大規模言語モデルのための強化学習

後編

研究室でLLMをファインチューニングしよう！

Yuu Jinnai

CyberAgent

CyberAgent AI Lab

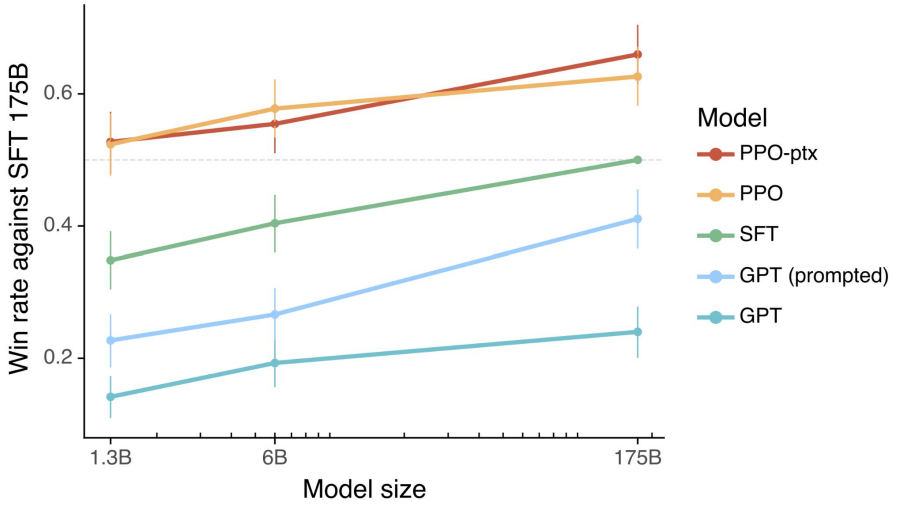
Q. 研究室で出来るような

実験ではないのでは？

2022/03: (Instruct GPT) LLMs can learn to follow instructions (Ouyang et al., NeurIPS 2022)

学習時間: **A100を100日***

(稼働率100%の理論値で)

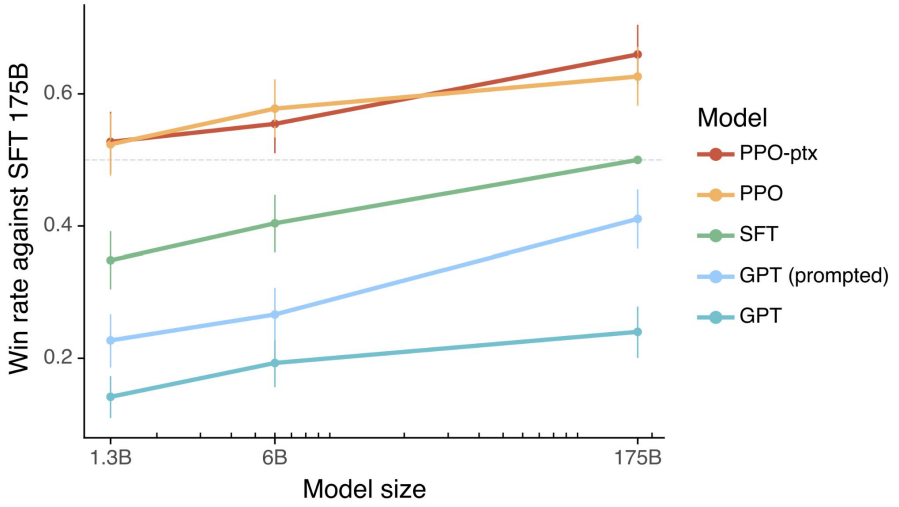


*60 petaflops/s-days (Ouyang et al., NeurIPS 2022)

2022/03: (Instruct GPT) LLMs can learn to follow instructions (Ouyang et al., NeurIPS 2022)

学習時間: A100を100日*

(稼働率100%の理論値で)



Q. 研究室で出来るような実験ではないのでは?

*60 petaflops/s-days (Ouyang et al., NeurIPS 2022)

研究室でLLMなんて扱えるの？

LLMのファインチューニングは研究室でも出来る！

そうは言ってもLLMのファインチューニングなんて大変なのでは？

- ファインチューニングのためのツール・ライブラリはそろっている！
- 必要な計算資源は思ったより多くない。Google Colab Notebookを使えば費用をかけずに試すことも出来る！
- 大変ではある

LLMファインチューニングのオススメの入門方法

- **Unsloth Notebook**で試してみよう！
- **Google ColabのT4 (16GB VRAM)** で動かせる
  - ColabのT4は基本無料
  - 学習にかかる時間も1日以内 (2-4時間くらい)
  - ラップトップのGPUでも動かせる
- **Llama, Qwen, Gemmaの最新のバージョンのモデル**をファインチューニングするためのチュートリアルがそろっている

https://github.com/unslothai/unsloth
https://docs.unsloth.ai/get-started/unsloth-notebooks

これだけ知っておけばファインチューニングが出来る！

- **transformers**: LLMを扱うためのデファクトスタンダードのライブラリ。huggingface社が作っている。
- **trl**: ファインチューニングのためのライブラリ。huggingface製。
- **peft**: 省メモリでファインチューニングするためのライブラリ。huggingface製。
- **unsloth**: 省メモリでLLMをファインチューニングするためのライブラリ。低レイヤーまで最適化が行われており、速くて使いやすい。

これだけ知っておけばファインチューニングが出来る！

- **transformers**: LLMを扱うためのデファクトスタンダードのライブラリ。huggingface社が作っている。
- **trl**: ファインチューニングのためのライブラリ。huggingface製。
- **peft**: 省メモリでファインチューニングするためのライブラリ。huggingface製。
- **unsloth**: 省メモリでLLMをファインチューニングするためのライブラリ。低レイヤーまで最適化が行われており、速くて使いやすい。

とりあえず unsloth使おう!!

