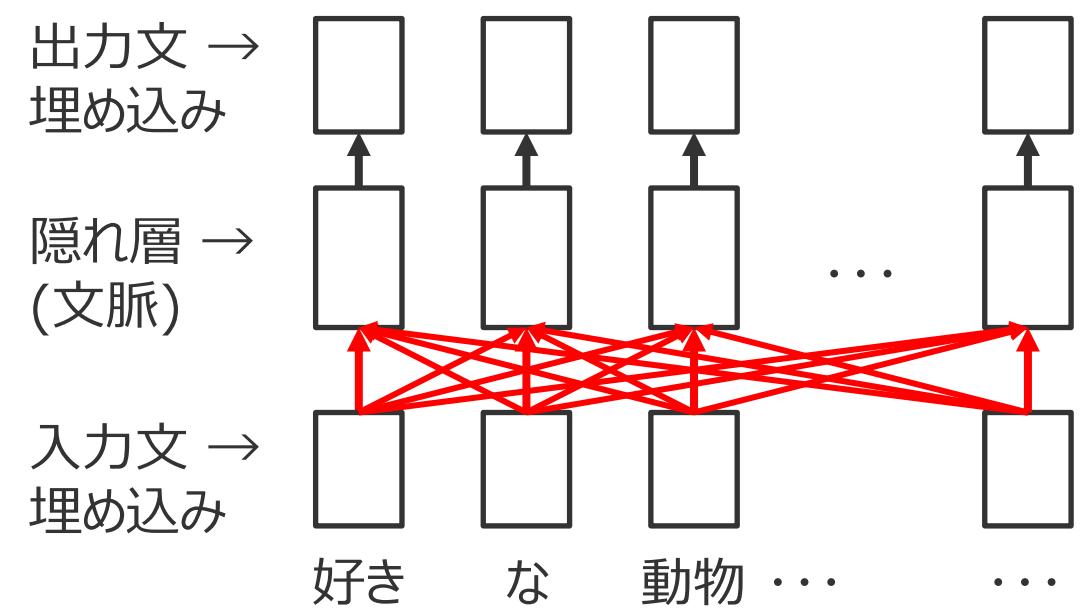
## 従来(LSTM)

遠く離れた単語の関係性  
を捕まえにくい



## Self-Attention

遠く離れた単語も直接  
関係性を考慮できる



# Transformer [Vaswani+, NIPS2017]

- 単語間の関係を RNN や CNN を用いずアテンションのみを用いて表現したエンコーダデコーダ型モデルにより、機械翻訳で圧倒的な SOTA を達成

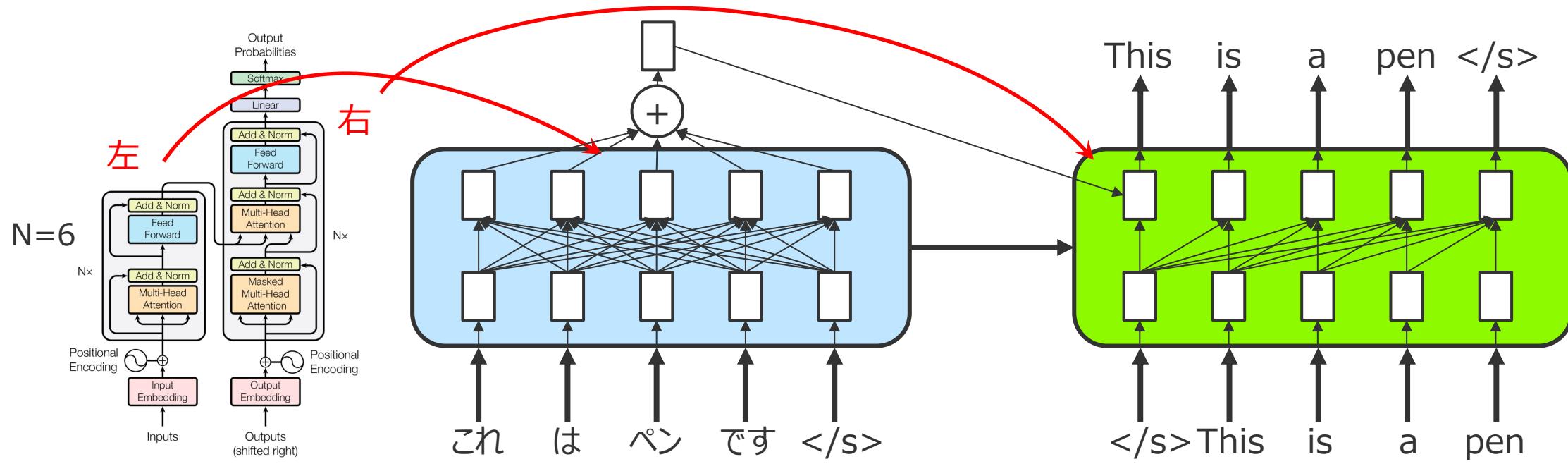


Figure 1: The Transformer - model architecture.

よく見る図

Transformer エンコーダ

引用: [西田,2022] JSAI2022 チュートリアル講演資料の一部を修正して作成

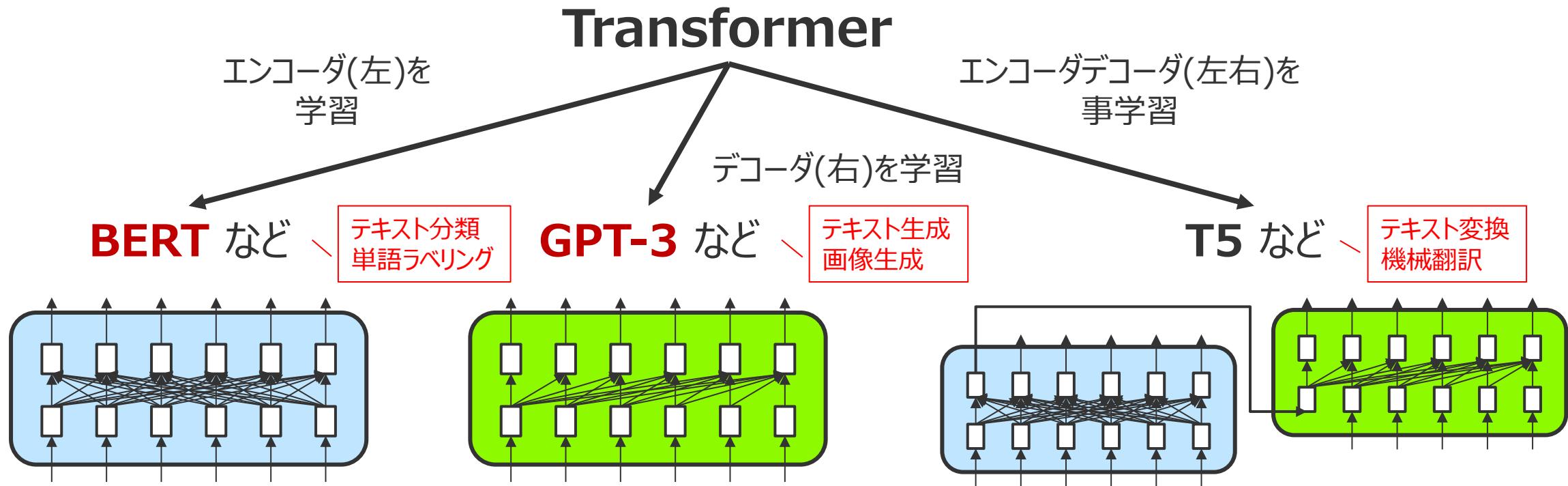
双向型のNNモデル  
(右側の単語も使う)

Transformer デコーダ

自己回帰型のNNモデル  
(出力を入力に戻す)

# Transformer [Vaswani+, NIPS2017]

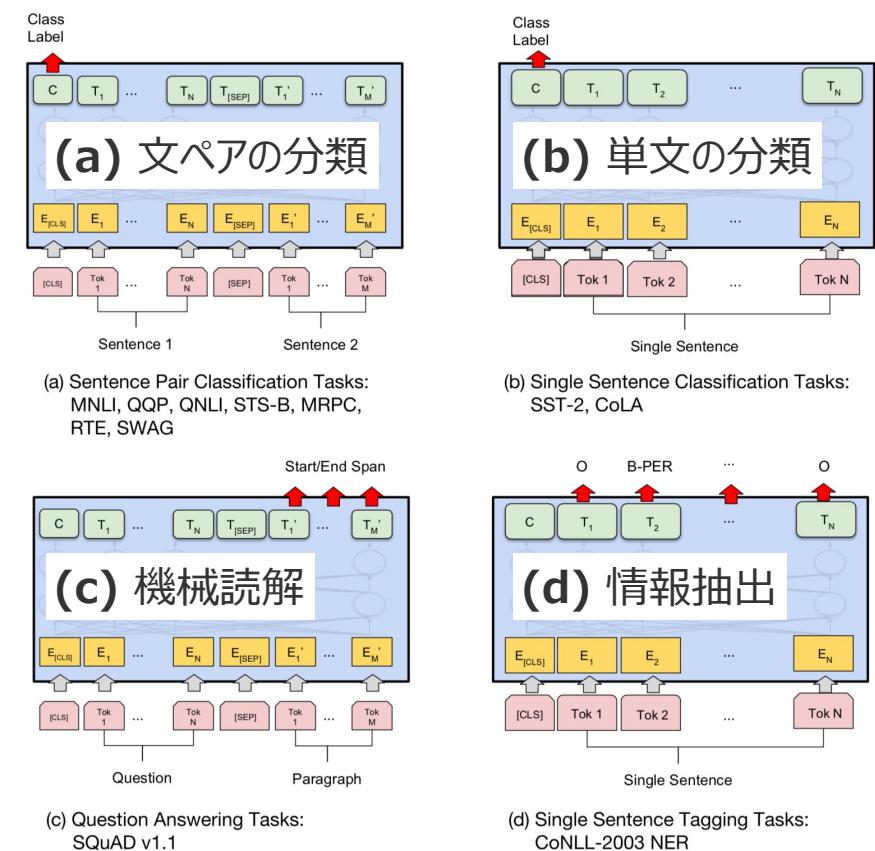
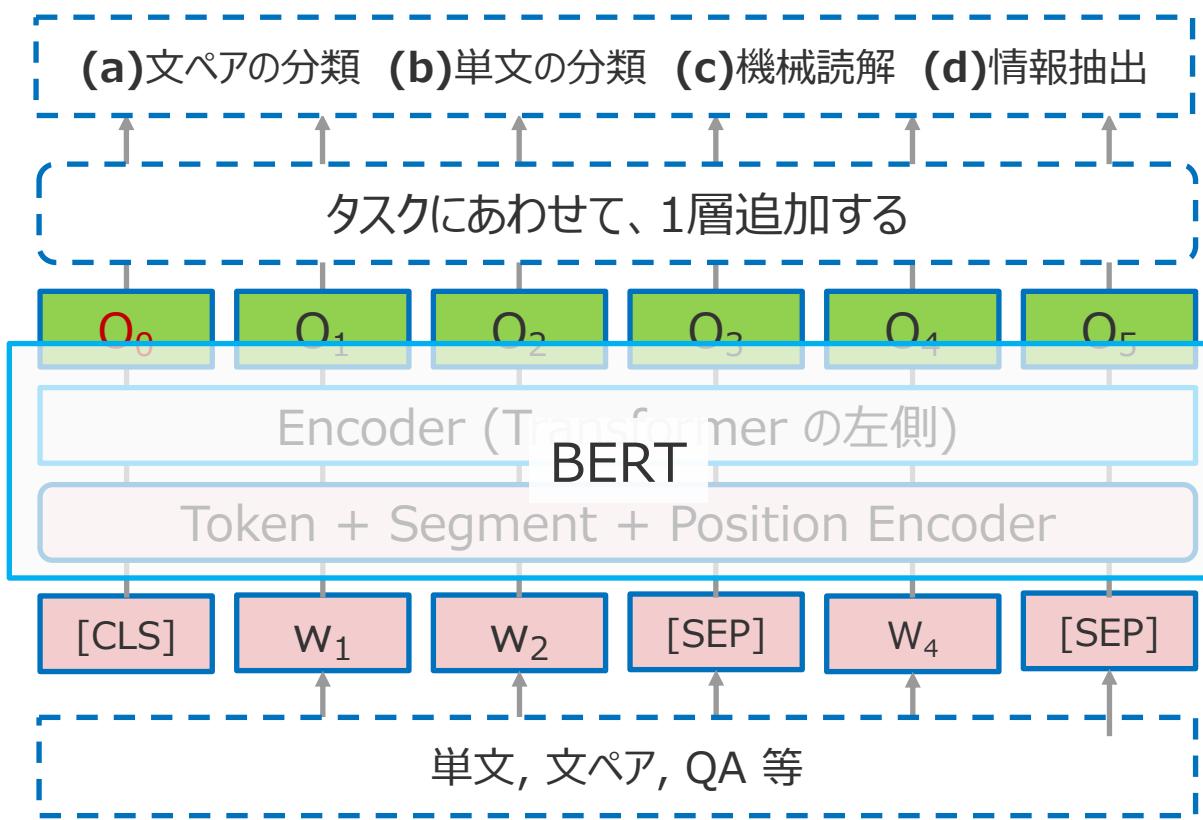
- 近年の基盤モデルの殆どがモデルの一部に Transformer を採用
- コンピュータビジョンの分野にも Transformer が高い性能を發揮



