

大規模言語モデルの回顧

GPT-1からChatGPTまでの発展

丁 世堯

京都大学 情報学研究科 伊藤研

2025年3月28日

大規模言語モデル (LLM: Large language models)

MO

ChatGPTができることを教えてください



はい、私(ChatGPT)ができることは以下の通りです。



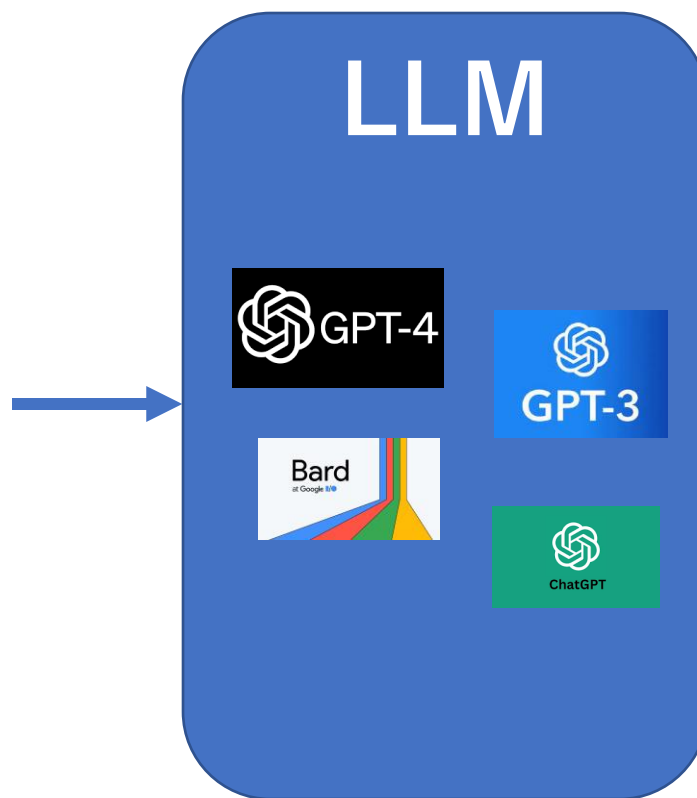
1. 質問に答える：私はあらゆるトピックについての質問に答えることができます。例えば、科学、歴史、文化、芸術、技術、スポーツ、料理など、幅広い分野に関する質問に対応できます。
2. 文章生成：私は文章を自動生成することができます。例えば、記事、小説、詩、レポート、ビジネス文書などを生成することができます。
3. 対話：私は自然な言語での対話を行うことができます。ユーザーが言葉を入力すると、それに応じた回答を生成することができます。
4. 翻訳：私は様々な言語を相互に翻訳することができます。
5. 推論：私は与えられた情報から論理的な推論を導き出すことができます。
6. 検索：私はインターネット上の情報を検索して、回答を提供することができます。
7. 感情認識：私は文章や発言の感情を認識することができます。
8. テキスト要約：私は与えられた長い文章を要約することができます。

以上が私(ChatGPT)ができる主なことです。ただし、私にはまだ限界があります。完璧な回答を提供できないこともあります。

↺ Regenerate response ↻

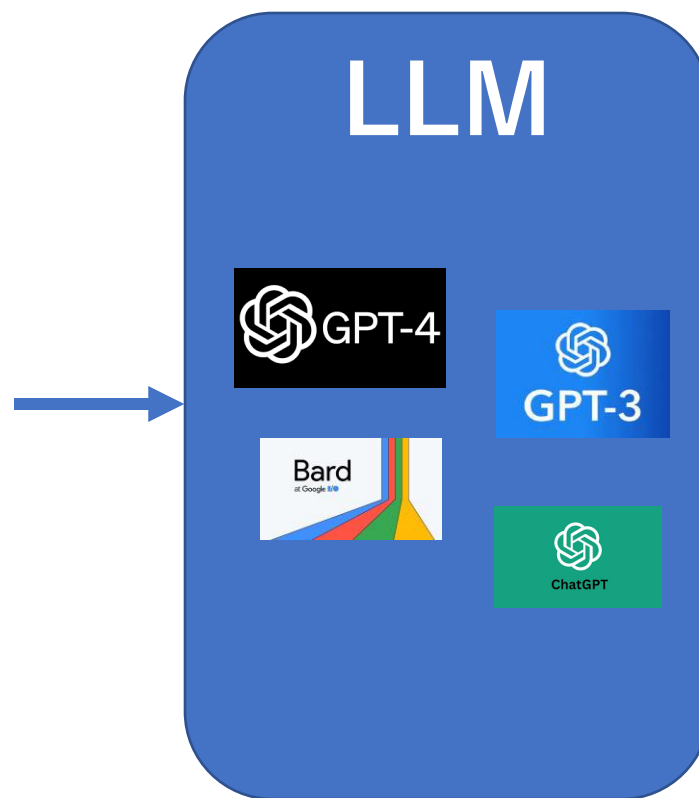
大規模言語モデル (LLM: Large language models)