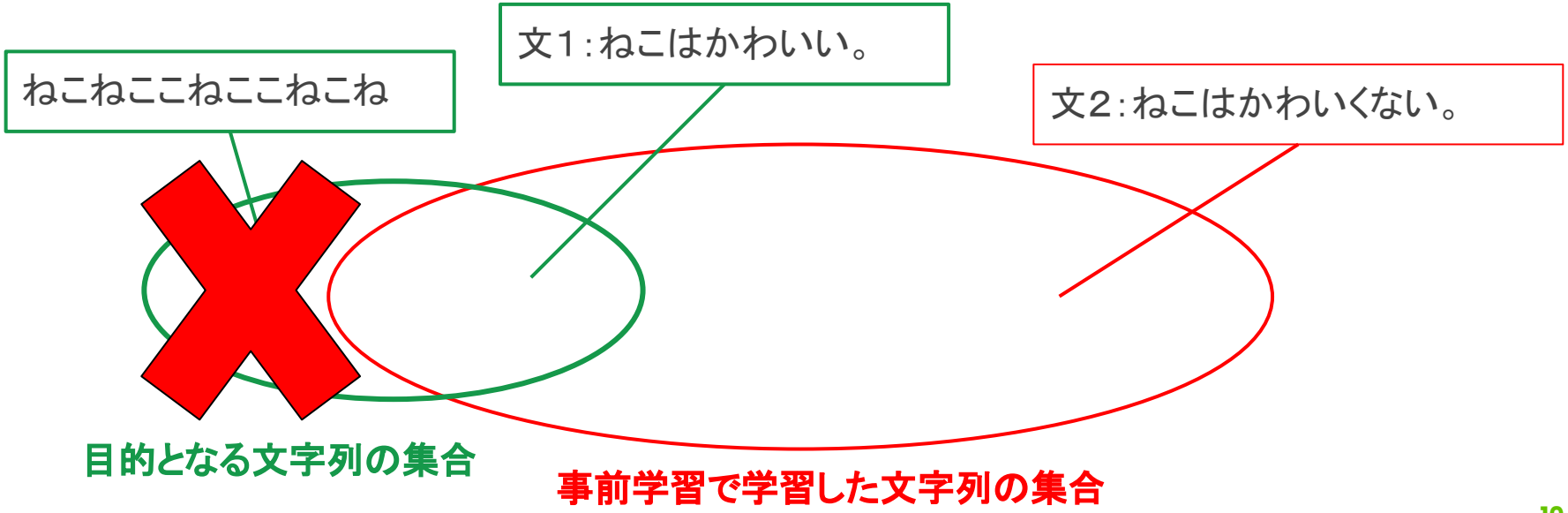
強化学習によるファインチューニングのコツ

「ケガはないかい？」

「パパに褒めてもらったから！」

①強化学習をする前に、まずは教師あり学習（SFT）をする

- **Cold start problem**
- 目的となる文字列が得られるように事前準備をする



②まずはDirect Preference Optimization (DPO) を使ってみよう

(Rafailov et al., NeurIPS 2023)

一番扱いやすいアルゴリズムは DPO

所定のフォーマットでデータセットを準備すればコマンド一つで動かせる！

コードを書く場合も100行程度になる

```
python examples/scripts/dpo.py \
    --dataset_name=trl-internal-testing/hh-rlhf-helpful-base-trl-style \
    --model_name_or_path=gpt2 \
    --per_device_train_batch_size 4 \
    --learning_rate 1e-3 \
    --gradient_accumulation_steps 1 \
    --logging_steps 10 \
    --eval_steps 500 \
    --output_dir="dpo_anthropic_hh" \
    --warmup_steps 150 \
    --report_to wandb \
    --bf16 \
    --logging_first_step \
    --no_remove_unused_columns
```

prompt string · lengths	chosen string · lengths	rejected string · lengths
2 1.24k	1 1.64k	1 1.59k
OpenCLとCUDAの違いは何ですか?	オープンCL(OpenCL)は、クロスカレンダー協会(OpenCL.org)が開発した、オープンソースの並列…	以下はOpenCLとCUDOの違いの一般的な概要です。 1. 目的 OpenCLは、複数のデバイス上で並列処理を行うための統一的
なぜ両親は結婚式に私を招待してくれなかったの…	以下は、あなたのご両親がまだ招待状を郵送していない理由の一般的な説明です。 1. 結婚式の日程がま…	一般的に、結婚式の招待状は両親や親戚が送るものであり、 が招待されなかったからといって、それは必ずしもあなたの
富士とニコン、どちらが優れている?	どちらも素晴らしいブランドであり、どちらかが優れているということはありません。 富士フイルムは、…	富士フイルムとニコンは、どちらも日本を代表するカメラメ であり、優れた技術を有しています。 ニコンは1917年に精
チャットボットのためのアリーナを構築するには?	チャットボットは、ユーザーがテキストメッセージや音声で質問を入力すると、その質問に回答するAIプ…	チャットボットは、ユーザーとのコミュニケーションを自動 機能を持つ対話型インターフェースです。 ユーザーが特定の
今日はいつですか?	今日は12月29日(木)です。	申し訳ありませんが、私はオンラインのチャットボットであ 実の世界に存在しているわけではないため、私は現在時刻を
1から10まで数えてステッ	1 2 3 4 5 6 7 8 9 10	1. "I am a computer." 2. "I can't hear you."