引用: [西田,2022] JSAI2022 チュートリアル講演資料の一部を修正して作成

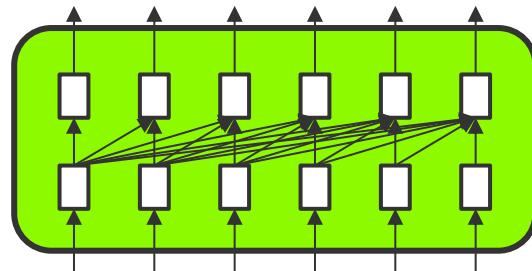
# BERT [Devlin+ (Google), NAACL19]

- 機械読解タスクで人間のスコアを超える、多数のNLPタスクで高性能を出し注目を浴びた
- 双方向 Transformer ブロックを24層重ねた言語モデル
- 出力層をタスク毎に1層のみ追加して、様々なタスクに適応できる



# GPT-3 [Brown+ (OpenAI), NeurIPS2020]

- GPT-1(1億パラメタ), GPT-2(15億パラメタ)と同じ自己回帰モデルだが超大規模
- 超大量(3000億トークン)のテキスト, 超巨大(96層)のTransformer デコーダで1750億※のパラメタを学習 (例: BERTは, 3.3億トークン, 24層, 3.4億パラメタ)
- タスクの説明もテキストとして入力し, 様々なタスク(マルチタスク)を実現
  - Zero-shot: タスク説明のみを与え全くサンプルを与えない
  - One-shot: タスク説明と1つのサンプルのみを与える
  - Few-shot: タスク説明と少数(10から100)のサンプルを与える



Transformer デコーダ

自己回帰型のNNモデル  
(出力を入力に戻す)

※ 2022年4月に Google が公開した PaLM は 5400億 [Chowdhery+, 2022]

# GPT-3 [Brown+ (OpenAI), NeurIPS2020]

The three settings we explore for in-context learning

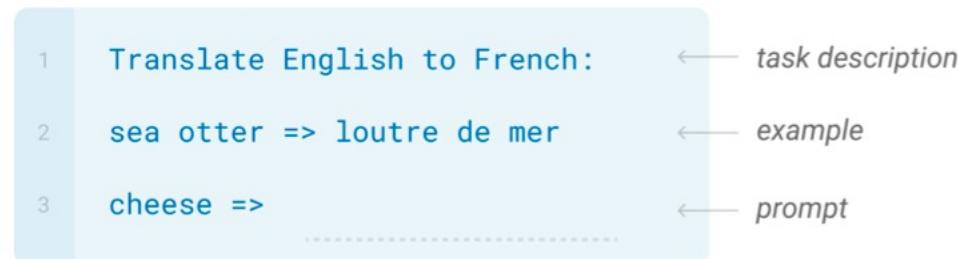
## Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.



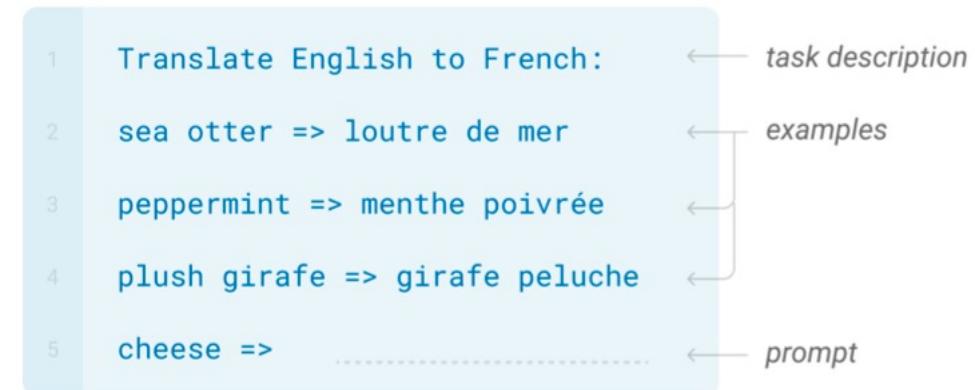
## One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.



## Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.



**In-context learning:**  
タスクをその場で学習する能力

# (再掲) ChatGPT — 基盤モデル GPT-3 をファインチューニングしたモデル

GPT3.5(2022年初期に学習)をベースに、人間の質問に答えるようにファインチューニングと、人間の人間の好みに合った出力出すように RLHF(人間のフィードバックに基づく強化学習)した、対話に特化した AI モデル

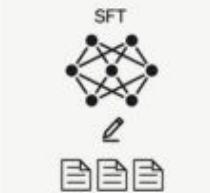
## Step1

人間の用意した望ましい回答で GPT3.5 を fine-tuning

A prompt is sampled from our prompt dataset.



A labeler demonstrates the desired output behavior.

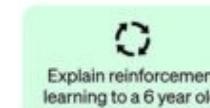


This data is used to fine-tune GPT-3.5 with supervised learning.

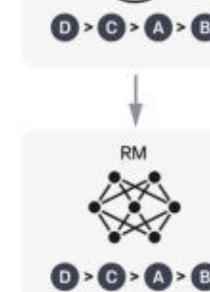
## Step2

Step1の出力に人間がランク付けし、報酬モデルを学習

A prompt and several model outputs are sampled.



A labeler ranks the outputs from best to worst.



This data is used to train our reward model.

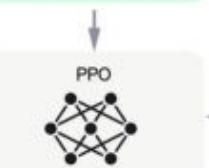
## Step3

Step2で学習した報酬モデルを使ってPPOで強化学習

A new prompt is sampled from the dataset.



The PPO model is initialized from the supervised policy.



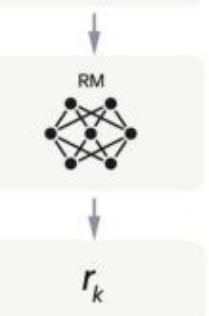
The policy generates an output.



The reward model calculates a reward for the output.



The reward is used to update the policy using PPO.



## 主な特徴

- 前の対話内容に続く質問への回答が可能
- 間違った回答をすることも多い(例: 肉じゃがのレシピを訊くと、ホワイトソースを入れようとした)
- 間違いを認めることもできる
- 正しくない前提に対する異議を唱えることもできる
- 不適切なリクエストには応じない
- スクリプトやコード生成も可能

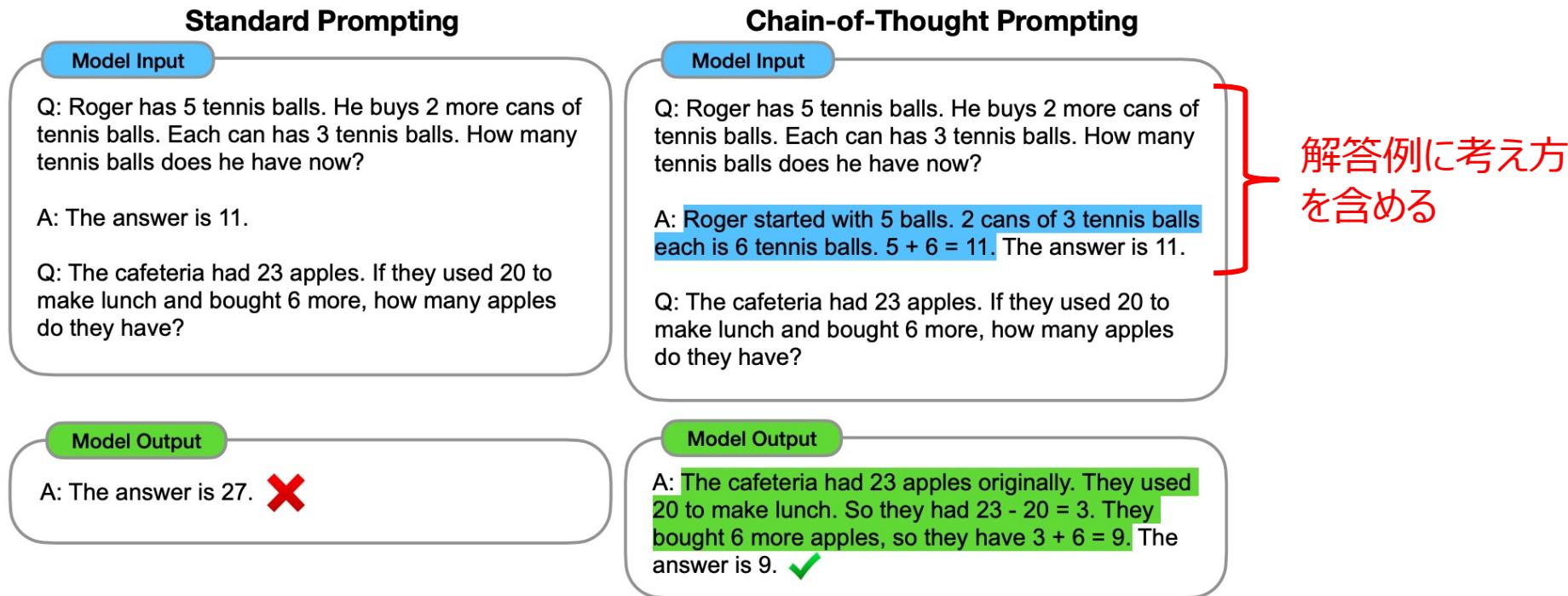
## 他のツールとの連携

- ChatGPT for Google: Google の検索結果に ChatGPT の出力を表示してくれる拡張機能
- ArxivGPT: Arxiv の検索結果に論文の要約やポイントを表示してくれるかく超機能

引用: <https://openai.com/blog/chatgpt/>

# プロンプトエンジニアリングの手法

- 「考え方」をプロンプトで与えることで、推論能力が大きく向上する
- Chain-of-Thought [[Wei\(Google\)+, 2022](#)]