今朝、スタバで



大規模言語モデル (LLM: Large language models)

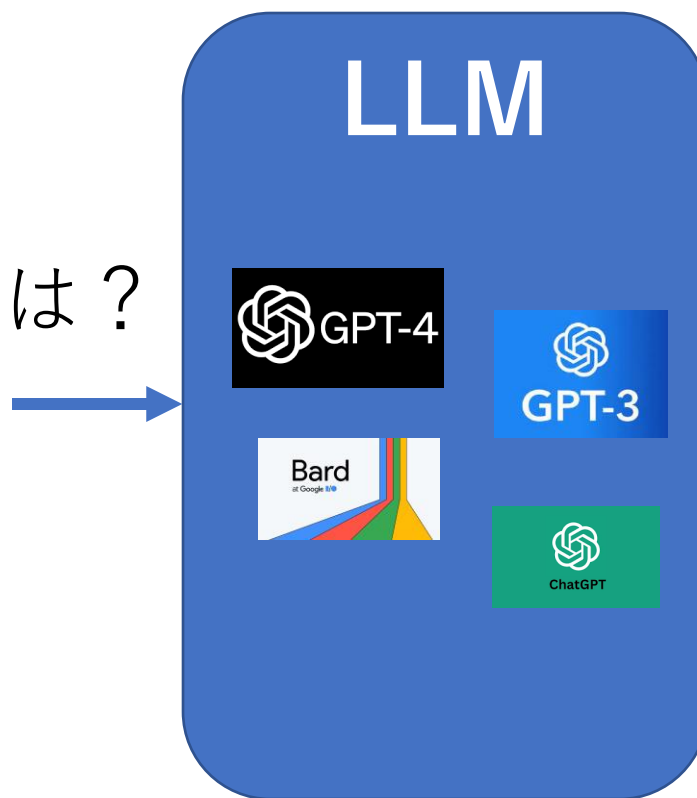
今朝、スタバで



今朝、スタバで
コーヒーを飲みました

大規模言語モデル (LLM: Large language models)

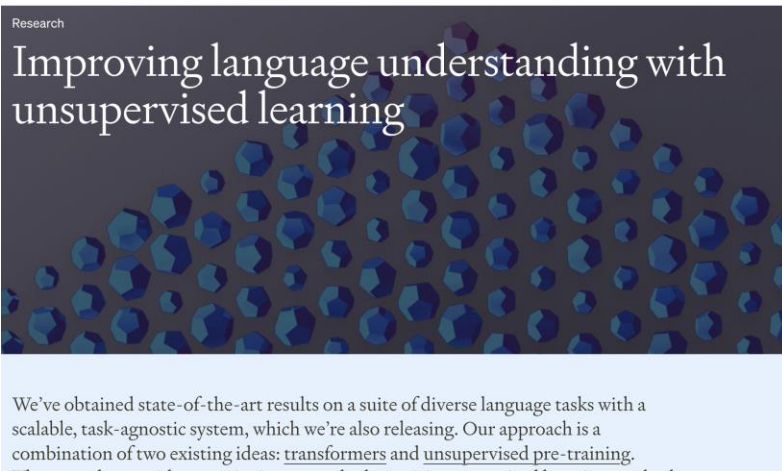
岡崎出身の戦国武将は？



岡崎出身の戦国武将は？
徳川家康です。

GPT-1 (2018)