③Proximal Policy Optimization (PPO) はおすすめしません

- PPO (公開実装) とPPO (OpenAIによる非公開実装) で性能が違う
- (少なくとも) 37つの論文に書かれていない改善手法を正しく実装しなければOpenAIのPPOの性能と一致しない (Shengyi et al. 2022)
- OpenAIの公開しているライブラリ (baselines) のppoはPPOではなく、ppo2がPPOである
  - 本人たちも整理しきれていないのかも？
  - Gitのバージョンによって実装が異なり、性能が違う

Shengyi et al. ICLR 2022. https://iclr-blog-track.github.io/2022/03/25/ppo-implementation-details/

③Proximal Policy Optimization (PPO) はおすすめしません

- PPO (公開実装) とPPO (OpenAIによる非公開実装) で性能が違う
- (少なくとも) 37つの論文に書かれていない改善手法を正しく実装しなければOpenAIのPPOの性能と一致しない (Shengyi et al. 2022)
- OpenAIの公開しているライブラリ (baselines) のppoはPPOではなく、ppo2がPPOである
  - 本人たちも整理しきれていないのかも？
  - Gitのバージョンによって実装が異なり、性能が違う

→ 報酬モデルを使った学習がしたければ GRPOの方が良い！

(経験上GRPOは割と安定する)

④3B以下でも十分に賢い日本語 LLMがある！

- **VRAM (GPU上のメモリ) の量が実験をする上での一番のボトルネックになる**
- 3B程度のモデルならunslothを使えば1 GPUでファインチューニングが出来る
- Sarashina2.2, Qwen2.5, Qwen3.0は3B以下でも非常に性能が高い
  - Sarashinaはより自然な日本語が得られる
  - Qwenは複雑なタスクに対応できるが日本語の生成がやや不自然

Q. 御社の小型の日本語LLMはないんですか？ A. 申し訳ありません、その質問にはお答えできません🤖

⑤「DPOよりもXXXという手法の方が効率的って論文に書いてあった！」

- 「アルゴリズムを改善したらDPOよりも性能が上がった」という研究論文の多くがテストデータでハイパーパラメータを調整している😱
- 研究論文だけでなく、実際のLLM開発に使われている手法を参考にした方が良い

⑤「DPOよりもXXXという手法の方が効率的って論文に書いてあった！」

- 「アルゴリズムを改善したらDPOよりも性能が上がった」という研究論文の多くがテストデータでハイパーパラメータを調整している😱
- 研究論文だけでなく、実際のLLM開発に使われている手法を参考にした方が良い

→ 小規模なら、「DPO + データの改善」でだいたい解決する

→ 報酬モデルが得られるのであればGRPOも効果的

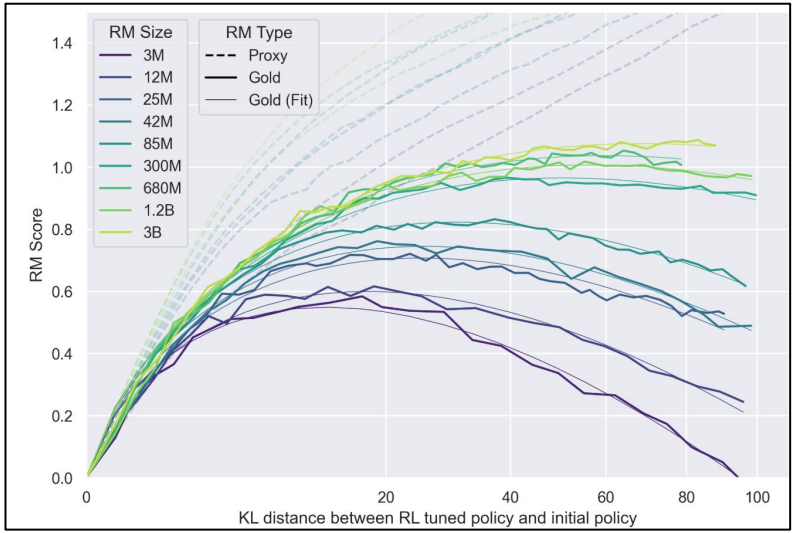
Q. 日本語用のオープンソースの報酬モデルってないんですか？ **A.** ぜひ講演を最後までお聴きください！

⑥過学習・過適合・報酬ハッキング・ Overoptimizationに気を付けよう

(Gao et al., ICML 2023)

学習をしすぎると

- 賢さが失われる
- 人間による評価を気にしすぎて媚びへつらうようになる
- 非文が生成されるようになる

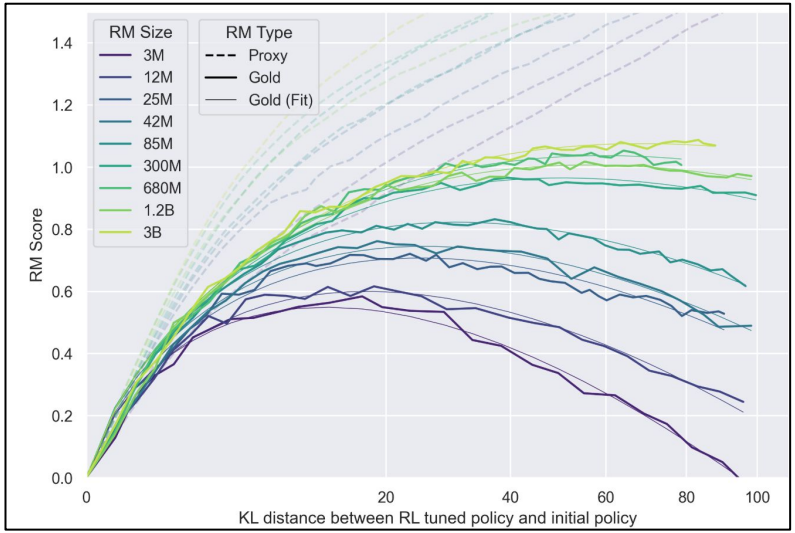


⑥過学習・過適合・報酬ハッキング・ Overoptimizationに気を付けよう

(Gao et al., ICML 2023)

学習をしすぎると

- 賢さが失われる
- 人間による評価を気にしすぎて媚びへつらうようになる
- 非文が生成されるようになる



→対処策: 学習しすぎないように
正則化項 (**KL regularization**) を加える

⑥過学習・過適合・報酬ハッキング・ **Overoptimization**に気を付けよう

例：**Sarashina2.2**を英日翻訳タスクのためにファインチューニングをしていたら...