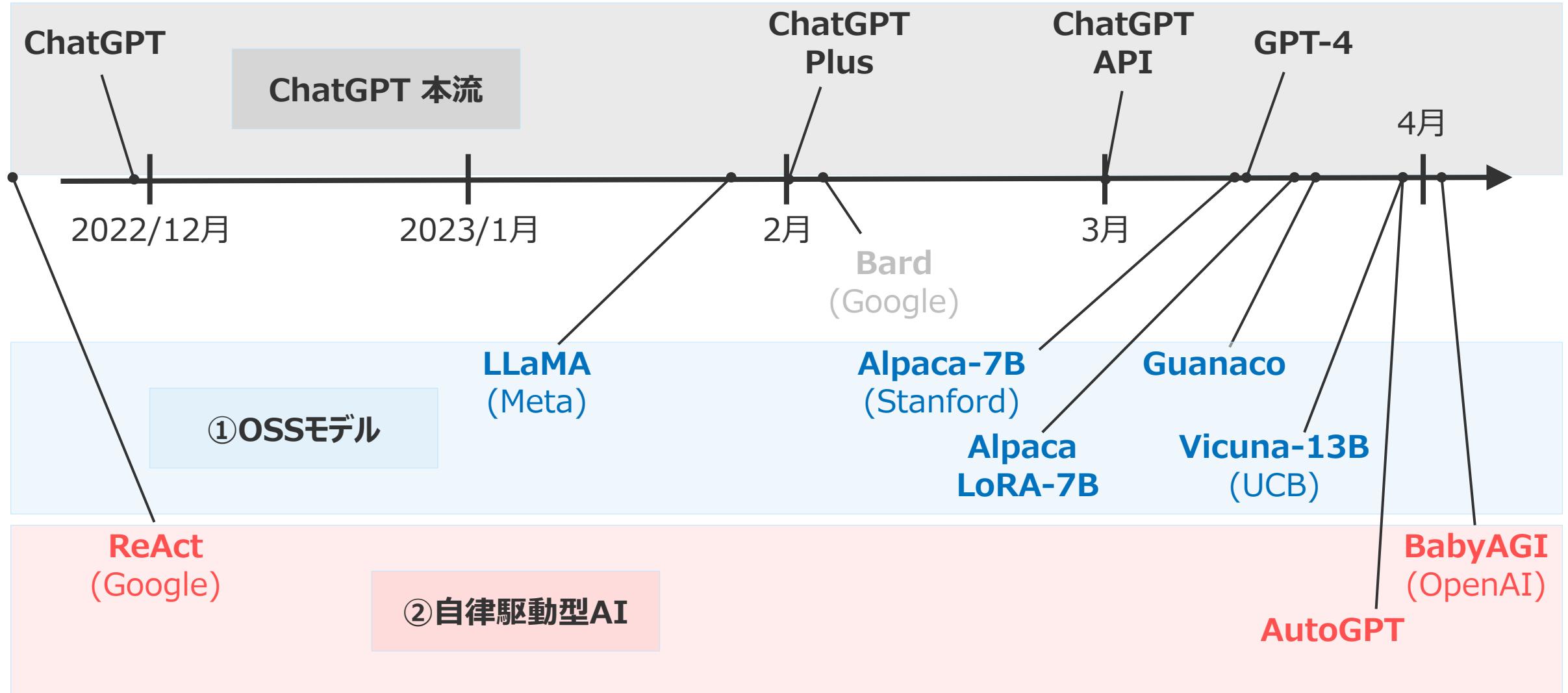
- 「ステップバイステップで」をプロンプトに追加すると計算などが正確になりやすい [[Kojima+, 2022](#)]

引用 <https://arxiv.org/pdf/2201.11903.pdf>

## ChatGPT 登場後の動向

# ChatGPT 登場以降

ChatGPT の登場(2022/11/30)以降、手元でLLMが動かせる「①OSSモデルやモデル生成技術」および外部のデータソースとの連携やプロンプト自動生成などの「②自律駆動型AI技術」などがあり、急速に進んでいる



# ChatGPT 登場以降 — 2つのトレンド

ChatGPT の登場以降のトレンドとしては、手元でLLMが動かせる「①OSSモデルやモデル生成技術」および外部のデータソースとの連携やプロンプト自動生成などの「②自律駆動型AI技術」などがあり、急速に進んでいると見られる

トレンド	説明	代表例
① OSSモデル	<ul style="list-style-type: none"><li>OpenAI の API を利用する場合、外部(OpenAI)にデータ送信することになるため、手元に LLM を構築するニーズがある</li><li>ChatGPT(175B)レベルの大規模モデルを載せるにはコンピューティングコストがかかりすぎ、パラメタを減らすと精度が下がる課題がある</li></ul> <p>→ AWS等のパブクラで動作可能な軽量で高精度のOSSモデルや学習手法が登場</p>	<ul style="list-style-type: none"><li>LLaMA(22/1、Meta)</li><li>Alpaca(23/3、Stanford)</li><li>Vicuna(23/3、UCBerkeley)</li></ul> など
② 自律駆動型AI	<ul style="list-style-type: none"><li>言語モデルは様々なタスクに応用されているが学習時点の(古い)知識しか利用できない</li><li>自身の内部表現を用いて推論の道筋を生成するため、反応的に探索・推論したり、知識を更新する能力が制限され、また一時的な記憶を持つこともできない</li></ul> <p>→ 行動と行動結果に対する推論を繰り返してタスクを達成させる仕組みが登場</p>	<ul style="list-style-type: none"><li>ReAct(22/11、Google)</li><li>AutoGPT(23/3)</li><li>BabyAGI (23/4/3、OpenAI)</li></ul> など

# 代表的な OSSモデル — 2023/5月末時点

2023/3月の Alpaca 登場以降、命令追従型学習によって GPT-3 を超える軽量モデルが相次いで登場した

分類	モデル名	アーキ	提供元	リリース	サイズ	ライセンス	日本語	例:「ペンギンはなぜ空を飛べないですか?」
クラウド	GPT-4	クローズド	OpenAI	2023/3/14	175B	有償API	◎	ペンギンが飛べない理由はいくつかあります。主な理由は、彼らの身体…
	<a href="#">Claude</a>	クローズド	Anthropic	2023/3/14	50B	有償API	◎	ペンギンは空を飛べない主な理由は次のとおりです。1. 翼がないため…
研究利用のみ	<a href="#">LLaMa</a>	LLaMa	Meta	2023/2/24	<a href="#">7-65B</a>	× 制限あり	△	ペンギンは空の中の風に乗り飛ぶことができます。
	<a href="#">Alpaca (LLaMa)</a>	LLaMa	Stanford	2023/3/13	<a href="#">7B</a>	× 制限あり	△	ペンギンは、空の下で彼は翼を叩くことができます。しかし、彼は午前…
	<a href="#">Alpaca-LoRA (LLaMa)</a>	LLaMa	Eric Wang	2023/3/17	<a href="#">7B</a>	× 制限あり	△	ペンギンは空を飛べないためには、彼は彼の背骨を抜く必要があります…
	<a href="#">Guanaco (LLaMa)</a>	LLaMa	Joseph Cheung	2023/3/19	<a href="#">7B</a>	× 制限あり	△	ペンギンが空を飛べない理由は、空を飛ぶことができないためです。ペ…
	<a href="#">GPT4All (LLaMa)</a>	LLaMa	Nomic AI	2023/3/29	<a href="#">13B</a>	× 制限あり	×	…
	<a href="#">Vicuna (LLaMa)</a>	LLaMa	LMSYS	2023/3/30	<a href="#">13B</a>	× 制限あり	○	ペンギンは、自然選択によって空を飛ぶことができないとされています。…
	<a href="#">Koala (LLaMa)</a>	LLaMa	BAIR	2023/4/4	<a href="#">7-13B</a>	× 制限あり	×	I'm sorry, but I'm not quite sure what you mean by …
	<a href="#">Stable-Vicuna (LLaMa)</a>	LLaMa	Stability AI	2023/4/28	<a href="#">13B</a>	× 制限あり	?	…
商用利用可能	<a href="#">Cerebras-GPT</a>	GPT-2	Cerebras	2023/3/28	<a href="#">13B</a>	Apache 2.0	△	ペンギンが空を飛べない理由としては、空を飛ぶことができないことが挙…
	<a href="#">Dolly2.0 (GPT-J-Alpaca)</a>	GPT-J	Databricks	2023/4/12	<a href="#">6B</a>	Apache 2.0	△	よくわかりませんが、ペンギンは空を飛べないのではなく、空を飛ぶことを…
	<a href="#">StableLM-Alpha</a>	GPTNeoX	Stability AI	2023/4/19	<a href="#">7B</a>	CC BY-SA-4.0	△	空を飛べるためには、空気を回し、目的地を知りません。ペンギンは、…
	<a href="#">GPT4All (GPT-J)</a>	GPT-J	Nomic AI	2023/4/24	<a href="#">6B</a>	Apache 2.0	×	It seems like you are having trouble with your writing. …
	<a href="#">RedPajama-INCITE</a>	GPTNeoX	Together	2023/5/5	<a href="#">3-7B</a>	Apache 2.0	△	ペンギンは空を飛べないのでしょうか。
	<a href="#">MPT-7B</a>	MPT	MosaicML	2023/5/5	<a href="#">7B</a>	Apache 2.0	△	もちろん、飛んでくれます！私は飛んでくれます！でも、飛んでくれない…
日本語特化 (商用利用可能)	<a href="#">open-calm</a>	GPTNeoX	CyberAgent	2023/5/16	<a href="#">1-7B</a>	CC BY-SA-4.0	△	ペンギンが空を飛べない理由は、水中で息ができないから、というのを…
	japanese-gpt-neox-3.6b	GPTNeoX	rinna	2023/5/17	<a href="#">3.6B</a>	MIT	△+	ペンギンは、翼のような推進力を生み出す筋肉がなく、空気力学的に…

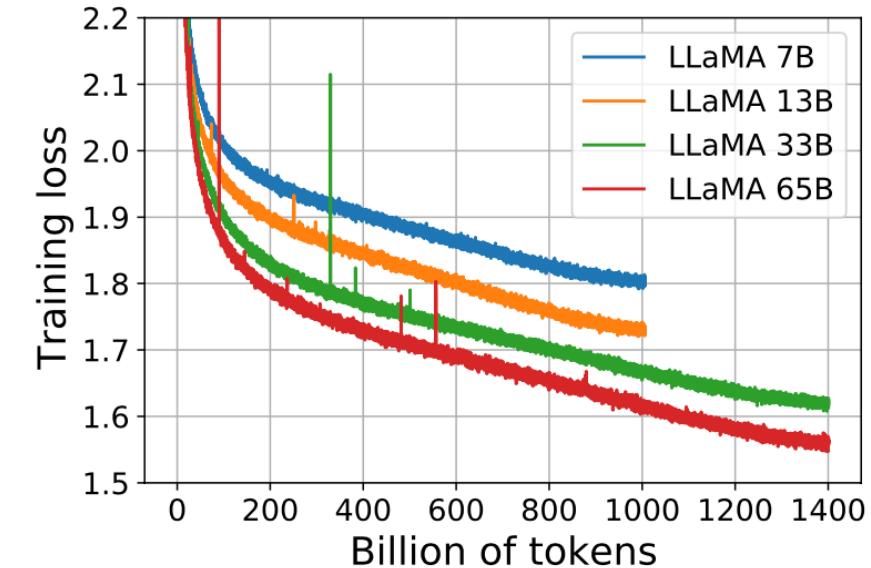
# 【工学系向け】LLaMA [[Touvron+ \(Meta\), 2023/2/27](#)]

コーパスを大きくすることで小さいモデルでも GPT-3 の性能を上回ることを報告 → 1.4兆のトークンでの学習 (データの収集方法・細かいモデル構造を改良)により、13BモデルでGPT-3(175B)を上回り、65BモデルでPaLM(540B)に匹敵する性能

		BoolQ	PIQA	SIQA	HellaSwag	WinoGrande	ARC-e	ARC-c	OBQA
GPT-3	175B	60.5	81.0	-	78.9	70.2	68.8	51.4	57.6
Gopher	280B	79.3	81.8	50.6	79.2	70.1	-	-	-
Chinchilla	70B	83.7	81.8	51.3	80.8	74.9	-	-	-
PaLM	62B	84.8	80.5	-	79.7	77.0	75.2	52.5	50.4
PaLM-cont	62B	83.9	81.4	-	80.6	77.0	-	-	-
PaLM	540B	<b>88.0</b>	82.3	-	83.4	<b>81.1</b>	76.6	53.0	53.4
LLaMA	7B	76.5	79.8	48.9	76.1	70.1	72.8	47.6	57.2
	13B	78.1	80.1	50.4	79.2	73.0	74.8	52.7	56.4
	33B	83.1	82.3	50.4	82.8	76.0	<b>80.0</b>	<b>57.8</b>	58.6
	65B	85.3	<b>82.8</b>	<b>52.3</b>	<b>84.2</b>	77.0	78.9	56.0	<b>60.2</b>

Table 3: Zero-shot performance on Common Sense Reasoning tasks.

