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

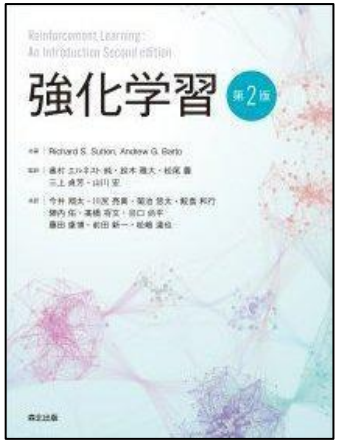
https://jinnaiyuu.github.io/
https://x.com/DINDIN92

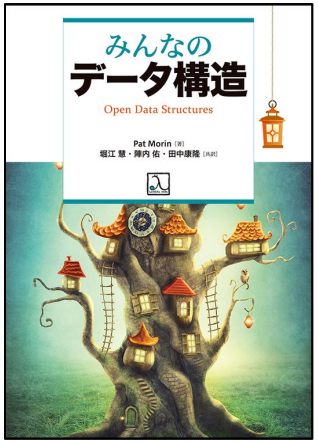
Yuu Jinnai 陣内 佑

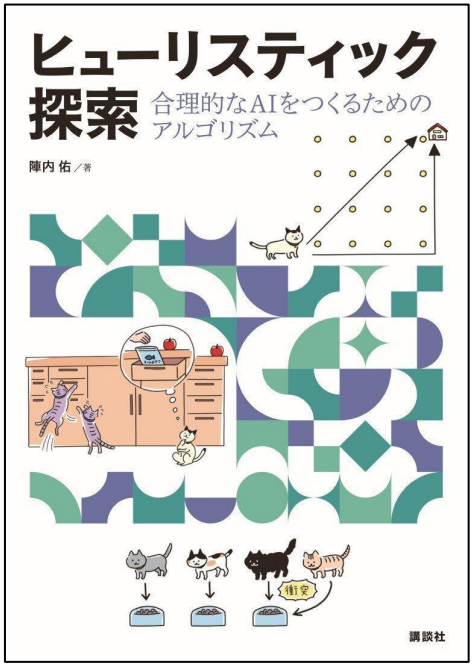
CyberAgent AI Lab, Reinforcement Learning Team

Research Interest

- Sequential Decision Making
  - Planning and Search (Text Generation)
  - Reinforcement Learning (RLHF)





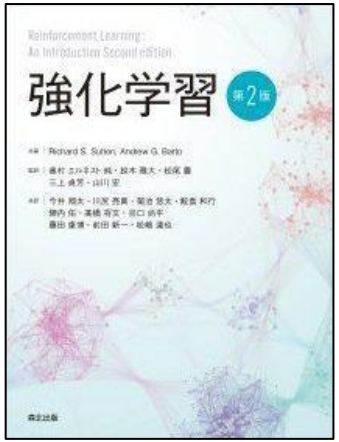


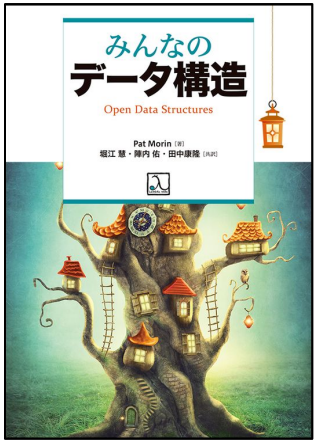
Yuu Jinnai 陣内 佑

CyberAgent AI Lab, Reinforcement Learning Team

Research Interest

- Sequential Decision Making
  - Planning and Search (Text Generation)
  - Reinforcement Learning (RLHF)