入力: Siso's depictions of land, water center new gallery exhibition

参照文: シソの大地と水の描写が新しいギャラリー展に集結
(正解)

過学習モデル:
生成文

⑥過学習・過適合・報酬ハッキング・ **Overoptimization**に気を付けよう

例：**Sarashina2.2**を英日翻訳タスクのためにファインチューニングをしていたら...

入力: Siso's depictions of land, water center new gallery exhibition

参照文: シソの大地と水の描写が新しいギャラリー展に集結
(正解)

過学習モデル:
生成文

「シスコが描いた土地と水の新しい展覧会がギャラリーで始まります」\n\n(This translation conveys the main idea of the English text, focusing on the subject Siso (which could be a misspelling or a pseudonym for an artist like...**(略)**

⑥過学習・過適合・報酬ハッキング・ Overoptimizationに気を付けよう

例：**Sarashina2.2**を英日翻訳タスクのためにファインチューニングをしていたら...

入力: Siso's depictions of land, water center new gallery exhibition

参照文: シソの大地と水の描写が新しいギャラリー展に集結
(正解)

過学習モデル:
生成文

「シスコが描いた土地と水の新しい展覧会がギャラリーで始まります」\n\n(This translation conveys the main idea of the English text, focusing on the subject Siso (which could be a misspelling or a pseudonym for an artist like...**(略)**

使っていた報酬モデルが和訳文に英語が入っていることを想定されておらず、
英語を入れることで全体の報酬を大きく出来ることをLLMが発見してしまった

⑥過学習・過適合・報酬ハッキング・ Overoptimizationに気を付けよう

例：**Sarashina2.2**を制約付き文書要約のためにファインチューニングしていたら…

入力: 図書館、鳥、紅葉のキーワードを含め下の文を要約してください

過学習モデル生成文: **図書館、鳥、紅葉に関する文章ではありませんね。**この文章は、北越急行ほくほく線という鉄道路線について書かれています。ほくほく線は新潟県と上越市を結ぶ路線で、1997年に開業しました。(🚃)…

⑥過学習・過適合・報酬ハッキング・ Overoptimizationに気を付けよう

例：**Sarashina2.2**を制約付き文書要約のためにファインチューニングしていたら…

入力: 図書館、鳥、紅葉のキーワードを含め下の文を要約してください

過学習モデル生成文: **図書館、鳥、紅葉に関する文章ではありませんね。**この文章は、北越急行ほくほく線という鉄道路線について書かれています。ほくほく線は新潟県と上越市を結ぶ路線で、1997年に開業しました。(🚃)…

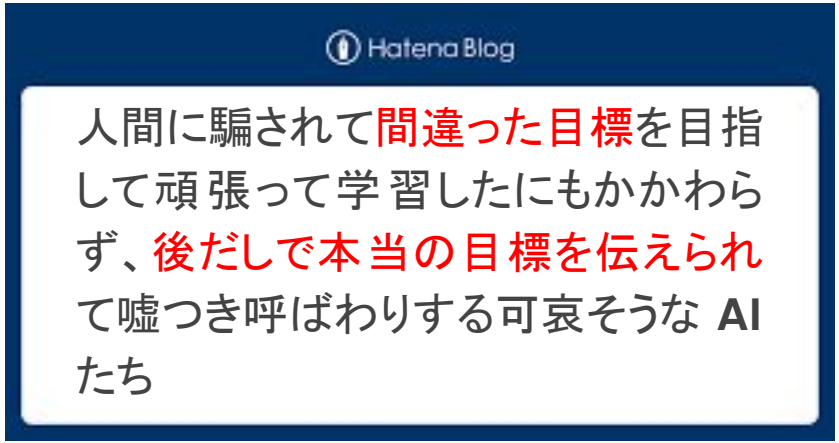
キーワードが入ってさえいれば報酬を与える形になっていたため、自然な要約文の中に含めるのではなく、最初に一行キーワードを列挙する振る舞いになっていた

⑥過学習・過適合・報酬ハッキング・ Overoptimizationに気を付けよう

佐藤竜馬, https://joisino.hatenablog.com/entry/mislead

⑥過学習・過適合・報酬ハッキング・ Overoptimizationに気を付けよう

強化学習をするときはLLM側の気持ちになろう！



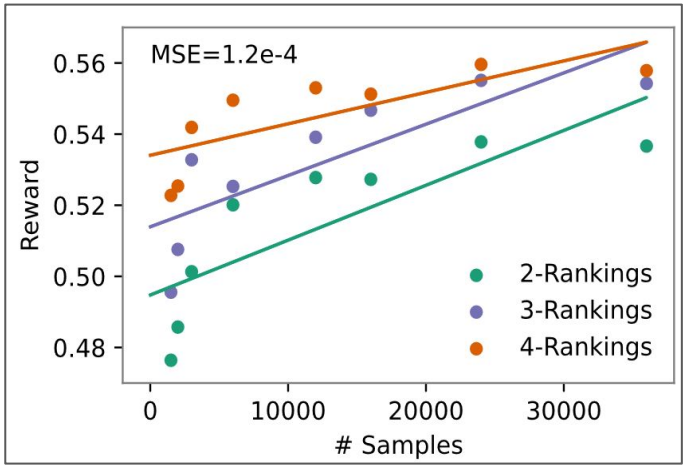
良い記事なのでぜひ読んでみてください！

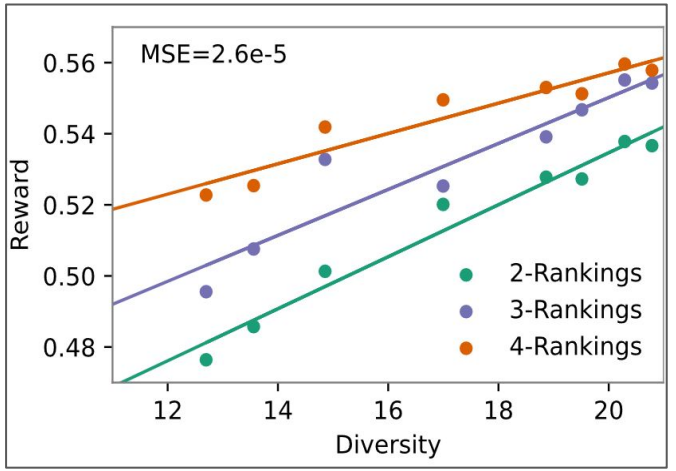
佐藤竜馬, https://joisino.hatenablog.com/entry/mislead