- LLaMA 130B は、ほとんどのベンチマークでGPT-3(1750B)を上回る
- LLaMA 650B は、Chinchilla 700B、および PaLM 5400B に匹敵

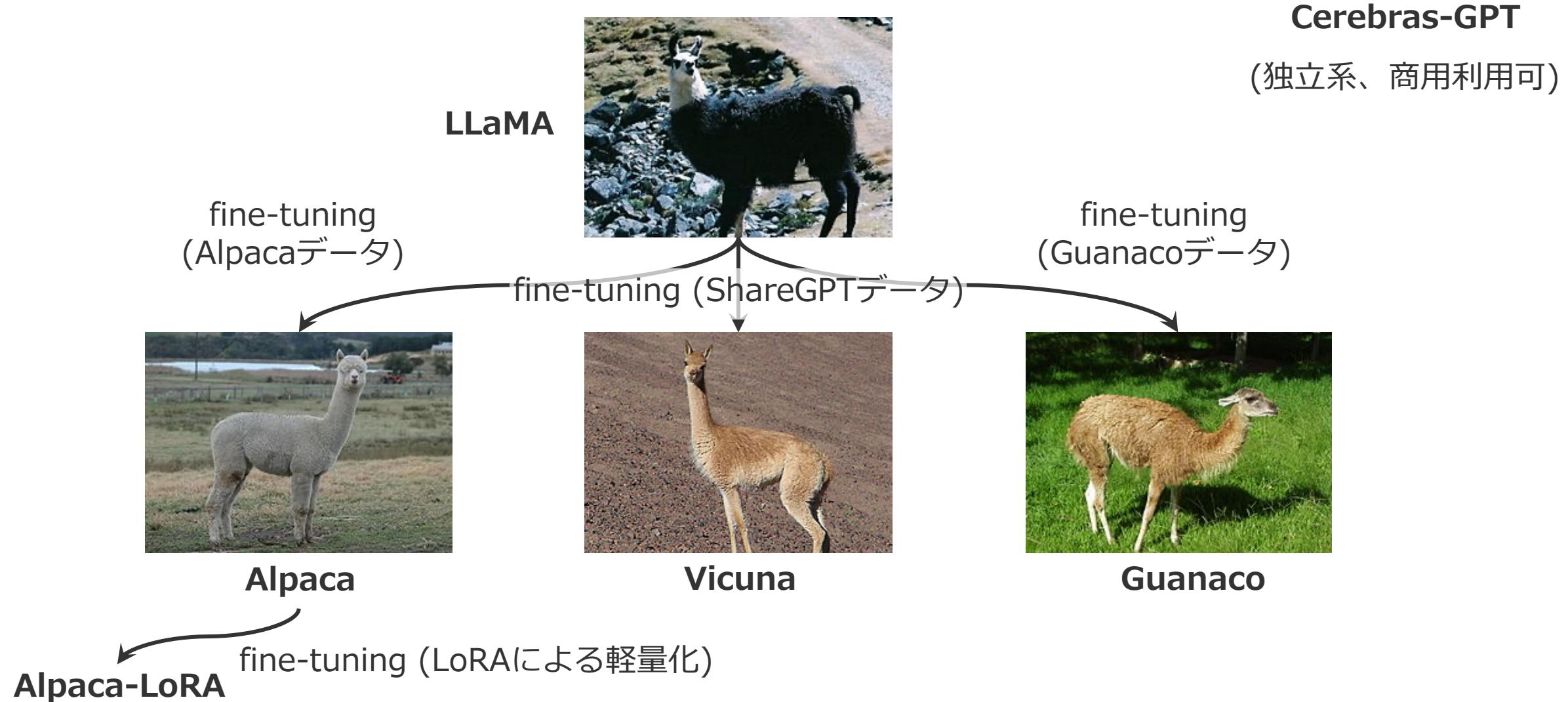


- LLaMA は、7B～65Bまで4種類が存在 (7Bと13Bは1兆、33Bと65Bは1.4兆のトークンで学習)
- 7～13Bであれば1枚のGPUで動作可能
- 公開されているデータセットを使ってトレーニングされているので検証可能

引用: <https://arxiv.org/pdf/2302.13971.pdf>

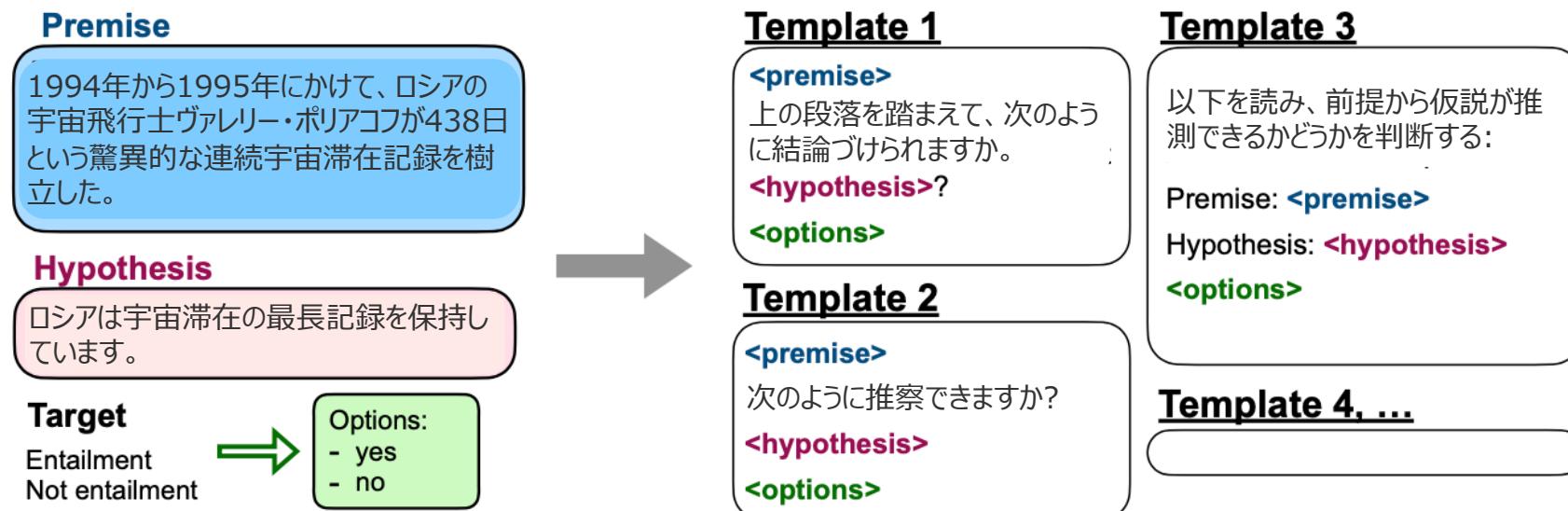
# 【工学系向け】LLaMA と 派生モデル

2023/3月の Alpaca 登場以降、同じ命令追従型学習によって GPT-3 を超える軽量モデルが相次いで登場した



# Instruction Tuning (FLAN) [Wei+,2022]

- GPT-3 など言語モデルの構造を変えずに、**複数のタスク**でファインチューニングする方法
- タスク毎にテンプレートを用意し「プロンプト(タスクの指示と事例)+出力」の形式に変換した学習データで言語モデルを追加学習する (=Instruction Tuning)
  - ゼロショットで解くタスクにおいて、GPT-3よりも高い精度を達成
  - 未知のタスクや指示に対しても精度よくテキストを生成

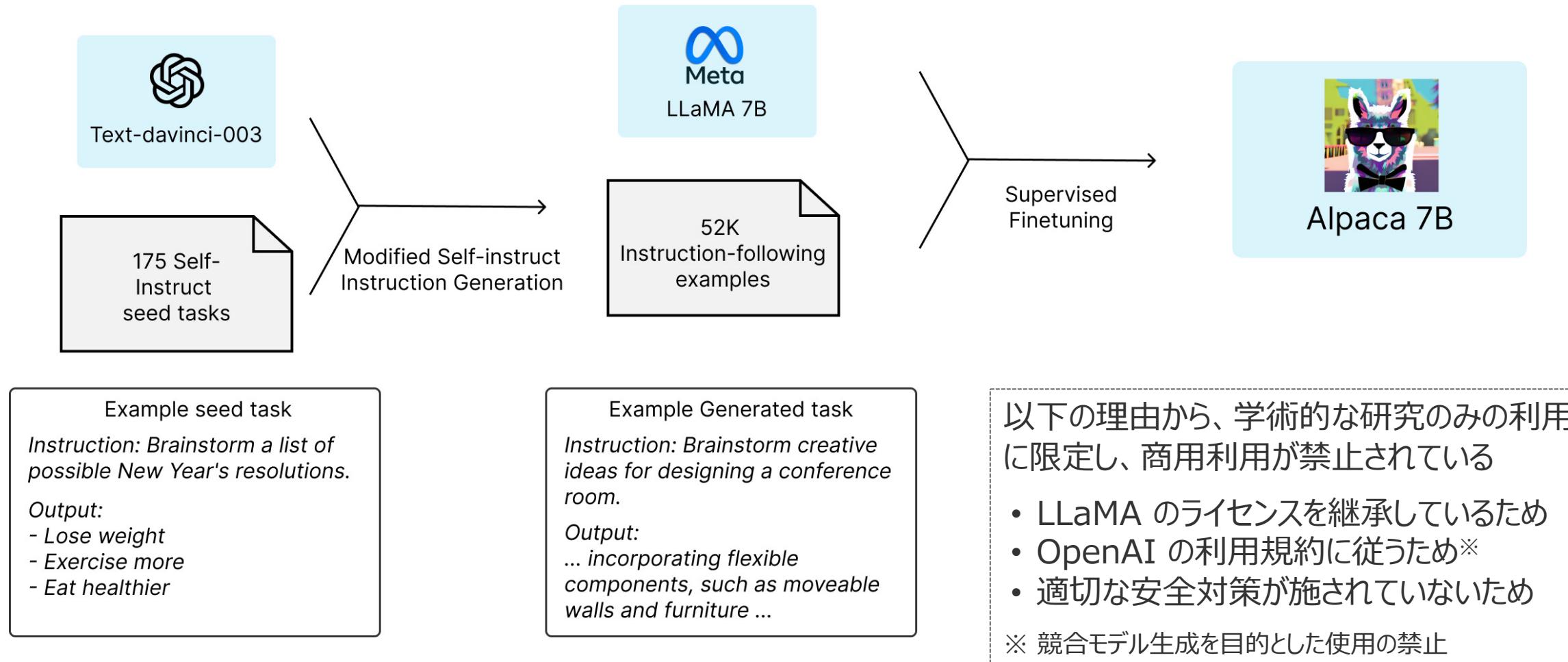


図：自然言語推論(NLI)タスクの例（前提出文が仮設を含意するか否かを自動判定するタスク）

引用: <https://arxiv.org/pdf/2109.01652.pdf>

# 【工学系向け】Alpaca [Taori+ (Stanford Univ.), 2023/3/13]

GPT-3 で生成した命令追従型のデータセットで LLaMA (Meta)をファインチューニング(Instruction Tuning)したモデルで、はるかに小さな環境で簡単・安価に GPT-3 に近い性能の再現ができる



引用: <https://crfm.stanford.edu/2023/03/13/alpaca.html>

# 【工学系向け】Alpaca [[Taori+ \(Stanford Univ.\), 2023/3/13](#)]

Alpaca はシードとなる175個の Instruct データ（命令追従型データ）から、text-davinci-003 (ChatGPT) を使って、52K の self-instruct データを生成し、それをファインチューニングに用いる

## ● Instruct データのサンプル

### フォーマット

Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request.