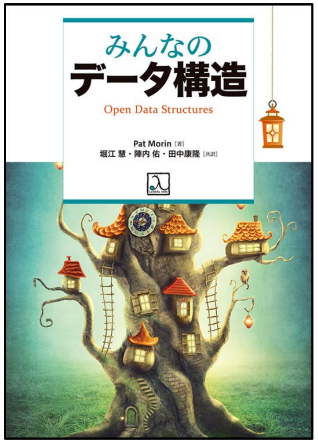


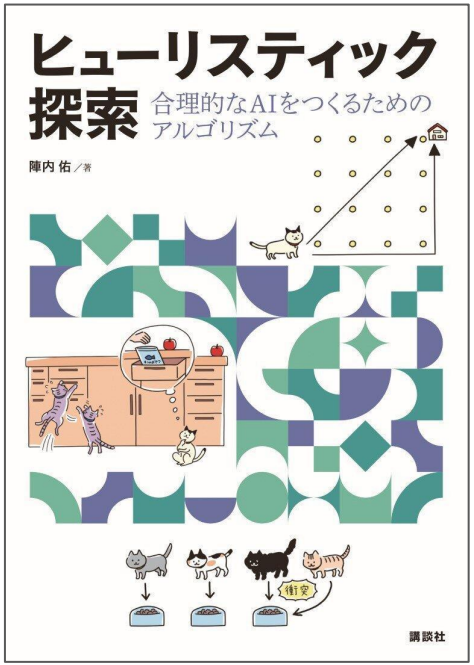
Yuu Jinnai 陣内 佑

CyberAgent AI Lab, Reinforcement Learning Team

Research Interest

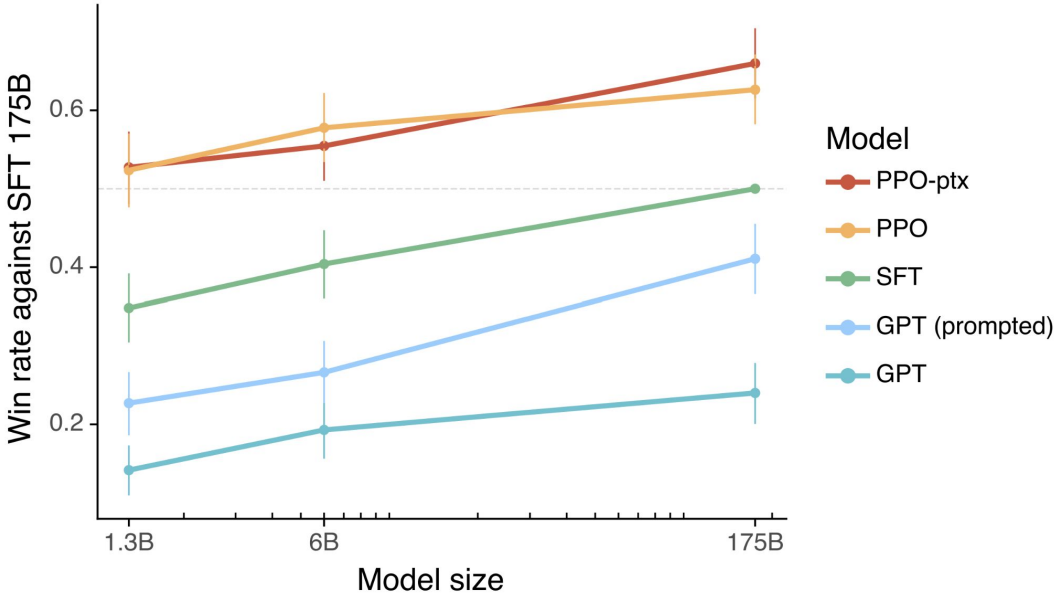
- Sequential Decision Making
  - Planning and Search (Text Generation)
  - Reinforcement Learning (RLHF)





なぜLLMに強化学習が使われるのか？

A. 他の手法よりも圧倒的に効率が良いから



(Ouyang et al., NeurIPS 2020)

はじめに: 言語モデルとは何か？

“Can’t be a bath!”

言語モデル (Language Model) とは？ (Jurafsky & Martin, 2000)