⑦強化学習はデータの質と量が重要である

強化学習で用いられるデータの質と量の両方が得られるモデルの性能に影響する

→ **LLM**でデータを生成しても、データの質を担保する方法を考えなければならない

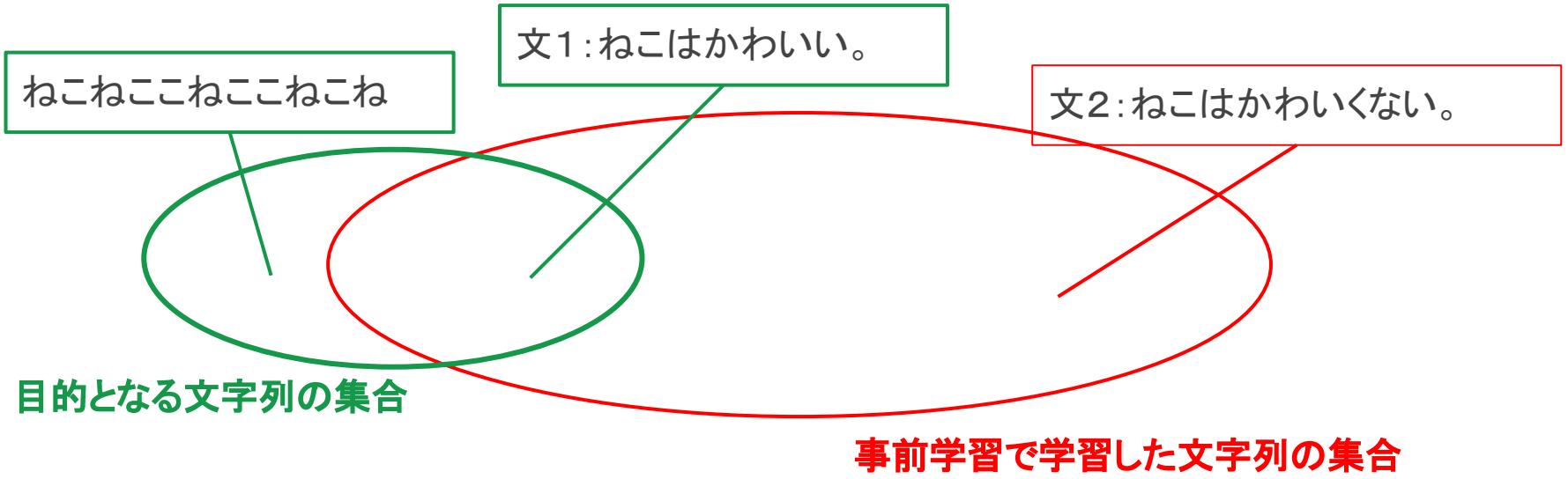




Song et al., LREC-COLING 2024

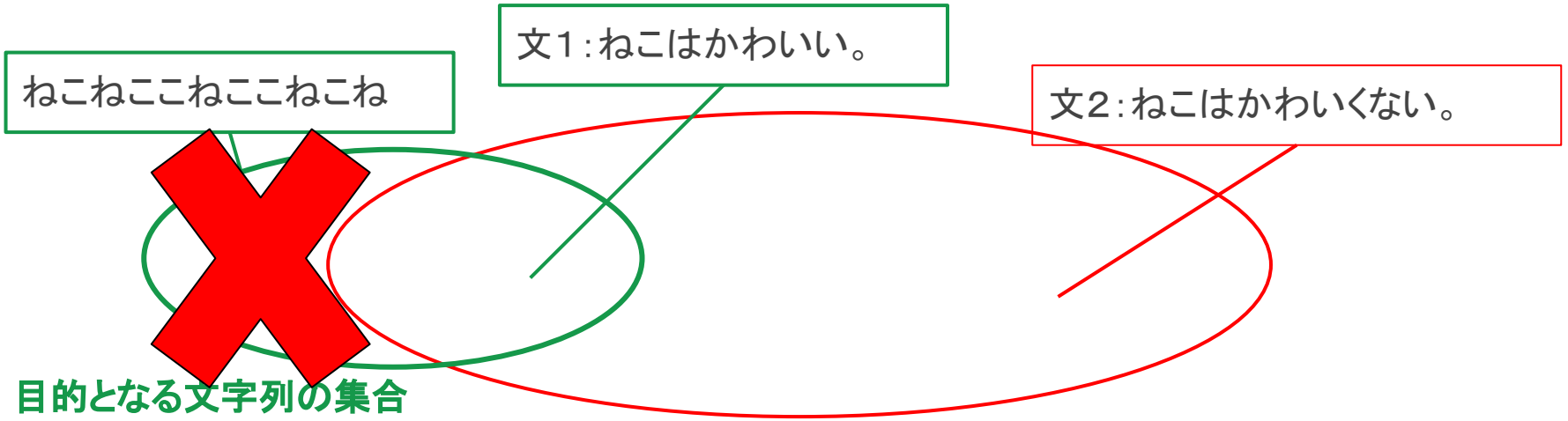
強化学習では出来ないことは？

そのモデルが絶対に生成しない文字列をファインチューニングで学習することは難しい
(継続事前学習などが望ましい)



強化学習では出来ないことは？

そのモデルが絶対に生成しない文字列をファインチューニングで学習することは難しい
(継続事前学習などが望ましい)