OpenAI



Improving Language Understanding by Generative Pre-Training

Alec Radford
OpenAI
alec@openai.com

Karthik Narasimhan
OpenAI
karthikn@openai.com

Tim Salimans
OpenAI
tim@openai.com

Ilya Sutskever
OpenAI
ilyasu@openai.com

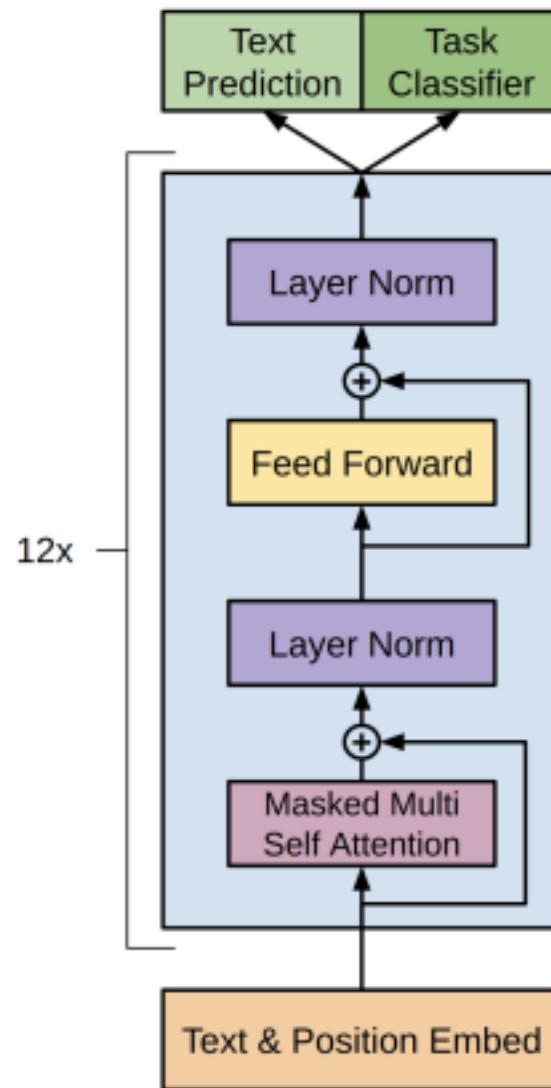
<https://onl.bz/8Q8TtSw>

1. 「**汎用**」とは何ですか?学習の際、汎用性が高く、移行可能なテキスト特徴表現を得るために、どのような目的関数が効果的ですか?
2. 汎用の特徴表現を得た後、それを異なる**下流タスク**にどのように移行させますか?

→GPT-1は、**事前学習**と**微調整**を行うことによってこれらの問題を解決しました。

GPT-1 : モデル構造

- 事前学習フェーズにおいて、GPTはモデルの主要モジュールとしてTransformerのデコーダ部分を採用しました。
- Transformerは2017年にGoogleが提案した特徴抽出モデルです。
- GPTは複数層のTransformerを積み重ねることで、プリトレーニングモデルを構築しました。



GPT-1：目標関数

あるテキストがあり、各単語を w としたとき、GPTは標準の言語モデルの目的関数を使用して、以下の尤度関数を最大化します：

$$L_1(U) = \sum_i \log P(u_i | u_{i-k}, \dots, u_{i-1}; \Theta)$$

具体的には、各単語 u_i の確率を予測します。

この確率は、それに先立つ u_{i-k} から u_{i-1} までの単語、およびモデルのパラメータ Θ に基づいています。

微調整段階では、特定の下流タスクのラベルが与えられた状況で、入力シーケンス x_1 から x_m までを与え、 y の確率を予測します。

$$P(y | x^1, \dots, x^m) = \text{softmax}(h_l^m W_y)$$

つまり、シーケンスを事前に訓練されたモデルに入力し、最後の層のTransformerの最後のトークン x_m の特徴 h_m を取得し、それを予測層に通すことで、対応するラベルの確率分布を得ることができます：

$$L_3(U) = L_2(U) + \lambda * L_1(U)$$

GPT-1：学習用のデータセット

モデルトレーニング

- トレーニングデータとして、初代GPTはBooksCorpusデータセットを使用しました。このデータセットは、約7000冊（約5GB）の独立した、様々なスタイルの書籍で構成される
- このデータセットを選ぶことによる主要な利点は、書籍テキストが多くの高品質な長文を含んでおり、モデルが長距離の情報依存を学ぶことを保証する点にあります。

モデルのいくつかの重要なパラメータは以下の通りです：

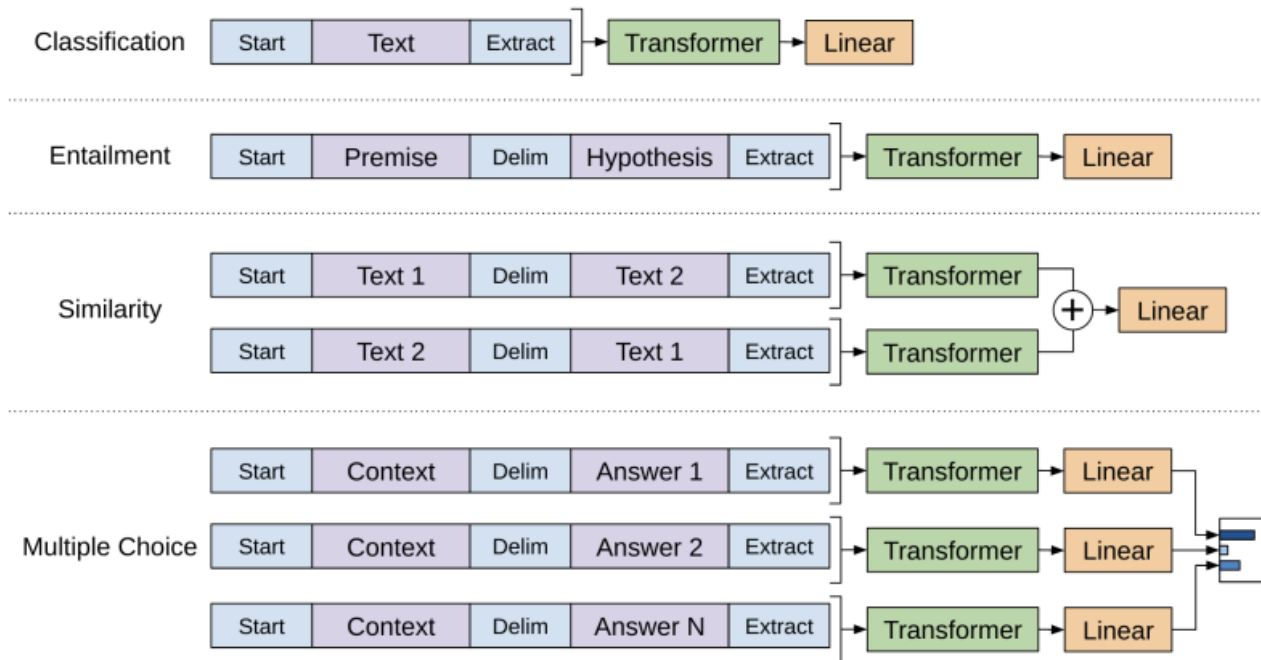
パラメーター	値
Transformer層の数	12
特徴の次元数	768
Transformerヘッドの数	12
総パラメーター数	1億1700万