### Instruction:  
{instruction}

### Input:  
{input}

### Response:

### seed\_task\_1

以下は、タスクを説明する命令と、さらなるコンテキストを提供する入力の組み合わせです。要求を適切に満たすような応答を書きなさい。

### Instruction:  
与えられたペアの間にはどのような関係があるのか？

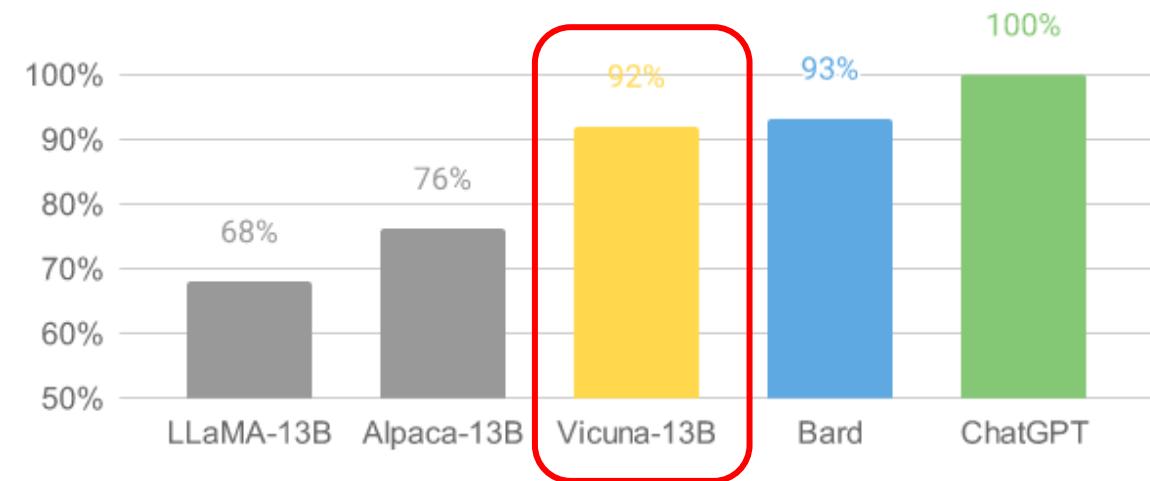
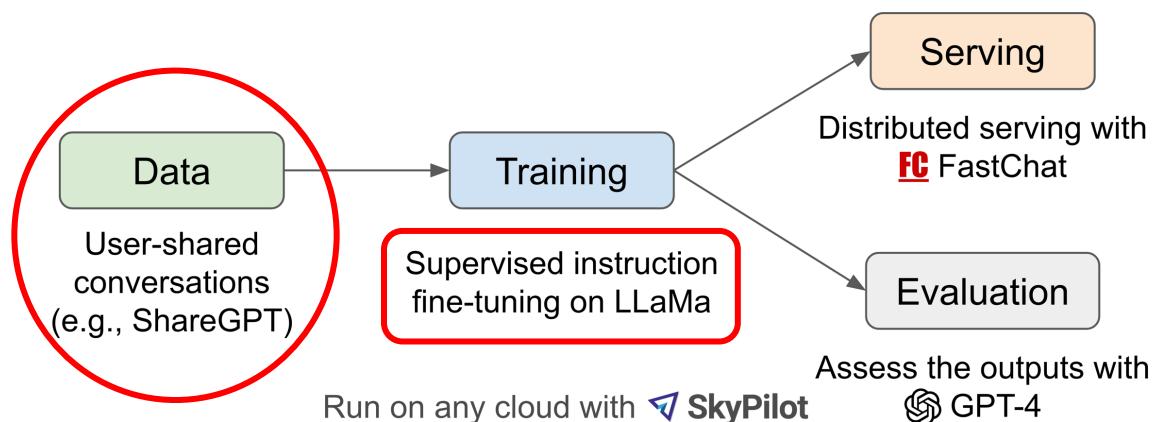
### Input:  
夜:昼 :: 右:左

### Response:  
与えられたペアの関係は、それらが反対であることです。

# 【工学系向け】Vicuna-13B [The Vicuna Team, 2023/3/31]

Vicuna-13B は **ChatGPTのプロンプトをシェアできるChrome拡張機能「ShareGPT」** のデータを用いて LLaMAのベースモデルを微調整することにより、Alpaca 7B などの他のオープンソースの大規模言語モデルよりも高品質なパフォーマンスを実現

各種チャットAIの応答品質を評価した結果、ChatGPTを100%とした場合、LLaMAが68%、Alpaca 7Bが76%だった一方で**Vicuna-13Bの品質は92%**に迫った



モデル	データセット	学習コード	評価指標	学習コスト (7B)	学習コスト (13B)
LLaMA	公開データセット (1T トークン)	N/A	ベンチマーク	82K GPU-hours	135K GPU-hours
Alpaca	davinci-003 API で Self-instruct (52K サンプル)	利用可能	著者による評価	\$500 (データ) + \$100 (学習)	N/A
Vicuna	ShareGPT 等の会話データ (70K サンプル)	利用可能	GPT-4 比較	\$140 (学習)	<b>\$300 (学習) ≈ 2</b>
Bard/ChatGPT	N/A	N/A	ミックス	N/A	N/A

引用: <https://lmsys.org/blog/2023-03-30-vicuna/>

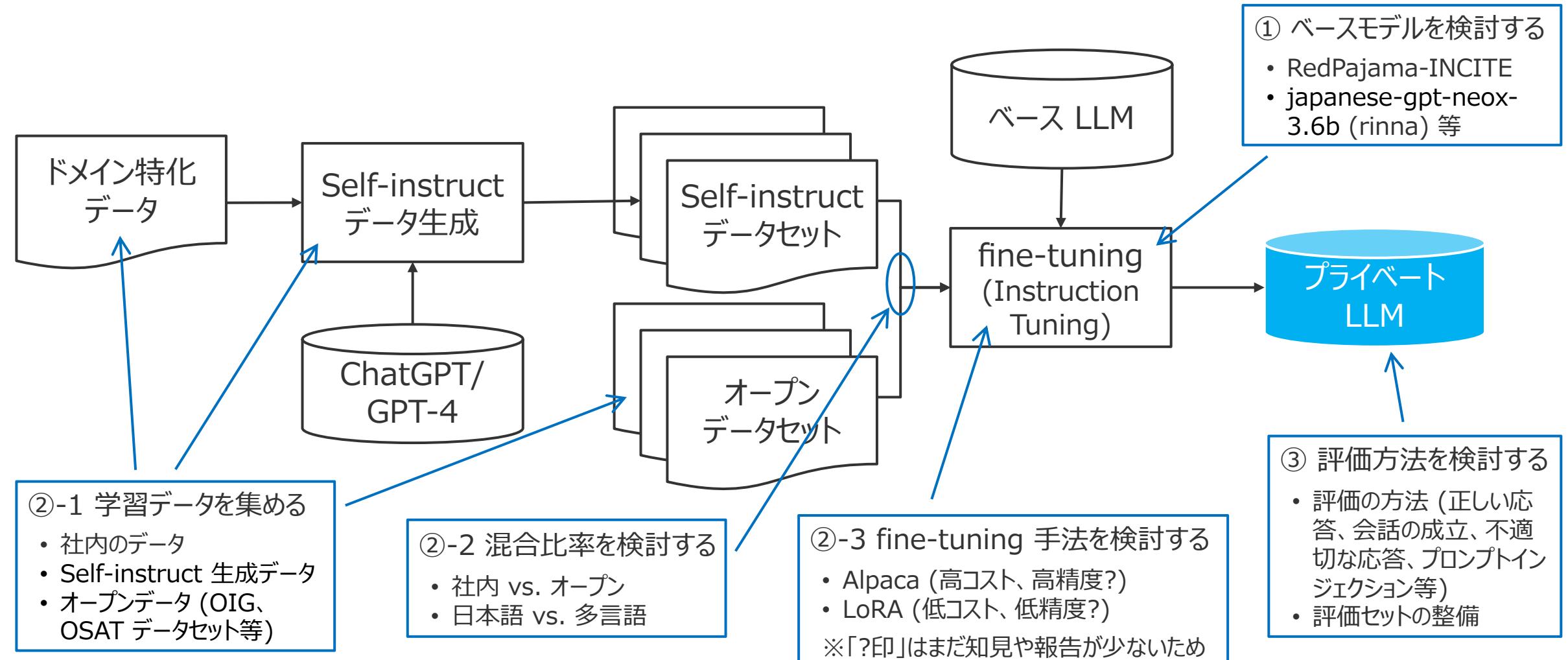
※2 SkyPilot スポット・インスタンスを活用し、7Bモデルのトレーニング費用は500ドルから140ドル程度に、13Bモデルのトレーニング費用は1000ドルから300ドル程度に削減

# LLM を利用するいくつかの方法

	フルスクラッチLLM	独自LLM	API 利用
説明	自らLLMをスクラッチ開発	OSSモデルをファインチューニング	LLM提供ベンダーのAPIを利用する
メリット	<ul style="list-style-type: none"> <li>公開されている利用可能な基盤モデルではパラメータサイズが不十分・過剰</li> <li>特定の言語やドメインの知識のみのデータからなるモデルを作れる</li> <li>自社で保有する大量のデータを活用できる（例：コールセンターログ）</li> <li>出力結果の品質をコントロールできる</li> </ul>	<ul style="list-style-type: none"> <li>オンプレミスで独自LLMを運用することで、情報漏洩リスクを低減可能</li> <li>プラットフォーム固有の機能、用語、コンテキストなど特定のニーズや要件に合わせてモデルを調整可能</li> <li>出力結果の品質を多少はコントロールできる</li> </ul>	<ul style="list-style-type: none"> <li>ChatGPT/GPT-4 の高精度が得られる</li> <li>データ管理・学習/運用環境の管理が不要</li> </ul>
デメリット	<ul style="list-style-type: none"> <li>ChatGPT/GPT-4 のような高精度なモデルを構築することは難しい</li> <li>モデルサイズの大きさから、独自に大規模な学習/運用環境が必要（コスト高）</li> </ul>	<ul style="list-style-type: none"> <li>ChatGPT/GPT-4 のような高精度なモデルを構築することは難しい</li> <li>独自に学習/運用環境が必要（フルスクラッチに比べ低成本）</li> </ul>	<ul style="list-style-type: none"> <li>カスタマイズの余地が少ない</li> <li>一握りのLLMプロバイダーへの依存により、サービス停止の可能性がある</li> </ul>
コスト例 (\$1=135円換算)	<ul style="list-style-type: none"> <li>LLaMA 65B のスクラッチ学習: A100 80GB × 2048枚 × 21日間 = 約5.5億円</li> </ul>	<ul style="list-style-type: none"> <li>Alpaca 13B のファインチューニング: A100 80GB × 8枚 × 3時間 = 約20万円以下</li> </ul>	<ul style="list-style-type: none"> <li>OpenAI のAPI料金(1Kトークン): gpt-3.5-turbo 0.27円, GPT-4(8K) 6.08円, GPT(32K) 12.15円</li> </ul>