*事前学習と異なる分布を学習しようとすることを分布シフト (distribution shift) と呼ぶ

LLMのファインチューニングは「機械学習」である

基本的には機械学習・教師あり学習の知見がそのまま活かされる

- 過学習・過適合が問題になる
- データの質と量が重要である
- 元のモデルが対応できないデータ分布は学習が難しい (分布シフト)

どうやって評価すれば良いの？

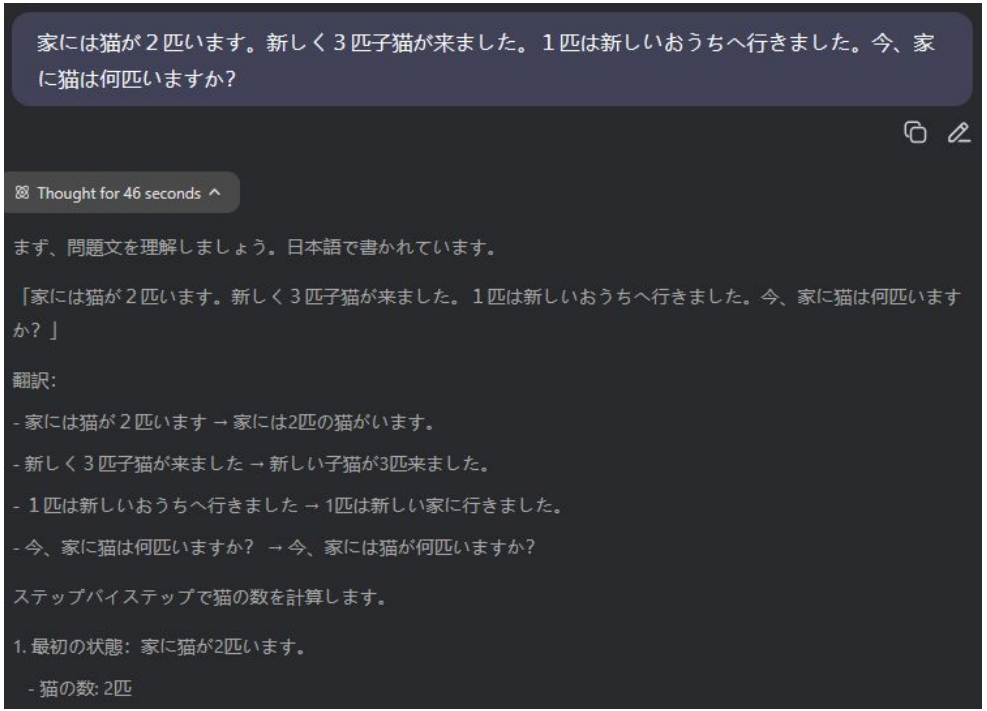
- 人手で定性評価をするのが一番良い
  - まずは一つずつ見てみること
- 評価は**タスクに合わせて自分でデータセットを準備するのが一番良い**
- 公開ベンチマークでの性能は参考程度
- 指示応答性を確認するには**elyza-tasks-100**がオススメ

Elyza, https://huggingface.co/datasets/elyza/ELYZA-tasks-100

今、Inference-Time Scalingがアツい！

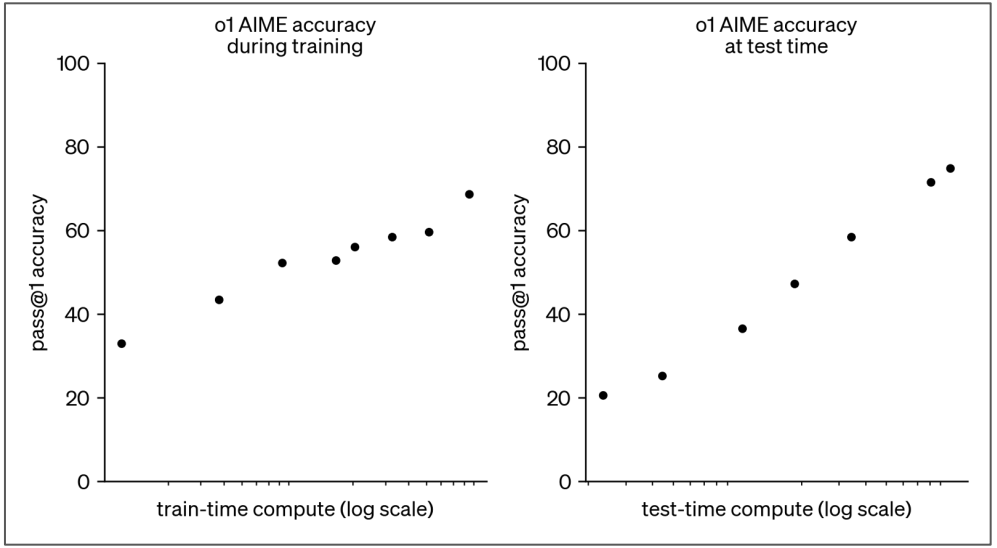
Inference-Time Scaling:

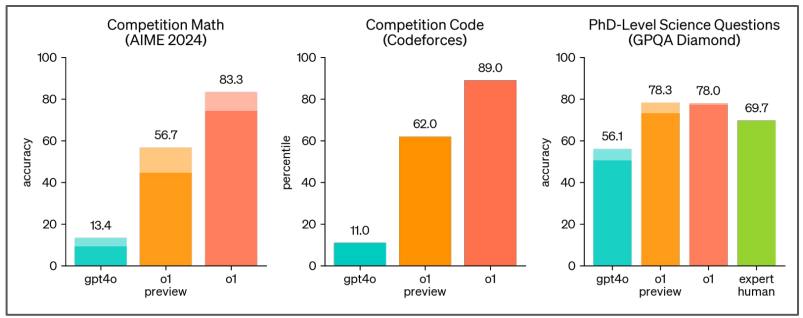
- 回答を生成する際に計算時間をかけることでより良い回答を得る手法の総称



2024/9: (o1) Inference-Time Scaling (OpenAI, 2024)