GPT-1 : 微調整

NLP(自然言語処理)では一般的な4つのタスク (テキスト分類、テキスト含意、テキスト類似性、質問応答タスク) があります。

これらのタスクでは、序列の前後に「Start」と「Extract」という特別な識別子を追加して開始と終了を表し、序列の間には分離を示すために必要な「Delim」という識別子を追加します。

下流タスクの入力序列がどのように変わろうと、最終的な予測層がどのように変わろうと、中間の特徴抽出モジュールは変わらず、非常に優れた移行能力を持っています。



GPT-1：実験結果

これはNLPタスクにおいて、プリトレーニングとファインチューニングの**パラダイム**を使用することを提案した初期の作業の一つです。

GPTの実験は、モデルの精度と一般化能力が**デコーダー層の数が増えるにつれて向上し続け**、現在もまだ改善の余地があることを証明しています(下左図)

プリトレーニングされたモデルは**ゼロショット学習**の能力を持ち、プリトレーニングが進むにつれてその能力が強化されます(下右図)



OpenAIは**ゼロショット学習**の能力をさらに強化するために、GPT-1を発表してから1年後にGPT-2をリリースしました。

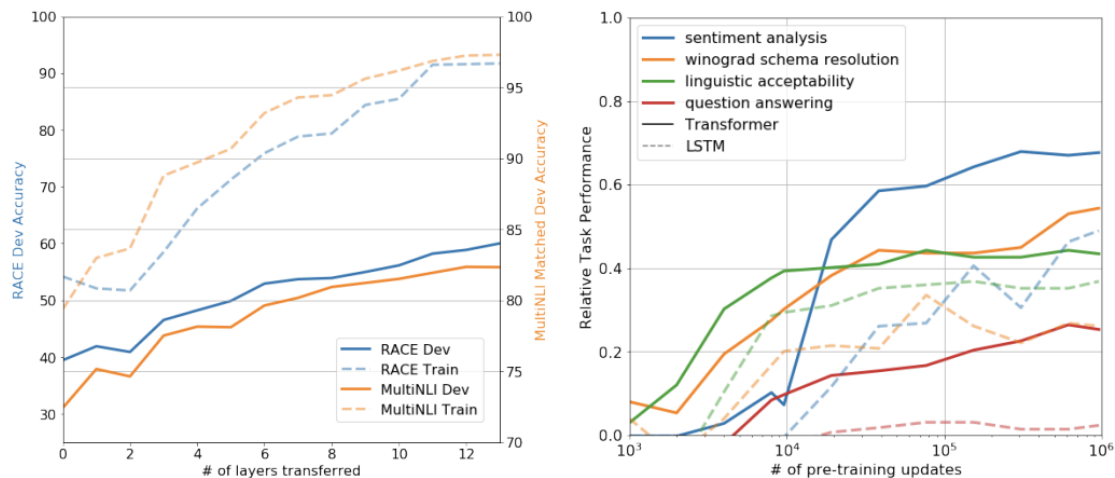


Figure 2: **(left)** Effect of transferring increasing number of layers from the pre-trained language model on RACE and MultiNLI. **(right)** Plot showing the evolution of zero-shot performance on different tasks as a function of LM pre-training updates. Performance per task is normalized between a random guess baseline and the current state-of-the-art with a single model.