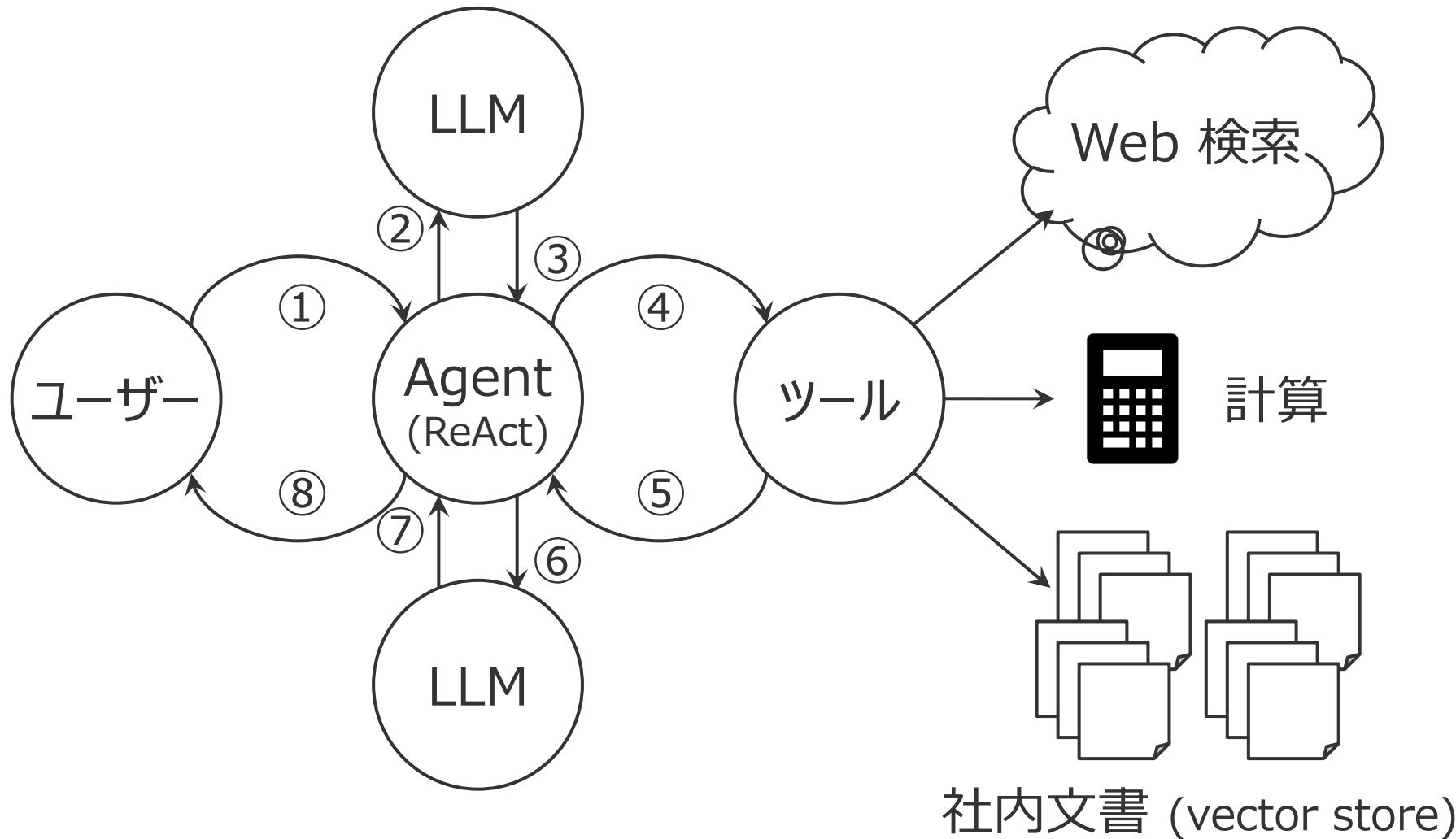
# 【工学系向け】独自LLM構築の難しさ

ChatGPT/GPT-4 に近い十分な応答精度を得るために「**学習データの種類(言語やタスク)や組み合わせ(比率含む)**」や「**学習データとfine-tuning方法(AlpacaやLoRA等)**」の組み合わせ等、未解明な問題が多い



# 自律駆動型AI とは

自律駆動型AI が **LLM** と外部リソース(データソースや言語処理系等)とのやり取りを介して**タスクを達成する**

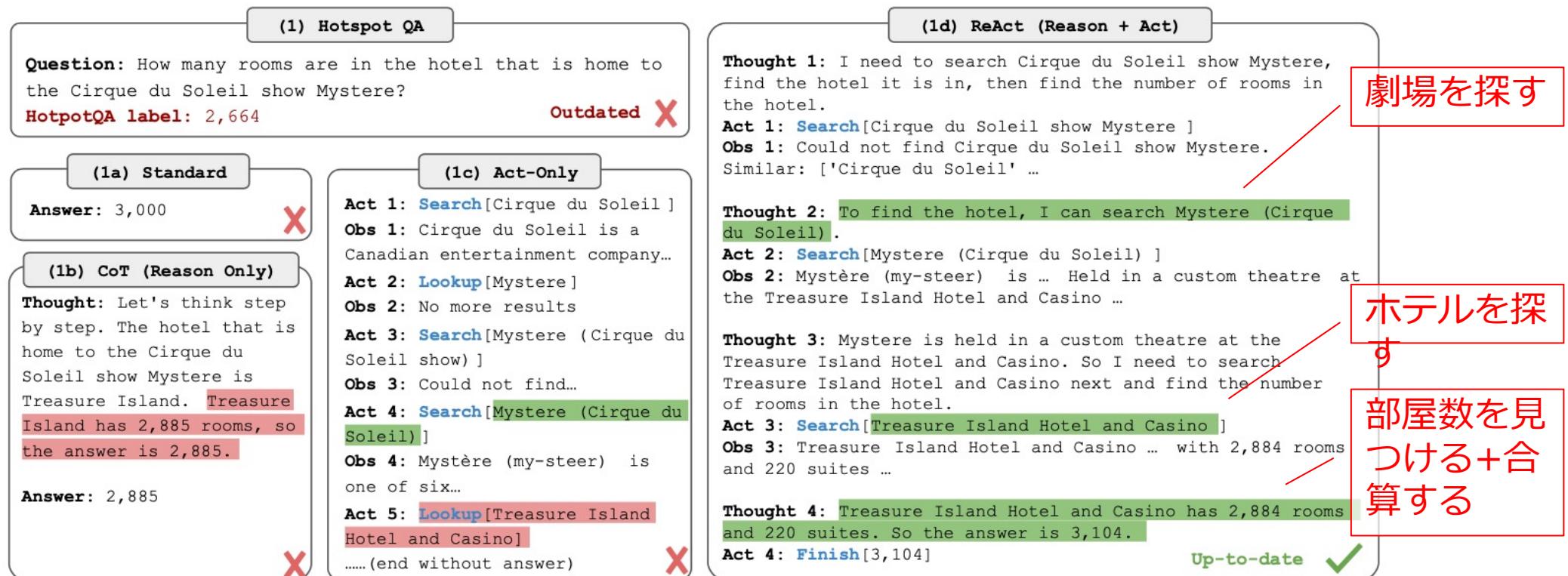


1. ユーザーが質問する
2. プロンプトとともにLLMに質問が送られる
3. LLMは、ユーザーに即座に回答するか、追加情報を得るためにツールを使用する指示を出して回答する
- 4,5. 追加情報を取得する
- 6,7,8. LLMは、追加の文脈に基づいて最終的な答えを構築する

# 自律駆動型AI が利用する技術

- **ReAct** [Yao+’23, ICLR] は、LLM による中間推論過程で思考と行動のフェーズを導入し、組み合わせた回答が可能  
思考フェーズ：モデルが現在の中間推論状態に対する状況を記述する  
行動フェーズ：外部知識資源との接続を行い、外部 API などを実行する
- **LangChain** は、LLMと外部リソース(データソースやツール)を組み合わせたアプリケーションの開発支援を目的として Harrison Chase 氏が開発したPythonライブラリで、ReAct によるツールの自動選択が可能な Agent 実装を含む

右図：  
ReAct 論文  
中の例



引用: <https://arxiv.org/abs/2210.03629>

# 自律駆動型AI はどうやって動いているか

**Input:** レオ・ディカプリオの恋人は誰でしょう？彼女の現在の年齢は0.43乗すると何歳でしょう？

## prompt:

次の質問に、できる限り答えてください。あなたは以下のツールにアクセスすることができます：

Search: 検索エンジン。時事問題に関する質問に答える必要があるときに便利です。入力は検索クエリである必要があります。

Calculator: 数学に関する質問に答える必要があるときに便利です。

次のような書式で入力します：

**質問:** 答えなければならない入力された問題

**思考:** 常に何をすべきかを考えるべき

**行動:** 取るべき行動 [Search, Calculator] のいずれかでなければならない

**行動の入力:** アクションへの入力

**観察:** 行動の結果

... (この思考・行動・行動の入力・観察は、N回繰り返すことができます)