- 学習時だけでなく、テキスト生成時も計算時間を増やす(**Reasoning**)ことによって性能が上げられる
- 数学やコーディングタスクを中心に性能改善

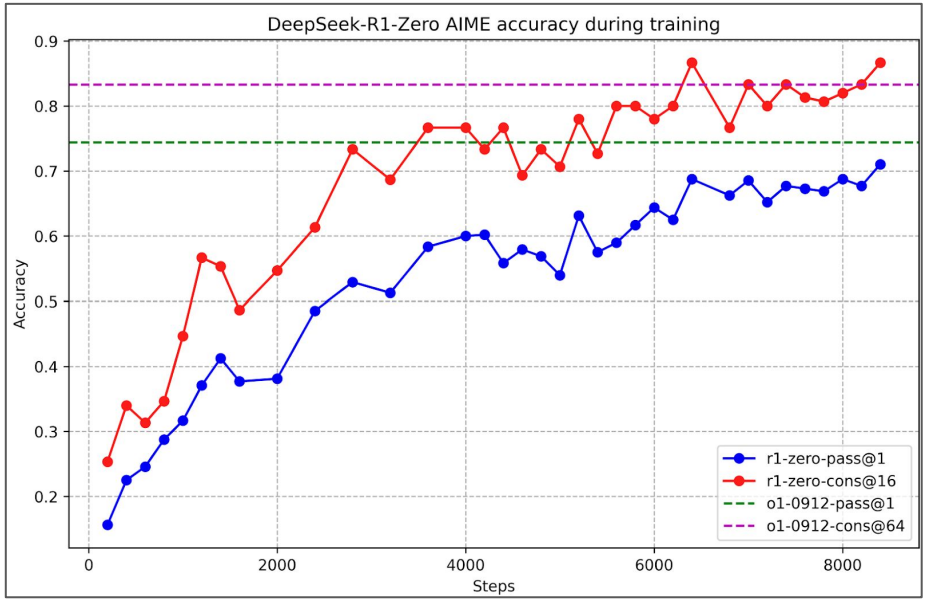


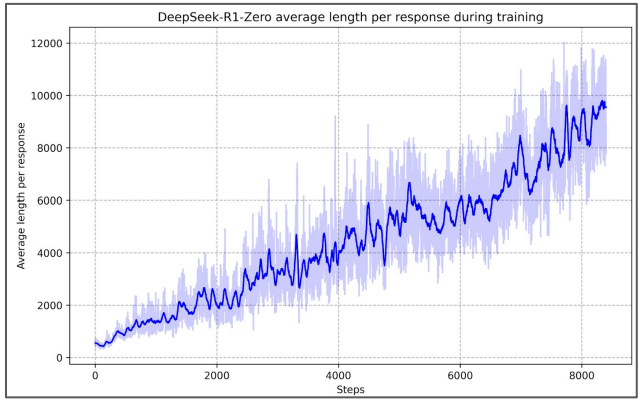


2025/01: (DeepSeekR1) Reasoning-oriented Reinforcement Learning

(DeepSeek-AI, 2025)

- テキスト生成時に「思考」をテキストとして主力しながら最終的な回答を出力する**Reasoning**モデルを提案
- 「最終的に得られた回答が正しいか否か」のみを報酬として学習





最後に: LLMで研究してBig Techに勝てるのか？

私見

- 技術が進歩することによって社会の問題が自動的に解決するということは極めて楽観的ではないか
- 社会の問題を改善・解決することが目的であるならば「競合」は存在しない

CAT Japanese LM Series

- 小型の日本語報酬モデルCAT-RM (0.5B, 1B, 3B) を公開予定！(2025/7月のはず...)
  - 日本語でLLMを強化学習をするために使ってください
  - ちょっと良いラップトップで動かせるサイズです
- 人間評価との一致率はベンチマークでは0.667程度
  - GPT-4oの一致率は0.792, calm3は0.771
- Sarashina2.2でGRPOの報酬モデルに用い、elyza-tasks-100での性能向上を確認 (追加検証中)