GPT-2 (2019)

Language Models are Unsupervised Multitask Learners

Alec Radford ^{*1} Jeffrey Wu ^{*1} Rewon Child ¹ David Luan ¹ Dario Amodei ^{**1} Ilya Sutskever ^{**1}

<https://onl.bz/iYh8y6W>

- GPT-1でもBERTでも、NLPタスクで主流とされるプリトレーニングとファインチューニングにおいては、下流タスクの教師ありデータがある程度必要であり、モデルレベルでも予測を行うための追加モジュールが必要です。
- これには多くの手作業が伴います。GPT-2はこの問題を完全に解消しようとしており、ゼロショット学習によって、他のタスクへの移行時に追加のアノテーションデータやモデルトレーニングが不要になります。

GPT-2 : コアアイデア

- GPT-1では、下流タスクを行うために、異なるタスクの入力シーケンスを変更し、開始記号、区切り記号、終了記号などの特殊記号をシーケンスに追加しましたが、ゼロショットの条件下では、異なる下流タスクに応じてこれらの記号を追加することはできません。
- ゼロショットの条件下では、異なるタスクの入力シーケンスは、トレーニング時に見たテキストと同じような形式で表現されるべきです。つまり、自然言語の形で入力される必要があります。

例えば、以下のように変更されました：

機械翻訳タスク：フランス語に翻訳してください、{ 英語のテキスト }、{ フランス語のテキスト }

読解タスク：質問に答えてください、{ ドキュメント }、{ 質問 }、{ 答え }

GPT-2 : モデル構造

- GPT-2のモデルの構造はGPT-1と同じですが、いくつかの点が調節されました。
- これらの調整は、主にトレーニング時のテクニックとして扱われ、GPT-2の革新としては扱われません。
- 具体的には以下の点です：
 1. 後置層正規化（post-norm）を前置層正規化（pre-norm）に変更しました。
 2. モデルの最後の自己注意層の後に、追加の層正規化を行います。
 3. パラメータの初期化方法を調整し、残差層の数に応じてスケーリングし、スケーリング比は $1:\sqrt{N}$ になります。
 4. 入力シーケンスの最大長を512から1024に拡大しました。



GPT-2が上述のモデル調整を行った主な理由は、モデルの層数が増えるにつれて、勾配の消失と勾配の爆発のリスクが高まるのを防ぐためです。これらの調整により、事前トレーニングプロセス中に各層間の分散変化を減少させ、勾配をより安定させることができます。

GPT-2 : 学習用のデータセット

- トレーニングデータに関しては、ゼロショット学習の効果を保証するために、十分に大きくてカバー範囲が広い必要があります。
- そのため、GPT-2は多数のネットワークテキストデータを特別にクロールし、最終的にWebTextと呼ばれるデータセットを得ました。
- Reddit上の高品質なポストを選択し、最終的に4500万のウェブページリンクと、800万の有効なテキストドキュメントを取得し、コーパスのサイズは40GBになりました。

Parameters	Layers	d_{model}
117M	12	768 GPT-1
345M	24	1024 BERT-large
762M	36	1280
1542M	48	1600 GPT-2

Table 2. Architecture hyperparameters for the 4 model sizes.

GPT-2 : 実験結果

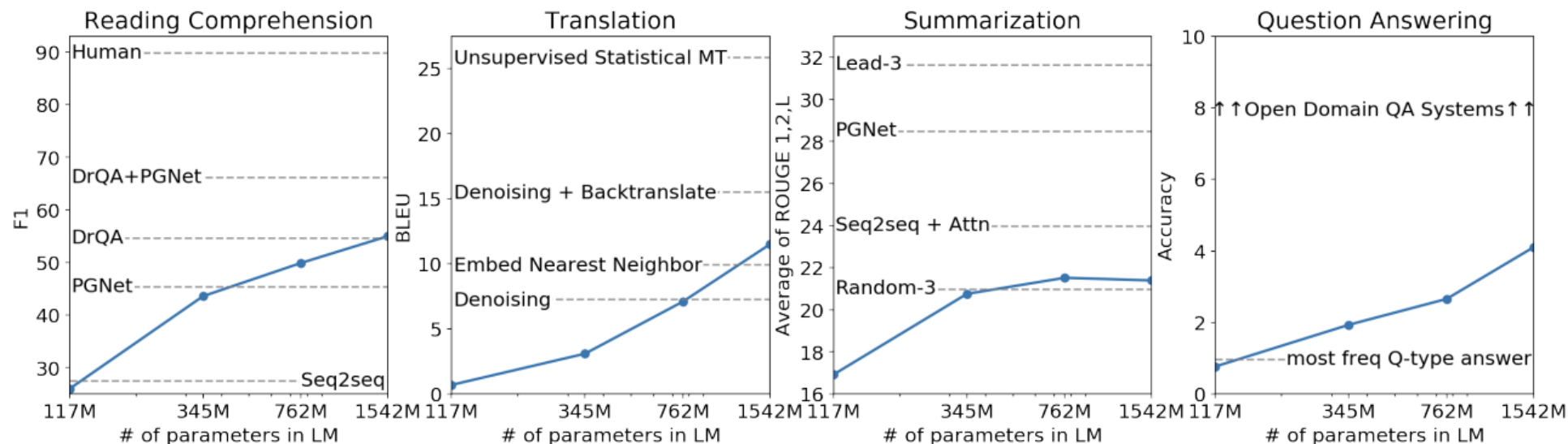


Figure 1. Zero-shot task performance of WebText LMs as a function of model size on many NLP tasks. Reading Comprehension results are on CoQA (Reddy et al., 2018), translation on WMT-14 Fr-En (Artetxe et al., 2017), summarization on CNN and Daily Mail (See et al., 2017), and Question Answering on Natural Questions (Kwiatkowski et al., 2019). Section 3 contains detailed descriptions of each result.