**思考:** 最終的な答えがわかった

**最終回答:** 入力された元の質問に対する最終的な回答

始める！

**質問:** レオ・ディカプリオの恋人は誰でしょう？彼女の現在の年齢は0.43乗すると何歳でしょう？

**思考:**

LLMへの入力

レオ・ディカプリオのガールフレンドが誰なのか、彼女の現在の年齢を調べたい。  
そして、彼女の年齢を0.43乗で計算してみる。

**行動:** Search

**行動の入力:** レオ・ディカプリオの恋人とその年齢

Searchの  
実行

**観察:** レオナルド・ディカプリオは、恋人のカミラ・モローネと数ヶ月で別れた後、彼の恋愛に関する長年の説を証明したようだ ...

**思考:**

LLMへの入力

レオ・ディカプリオが最近、恋人のカミラ・モローネと別れたことがわかった。0.43乗の年齢を計算する前に、彼女の年齢を知る必要があります。

**行動:** Search

**行動の入力:** カミラ・モローネ年齢

Searchの  
実行

**観察:** 25 years

**思考:**

LLMへの入力

Camila Morroneは25歳です。これで、彼女の年齢を0.43乗で計算できる。

**行動:** Calculator

**行動の入力:** 25 \*\* 0.43

Calculatorの  
実行

**観察:** 回答: 3.991298452658078

**思考:**

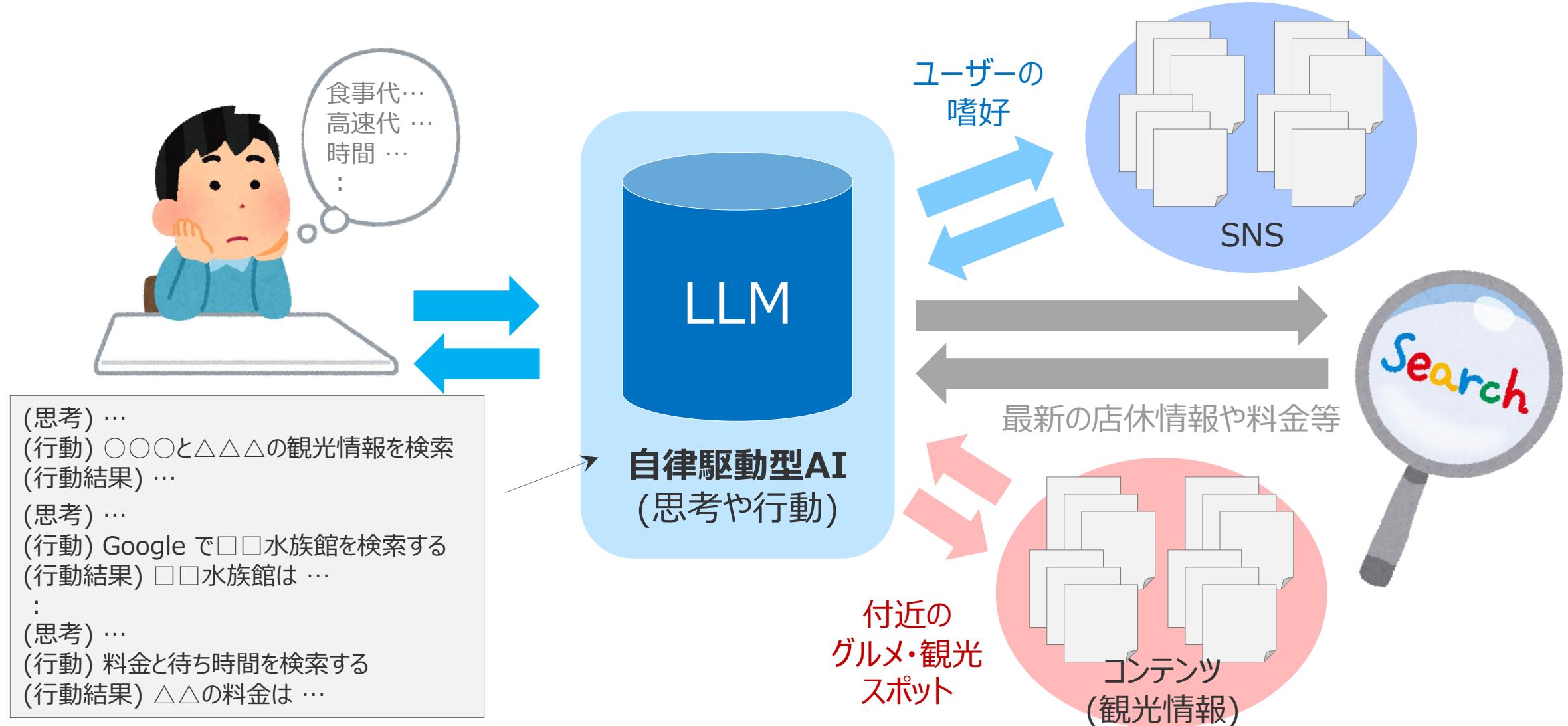
LLMへの入力

カミラ・モローネの年齢を0.43乗で計算すると、約3.99になります。

**最終回答:** カミラ・モローネの年齢を0.43乗すると約3.99になります。

# 自律駆動型AI の活用イメージ (ドライブプランの提案)

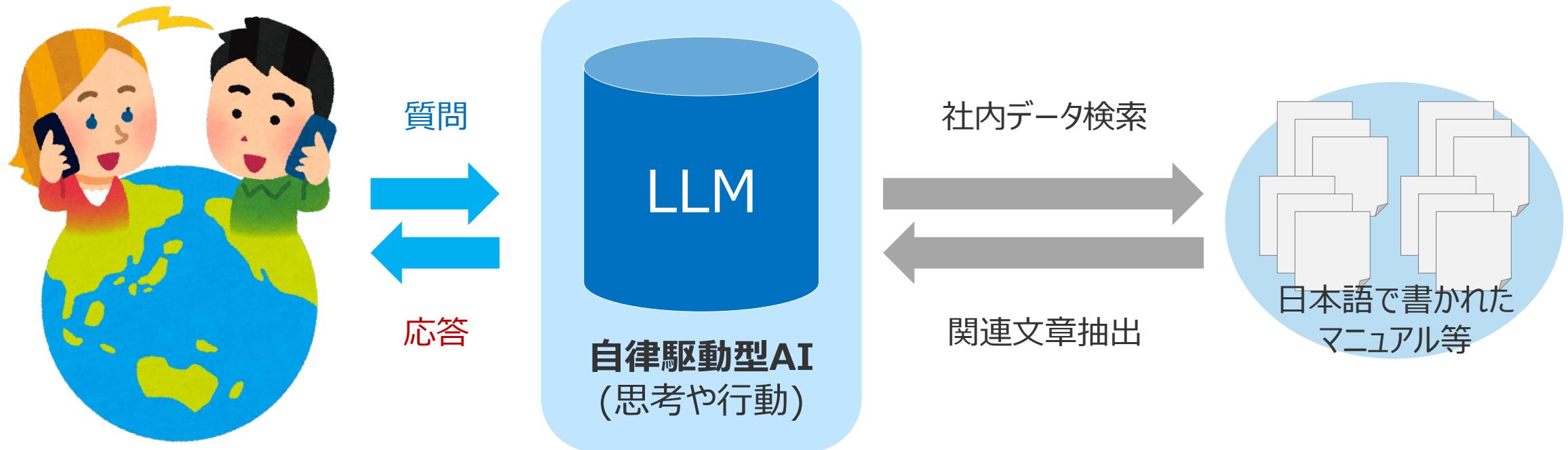
自律駆動型のAI により、LLM と社内外のデータソース(検索エンジンや観光情報などの外部コンテンツ)を組み合わせることで、ユーザーのリクエストにあわせた up-to-date なドライブプランの作成と、きめ細かい情報提供を行う



# 自律駆動型AI の活用イメージ (マニュアルに対する多言語QA①)

自律駆動型AI により、LLM と社内データ(社内規程や業務マニュアル等)のみで、質問に回答する  
→ 外部の知識(LLM内の知識や検索エンジンから得た情報)によるハルシネーション(誤った回答)を低減する