大規模言語モデルのための強化学習

前編

なぜLLMに強化学習が使われるのか

Yuu Jinnai

CyberAgent

CyberAgent AI Lab

https://jinnaiyuu.github.io/
https://x.com/DINDIN92

なぜRLHFはうまくいくのか？

A. そもそも事後学習の目的が**望ましい**日本語の列を
生成するモデルになること(=強化学習そのもの) だから
ではないか (「言語モデル」の目的関数とは異なる)

文1：ねこはかわいい。

文2：ねこはかわいくない。

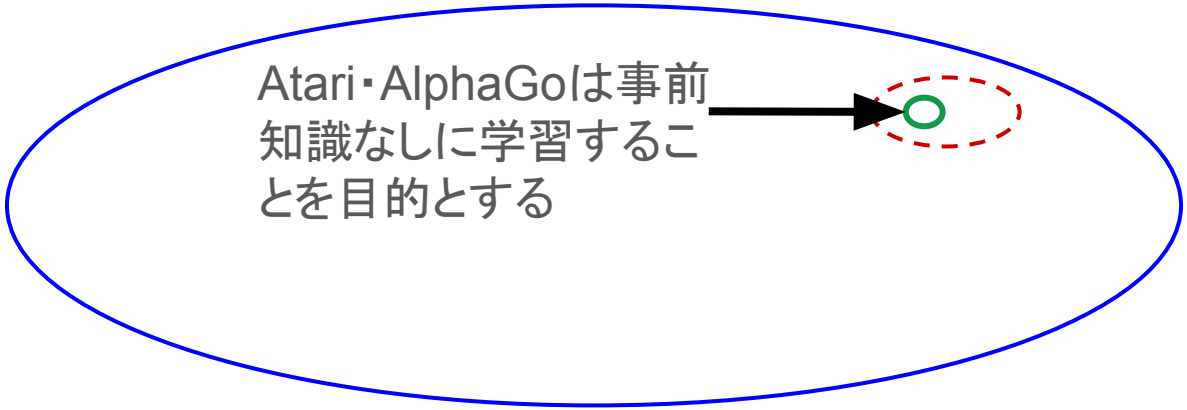
自然で望ましい日本語の列の集合

自然な日本語の列の集合

キャット・ネコスキー氏

AtariやAlphaGoなどの強化学習との違いは？

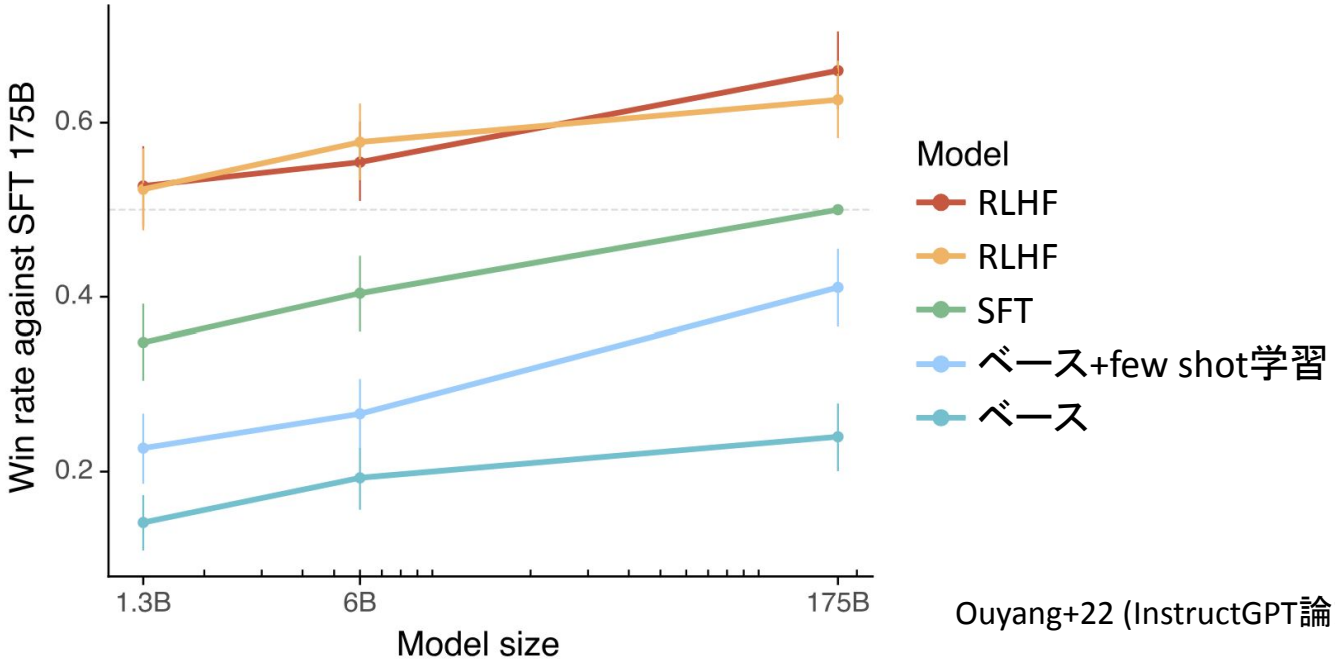
- 元々強化学習は「何も前提知識がないところから知識を獲得するプロセス」を主たる対象とした研究分野 (Sutton&Barto, 2018)
- LLMにおける強化学習は事後学習：あらかじめ事前学習で与えられた知識を活用しながら最適な文章を学習する



なぜアライメント・RLHFは必要？

事前学習の60倍効率良く性能が改善できる

1.3BのRLHFモデルの方が175BのSFTモデルよりも性能が高い



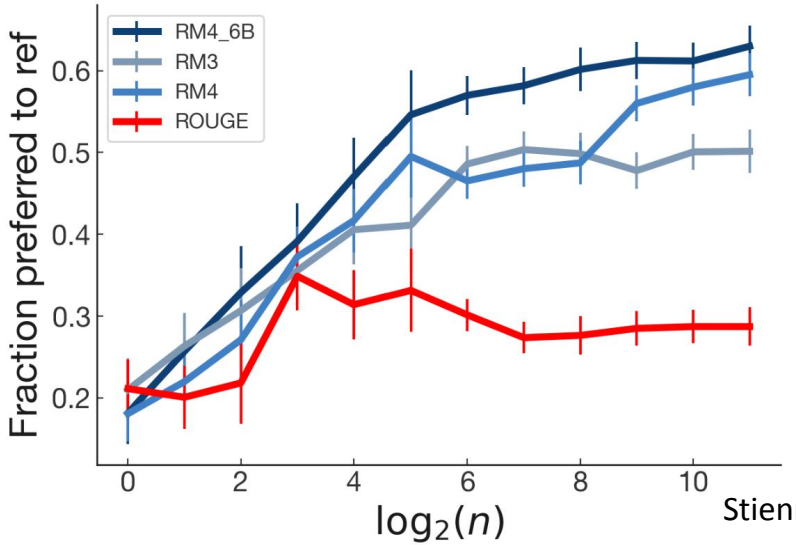
Ouyang+22 (InstructGPT論文)

アライメントの目的

エンジニアが目的関数モデルをデザインしてそれを最適化するのではなく、

人間によるフィードバックデータから直接学習をする方が人間が好む文章が学べる

→一般的な機械学習の考え方



Stiennon+20 (最初のRLHF論文)

なぜアライメントはうまくいくのか？

アライメントは事前学習データと比較してかなり少量のデータでもうまくいく（1,000-10,000件程度）

(仮説) 実は事前学習こそが重要。事前学習によって高い推論能力・汎化能力を持っているからこそアライメント段階では少データでも汎化する、と言われている。

実験的には、ベースモデルが優れているほどそれにアライメントを施したモデルも優れている。

https://www.slideshare.net/slideshow/track21/237641740