- 上記の結果から分かる通り、GPT-2は多くのタスクで教師なしアルゴリズムに比べて一定の向上を達成し、ゼロショットの能力を証明しました。
- しかし、多くのタスクで教師あり学習の微調整の方法と比較してまだいくつかの差があり、これが当時GPT-2の影響力がそれほど大きくなかった一因かもしれません。

GPT-2：まとめ

GPT-2と**GPT-1**には大まかに以下のような違いがあります：

- 1.学習方法：GPT-2は**ゼロショット**が、GPT-1では**事前学習と微調整**が主に行われました。
- 2.学習データのサイズ：GPT-2では800万文書**40GB**,GPT-1は**5GB**です。
- 3.モデルのサイズ:GPT-2は最大**15億**パラメーター、GPT-1は最大**1億**パラメーターです。
- 4.トレーニングパラメーター：**B**atchサイズが**64**から**512**に増加し、入力シーケンスのサイズが**512**から**1024**に増えました。
- 5.その他: モデル構造の調整、層正規化、およびパラメーター初期化方法が変わりました。

GPT-3 (2020)

- GPT-3では、サンプルが一切不要で非常に優れたパフォーマンスを発揮するモデルを追求するのではなく、**人間の学習方法のように、ごく少数のサンプル**だけであるタスクをマスターすることを考えています。そのため、GPT-3のタイトルは「Language Models are Few-Shot Learners」となっています。
- ここでの「few-shot」とは、以前の方法のように**少量のサンプルを使用して下流タスクで微調整を行うのではなく**、GPT-3のようなパラメータの規模では、たとえパラメータの微調整であっても、そのコストが計り知れないほど高いということを意味しています。

Language Models are Few-Shot Learners

Tom B. Brown*	Benjamin Mann*	Nick Ryder*	Melanie Subbiah*	
Jared Kaplan†	Prafulla Dhariwal	Arvind Neelakantan	Pranav Shyam	Girish Sastry
Amanda Askell	Sandhini Agarwal	Ariel Herbert-Voss	Gretchen Krueger	Tom Henighan
Rewon Child	Aditya Ramesh	Daniel M. Ziegler	Jeffrey Wu	Clemens Winter
Christopher Hesse	Mark Chen	Eric Sigler	Mateusz Litwin	Scott Gray
Benjamin Chess	Jack Clark	Christopher Berner		
Sam McCandlish	Alec Radford	Ilya Sutskever	Dario Amodei	
OpenAI				

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

GPT-3：モデル構造

GPT-3のモデル構造

- GPT-3では、GPTモデル構造を継続して使用していますが、従来のセルフアテンション（密集アテンションと呼ばれる）とは異なって、Sparse Transformerで提案されたsparse attentionモジュール（スパースアテンション）を導入しています。
- 密集アテンション：各トークン間でペアワイズにアテンションを計算し、計算量は $O(n^2)$ です。
- スパースアテンション：各トークンが他のトークンのサブセットとのみアテンションを計算し、計算量は $O(n * \log n)$ です。具体的には、スパースアテンションでは、相対距離がk以下、またはk、2k、3k...のトークンとのみアテンションを計算し、その他の全てのトークンに対するアテンションは0に設定されています。次のスライドに示されています。

GPT-3 : モデルサイズ

Model Name	n_{params}	n_{layers}	d_{model}	n_{heads}	d_{head}	Batch Size	Learning Rate
GPT-3 Small	125M	12	768	12	64	0.5M	6.0×10^{-4}
GPT-3 Medium	350M	24	1024	16	64	0.5M	3.0×10^{-4}
GPT-3 Large	760M	24	1536	16	96	0.5M	2.5×10^{-4}
GPT-3 XL	1.3B	24	2048	24	128	1M	2.0×10^{-4}
GPT-3 2.7B	2.7B	32	2560	32	80	1M	1.6×10^{-4}
GPT-3 6.7B	6.7B	32	4096	32	128	2M	1.2×10^{-4}
GPT-3 13B	13.0B	40	5140	40	128	2M	1.0×10^{-4}
GPT-3 175B or “GPT-3”	175.0B	96	12288	96	128	3.2M	0.6×10^{-4}

Table 2.1: Sizes, architectures, and learning hyper-parameters (batch size in tokens and learning rate) of the models which we trained. All models were trained for a total of 300 billion tokens.