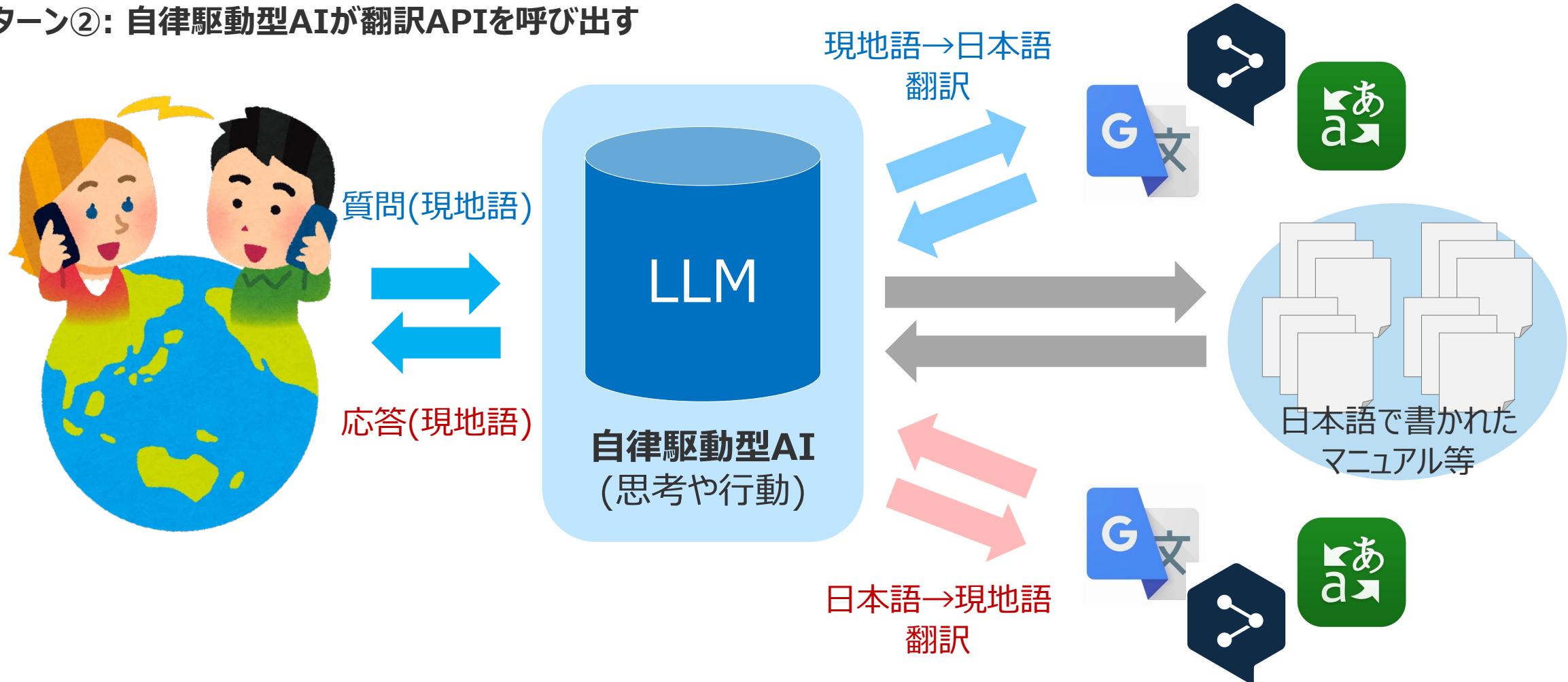
## パターン①: LLMが多言語にも対応してくれる



# 自律駆動型AI の活用イメージ (マニュアルに対する多言語QA②)

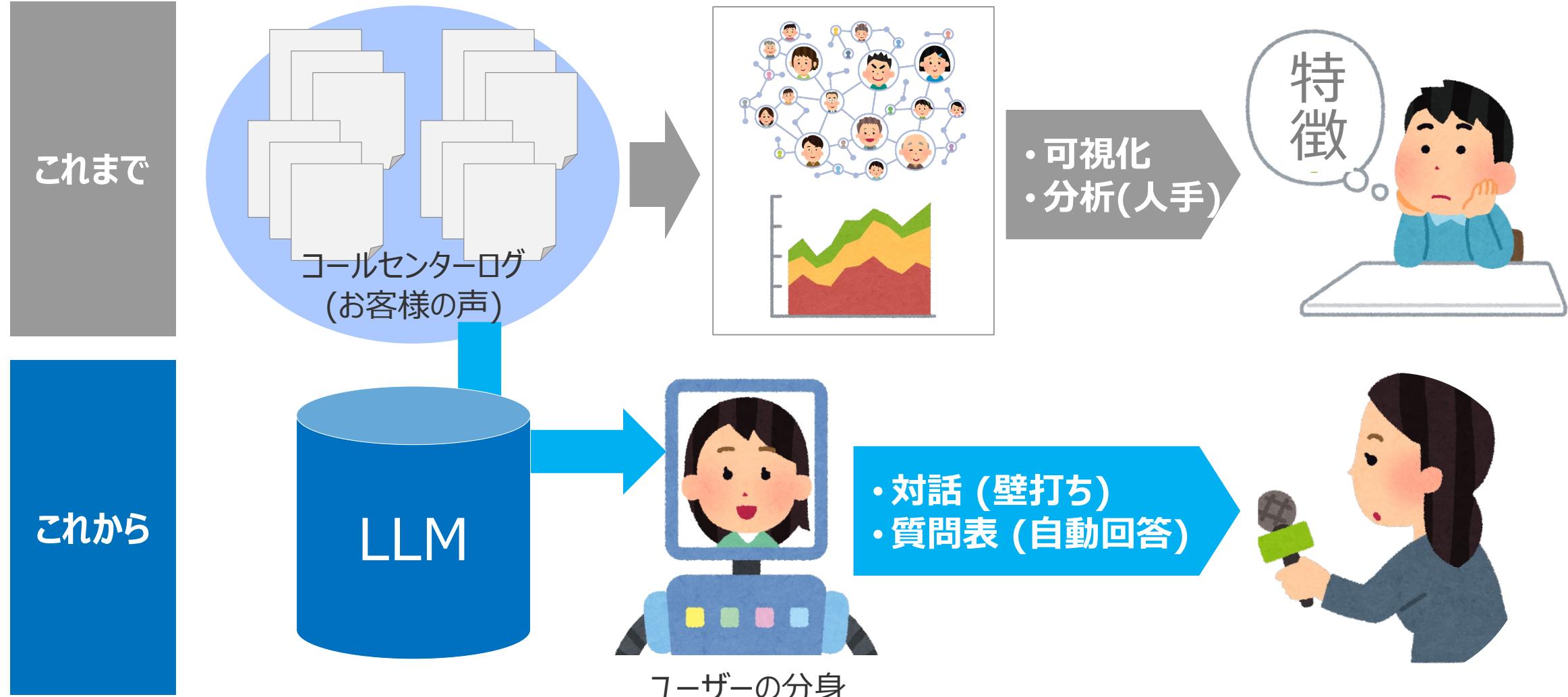
自律駆動型AI により、LLM が外部の翻訳APIと連携し、社内データ(社内規程や業務マニュアル等)のみで多言語の質問に回答する → 翻訳APIは、自律駆動型AIから必要に応じて自動的に呼び出される

## パターン②：自律駆動型AIが翻訳APIを呼び出す



# テキストマイニングの未来予想図 (お客様の声の分析の例)

LLM がユーザー(あるいはユーザーの集合)になりきり(=分身)、ユーザーの分身である LLM との対話やヒアリングを通して、ユーザー(あるいはユーザーの集合)の潜在的なニーズや隠された課題を探る → テキストマイニングの進化形



# まとめ

- ChatGPT の登場
  - 従来の自然言語処理は**タスクごとにモデルを学習**していた
  - ChatGPT ひとつで様々な自然言語処理タスクが解ける → 大きな衝撃
  - ChatGPT は、**大規模言語モデル**を人間が好む答えを出すよう追加学習したもの
- 大規模言語モデルの成り立ち
  - 従来の言語モデルは**文章を生成**(次の単語を予測)していた
  - **Transformer**で**文章生成する仕組みを超大規模化** → 知識と読解力を得た
- ChatGPT 登場後の動向
  - ChatGPT に迫る性能を出す **OSS モデル**や学習手法が相次いで登場中
  - **ChatGPT を利用してタスクを達成**する Agent の利用が広がる (予想)

# day 1 – レポート課題

- 以下を PDF ファイルで提出してください
  - ChatGPT と、Google Bard または Microsoft Bing AI Chat のどちらかまたは両方のツールを使って、以下の質問を試してください
    1. 日本の総理大臣は誰ですか？
    2. アメリカの大統領は誰ですか？
  - 応答内容の違いを観察し、違いの理由を考察(感想も可)を文章で記述してくださいアカウント作成できない等、何らかの事情で2つ以上のツールが試せない場合、本日の講義の感想を文章で記述してください

レポート形式	提出先	期限
PDF	manaba	次回～18:20

- 【参考情報】

  - ChatGPT の始め方: <https://book.st-hakky.com/docs/chatgpt-register/>
  - Google Bard の始め方: <https://www.gizmodo.jp/2023/05/what-is-google-bard.html>
  - Bing AI Chat の始め方: <https://www.sedesign.co.jp/dxinsight/bing-ai-chat#article-2>

## Q&A

# (参考) 無償で利用できる機械学習環境

- 近年、機械学習の教育・研究を目的とした研究用ツールが相次いで登場

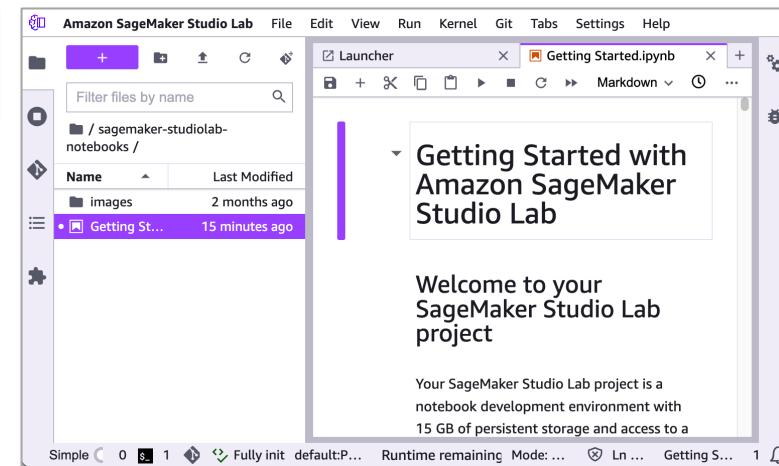
 Colaboratory

<https://colab.research.google.com>



 Amazon SageMaker Studio Lab

<https://studiolab.sagemaker.aws/>



演習で使用  
↓

	Colab(無償版)	Studio Lab
GPU	T4(16GB)	T4(16GB)
最長実行時間	12時間	CPU:12時間 GPU:4時間
メモリ	12GB	15GB
ディスク	CPU:100GB GPU:78GB	15GB (永続化)
ターミナル	×	○
ランタイムの保存と再開	×	○
費用	無償	無償
その他	Googleアカウントが必要	AWSアカウントは不要 (クレカ不要)

# SageMaker Studio Lab のアカウント作成

- <https://studiolab.sagemaker.aws/> にログインして、アカウントを作成してください

The screenshot shows the 'Request account' page of the Amazon SageMaker Studio Lab website. The page has a light blue background with abstract line patterns. At the top, there's a 'Sign in' link and a 'Request account' button, which is highlighted with a red box and a red arrow pointing to it from the top right. Below that is a large input field labeled 'Request account'. The form contains several text input fields: 'Enter your email\*', 'Enter your first name', 'Enter your last name', 'Select your country', 'Enter your company or organization name', 'Select your occupation', and 'Why are you interested in Amazon SageMaker Studio Lab?'. At the bottom, there's a yellow button labeled 'Enter referral code' which is also highlighted with a red box and a red arrow pointing to it from the bottom left. A purple 'Submit request' button is at the very bottom.

## アカウント作成手順

1. [アカウント作成フォーム](#)からアカウントの申し込みを行う  
**注意:** リファラルコードをアカウント作成フォームに忘れずに入力ください
2. 「Account request confirmed ...」のメールを受信し、メール内のリンクからアカウントを作成する  
→ リクエストの受付はすぐにメールが届きます
3. 「Verify your email ...」のメールを受信し、メール内のリンクからメールアドレスを認証する  
→ リファラルコードを利用している場合は2~3分以内に結果が届きます
4. 「Your account is ready ...」のメールを受信する  
→ これで「Sign in」できます

※ リファラルコードの有効期間: 2023/6/30 ~ 2023/7/7

## 参考資料 (2/2)

- [Honda+,2022] Shion Honda and Hidehisa Arai, "AI開発の新たなパラダイム「基盤モデル」とは." Recruit Data Blog (accessed 25 Jan 2023).
- [OpenAI,2023] OpenAI, R. "Introducing ChatGPT." <https://openai.com/blog/chatgpt> (accessed 5 April 2023).
- [OpenAI,2023] OpenAI, R. "GPT-4 technical report." arXiv (2023): 2303-08774.
- [Shohei N.,2023] Shohei N. "[比較表] Azure OpenAIと本家OpenAI APIの比較表." <https://zenn.dev/microsoft/articles/e0419765f7079a> (accessed 5 April 2023).
- [中島,2023] 中島佑允. "大規模言語モデルを活用するために知っておきたいこと." AI-SCHOLAR主催「What is GPT」講演資料 (2023).
- [Lambersy,2022] "How GPT-3 Writing Tools Work & 4 Things To Be Careful With When Using Them." <https://www.textcortex.com/post/how-gpt-3-writing-tools-work> (accessed 5 April 2023).
- [Harris+,1954] Harris, Zellig S. "Distributional structure." Word 10.2-3 (1954): 146-162.
- [Lin,2002] Lin, Jeng-Jong. "Applying a co-occurrence matrix to automatic inspection of weaving density for woven fabrics." Textile research journal 72.6 (2002): 486-490.
- [Mikolov+,2013] Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." arXiv preprint arXiv:1301.3781 (2013).
- [Mikolov+,2010] Mikolov, Tomas, et al. "Recurrent neural network based language model." Interspeech. Vol. 2. No. 3. 2010.
- [Sutskever+,NIPS2014] Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks." Advances in neural information processing systems 27 (2014).
- [Bahdanau+,2015] Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473 (2014).

# 參考資料 (1/2)

- [Vaswani+, NIPS2017] Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).
- [西田,2022] 西田京介. "自然言語処理とVision-and-Language." JSAI2022 チュートリアル講演資料 (2022).
- [Devlin+ (Google), NAACL19] Devlin, Jacob, Ming-Wei Chang, and Kenton Lee. "Google, KT, Language, AI: BERT: pre-training of deep bidirectional transformers for language understanding." Proceedings of NAACL-HLT. 2019.
- [Brown+ (OpenAI), NeurIPS2020] Brown, Tom, et al. "Language models are few-shot learners." Advances in neural information processing systems 33 (2020): 1877-1901.
- [Chowdhery+, 2022] Chowdhery, Aakanksha, et al. "Palm: Scaling language modeling with pathways." arXiv preprint arXiv:2204.02311 (2022).
- [Wei(Google)+, 2022] Wei, Jason, et al. "Chain of thought prompting elicits reasoning in large language models." arXiv preprint arXiv:2201.11903 (2022).
- [Kojima, 2023] Kojima, Takeshi, et al. "Large language models are zero-shot reasoners."
- [Touvron+ (Meta), 2023/2/27] Touvron, Hugo, et al. "Llama: Open and efficient foundation language models." arXiv preprint arXiv:2302.13971 (2023).
- [Wei+, 2022] Wei, Jason, et al. "Finetuned language models are zero-shot learners." arXiv preprint arXiv:2109.01652 (2021).
- [Taori+ (Stanford Univ.), 2023/3/13] Taori, Rohan, et al. "Alpaca: A strong, replicable instruction-following model." Stanford Center for Research on Foundation Models. <https://crfm.stanford.edu/2023/03/13/alpaca.html> 3.6 (2023): 7.
- [The Vicuna Team, 2023/3/31] Chiang, Wei-Lin, et al. "Vicuna: An open-source chatbot impressing gpt-4 with 90%\* chatgpt quality." See <https://vicuna.lmsys.org> (accessed 14 April 2023) (2023).
- [Yao+'23, ICLR] Yao, Shunyu, et al. "React: Synergizing reasoning and acting in language models." arXiv preprint arXiv:2210.03629 (2022).