定義：言語モデル

言語モデルとは、
文字列の集合上の確率分布である *

例：P(ねこはかわいい。) = 0.5

*厳密には文字ではなくトークンと呼ばれる、言語モデルにとっての「単語」のようなものの列

言語モデルは、文字列に対してその確率を与える（Jurafsky & Martin, 2000）

文1：ねこはかわいい。

言語モデルは、文字列に対してその確率を与える（Jurafsky & Martin, 2000）

文1：ねこはかわいい。

文章はひらがなと文末を表す”。”の列から構成されるとすると、*

$$文1 = [ね, こ, は, か, わ, い, い, 。]$$

*厳密には列の終端を表す特殊なトークン（end-of-string）が使われる

言語モデルは、文字列に対してその確率を与える（Jurafsky & Martin, 2000）

文1：ねこはかわいい。

文章はひらがなと文末を表す”。”の列から構成されるとすると、*

$$文1 = [ね, こ, は, か, わ, い, い, 。]$$

言語モデルPは、文章の集合の中で文1が出現する確率を表す

$$P([ね, こ, は, か, わ, い, い, 。]) = 0.5$$

***厳密には列の終端を表す特殊なトークン（end-of-string) が使われる**

言語モデルは、文字列に対してその確率を与える（Jurafsky & Martin, 2000）