GPT-3：実験設定

GPT-3は下流タスクの評価と予測に際して、三つの異なるアプローチを提供しています：

- **ゼロショット**：現在のタスクの自然言語説明のみを使用し、勾配更新は行いません。
- **ワンショット**：現在のタスクの自然言語説明に加えて、単一の簡単な入力出力例を使用し、勾配更新を行いません。
- **フューショット**：現在のタスクの自然言語説明に加えて、複数の簡単な入力出力例を使用し、勾配更新を行いません。

フューショットはin-context learningとも呼ばれています。ファインチューニングと同様に教師ありのデータが必要ですが、両者は以下のように違います。：

1. **【本質的な違い】** ファインチューニングは教師ありデータに基づいてモデルパラメータを更新しますが、フューショット（in-context learning）では、教師ありデータを使用しても勾配バック・プロパゲーションを行わず、**モデルパラメータは更新されません**。
2. フューショット（in-context learning）に必要な**データ量（10～100）**は、一般的にファインチューニングで使用するデータ量よりもはるかに少ないです。

GPT-3：実験結果

最終的に、多くの下流タスクでの実験を通じて、**フューショットの効果**が最も良く、ワンショットの効果がそれに続き、ゼロショットの効果が最も低いことが確認されました。

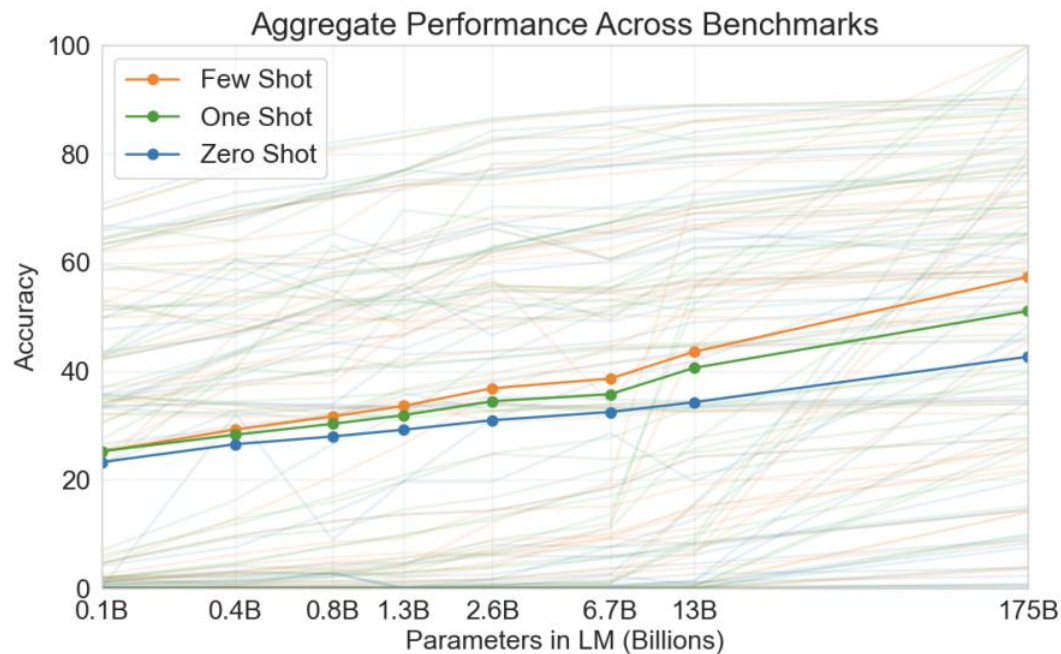


Figure 1.3: Aggregate performance for all 42 accuracy-denominated benchmarks While zero-shot performance improves steadily with model size, few-shot performance increases more rapidly, demonstrating that larger models are more proficient at in-context learning. See Figure 3.8 for a more detailed analysis on SuperGLUE, a standard NLP benchmark suite.

GPT-3 : まとめ

GPT-3と**GPT-2**には大まかに以下のような違いがあります：

- 1.モデルの性能:GPT-3はGPT-2を大きく上回り、人間が区別がつかないようなニュース記事を生成するレベルの能力があります。
- 2.学習方法：GPT-2はZero-shotが採用されていますが、GPT-3は主にFew-shotという非常に革新的な学習方法を採用しています。
- 3.モデル構造:GPT-3はsparse attentionモジュールを使用しています。
- 4.訓練データ：GPT-3は膨大なトレーニングコーパスを使用し、45TB（クリーニング後570GB）で、GPT-2の40GBよりもはるかに大きいです。
- 5.モデルのパラメーター：GPT-3は 膨大なモデルパラメータを有し、最大モデルでは1750億パラメータで、GPT-2の最大は15億パラメータです。

Instruct-GPT (2022)