文1：ねこはかわいい。

文章はひらがなと文末を表す"。"の列から構成されるとすると、*

$$文1 = [ね, こ, は, か, わ, い, い, 。]$$

言語モデルPは、文章の集合の中で文1が出現する確率を表す

$$P([ね, こ, は, か, わ, い, い, 。]) = 0.5$$

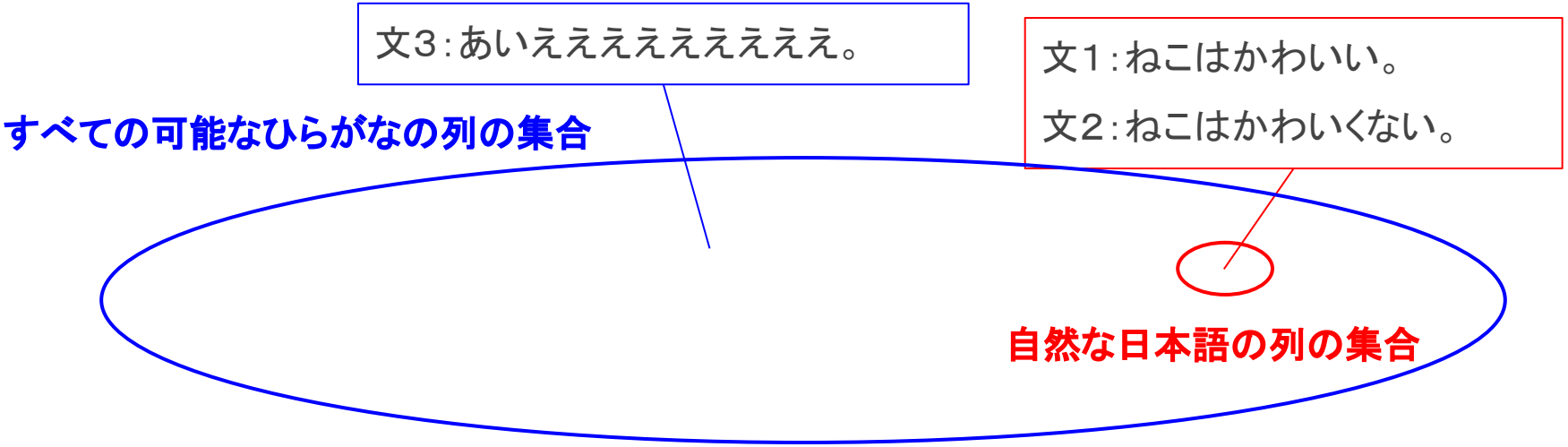
→言語モデルPによると、人間は
0.5の確率で「ねこはかわいい。」と発話する

*厳密には列の終端を表す特殊なトークン（end-of-string）が使われる

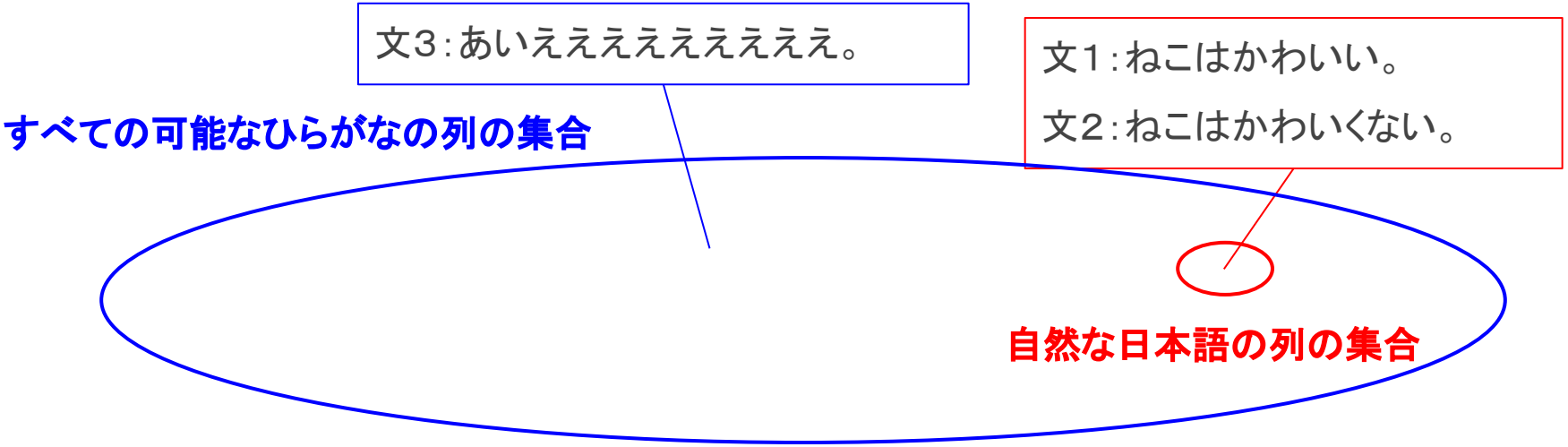
言語モデルは、文字列に対してその確率を与える（Jurafsky & Martin, 2000）

文1：ねこはかわいい。	= [ね, こ, は, か, わ, い, い, 。]
文2：ねこはかわいくない。	= [ね, こ, は, か, わ, い, く, な, い, 。]
文3：あいえええええええええ。	= [あ, い, え, え, え, え, え, え, え, え, え, 。]

言語モデルは、「自然な言語」が何かを表現することができる



言語モデルは、「自然な言語」が何かを表現することができる

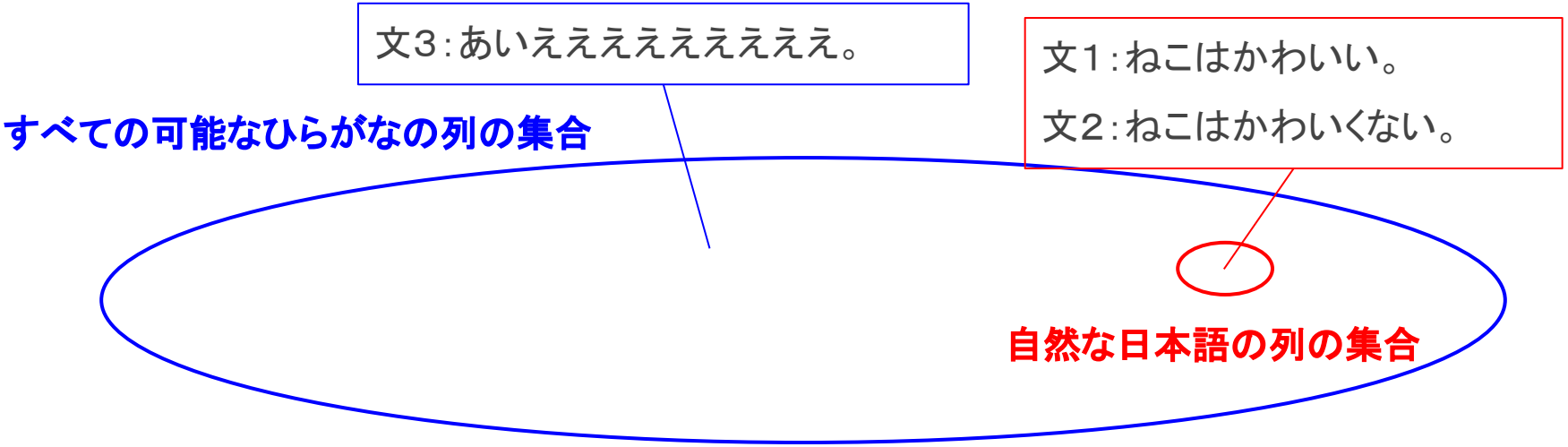


P(自然な日本語の列) >> P(それ以外のひらがなの列)

例: P(文1) = P(文2) = 0.01
P(文3) = 0.000001

実際のLLMの各文章の確率は非常に小さいことが多い。例えば自然な文でも10^{-20}未満、自然ではない文は10^{-100}など。

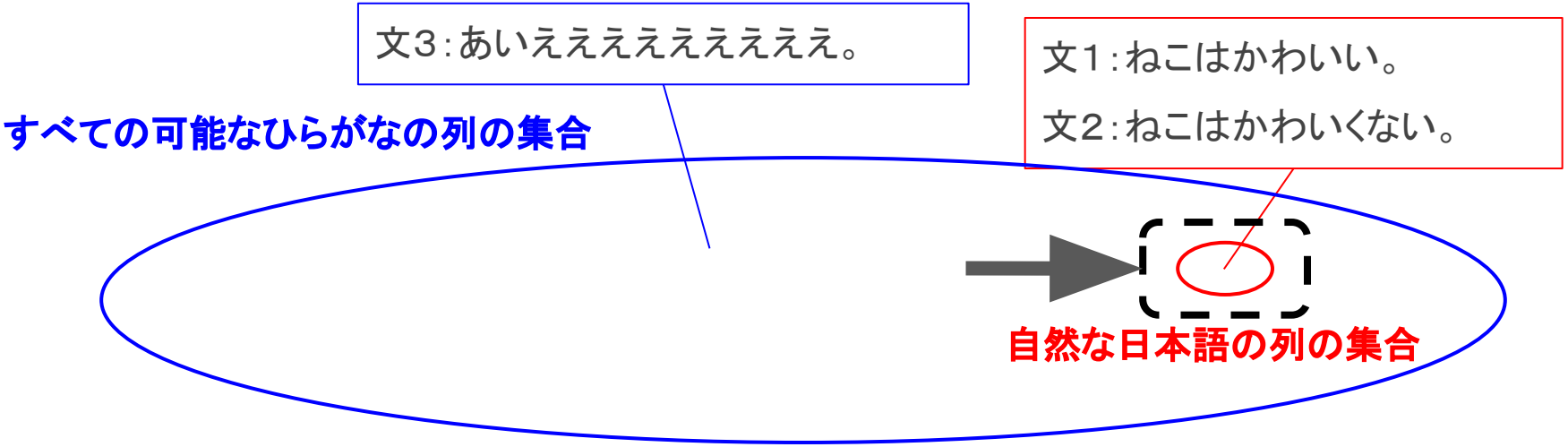
言語モデルは、「自然な言語」が何かを表現することができる



P(自然な日本語の列) >> P(それ以外のひらがなの列)