- GPT-3は、多くのNLPタスクやテキスト生成能力で驚異的な成果を見せていますが、**偏見を持ったり、真実でないり、有害**で社会に悪影響を与える情報を生成することがまだあります。
- このような状況下で、OpenAI「**Alignment**」という概念を提案しました。
- これは、モデルの出力が**人間の真の意図と一致し、人間の好みに合致すること**を意味します。そのため、モデルの出力をユーザーの意図により「アライン」させるために、InstructGPTというプロジェクトが開始されました。

Training language models to follow instructions with human feedback

Long Ouyang* Jeff Wu* Xu Jiang* Diogo Almeida* Carroll L. Wainwright*

Pamela Mishkin* Chong Zhang Sandhini Agarwal Katarina Slama Alex Ray

John Schulman Jacob Hilton Fraser Kelton Luke Miller Maddie Simens

Amanda Askell† Peter Welinder Paul Christiano*†

Jan Leike*

Ryan Lowe*

OpenAI

<https://arxiv.org/pdf/2203.02155.pdf>

Instruct-GPT : 学習の流れ

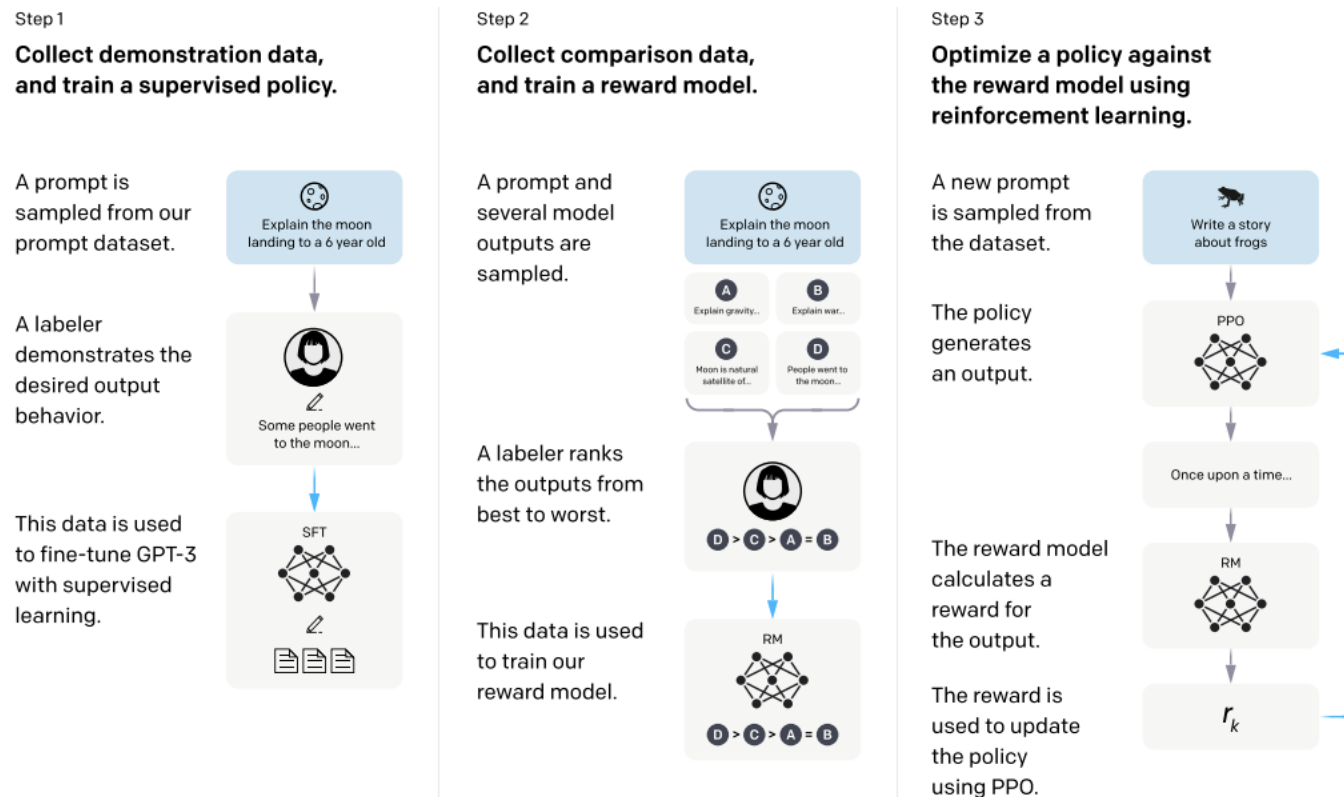


Figure 2: A diagram illustrating the three steps of our method: (1) supervised fine-tuning (SFT), (2) reward model (RM) training, and (3) reinforcement learning via proximal policy optimization (PPO) on this reward model. Blue arrows indicate that this data is used to train one of our models. In Step 2, boxes A-D are samples from our models that get ranked by labelers. See Section 3 for more details on our method.

TransformerによるLLM