自然な日本語の列に対して高い確率を与え、それ以外の列に対して（比較的）低い確率を与えることで、「自然な日本語の言語モデル」が得られる

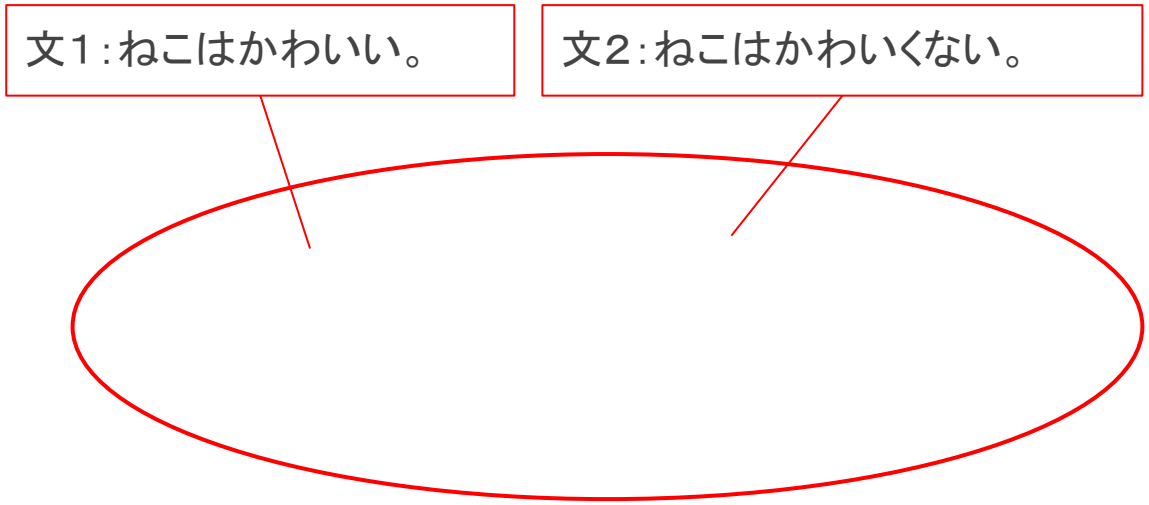
言語モデルは、「自然な言語」が何かを表現することができる



P(自然な日本語の列) >> P(それ以外のひらがなの列)

自然な日本語の列に対して高い確率を与え、それ以外の列に対して（比較的）低い確率を与えることで、「自然な日本語の言語モデル」が得られる

言語モデルは、「自然で望ましい言語」が何かを表現することができる



自然な日本語の列の集合

言語モデルは、「自然で望ましい言語」が何かを表現することができる

文1：ねこはかわいい。

文2：ねこはかわいくない。

自然な日本語の列の集合

キャット・ネコスキー氏

言語モデルは、「自然で望ましい言語」が何かを表現することができる

文1：ねこはかわいい。

文2：ねこはかわいくない。

自然な日本語の列の集合

ねこはすきですか？

キャット・ネコスキー氏

言語モデルは、「自然で望ましい言語」が何かを表現することができる

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

ねこはすきですか？

キャット・ネコスキー氏

言語モデルは、「自然で望ましい言語」が何かを表現することができる

文1：ねこはかわいい。

文2：ねこはかわいくない。

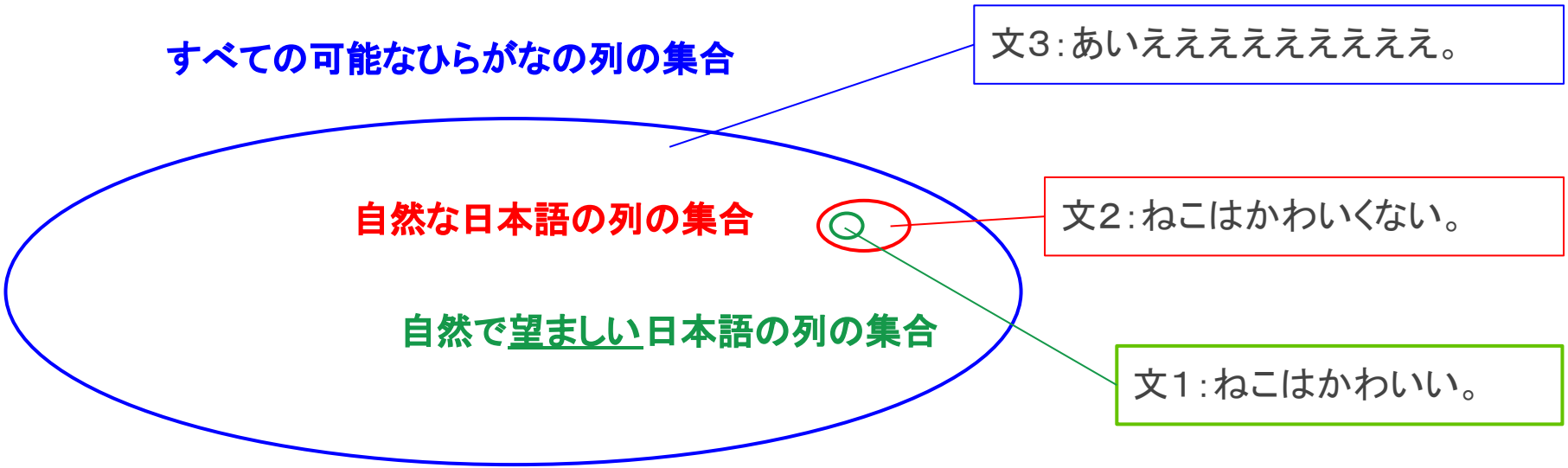
自然で望ましい日本語の列の集合

自然な日本語の列の集合

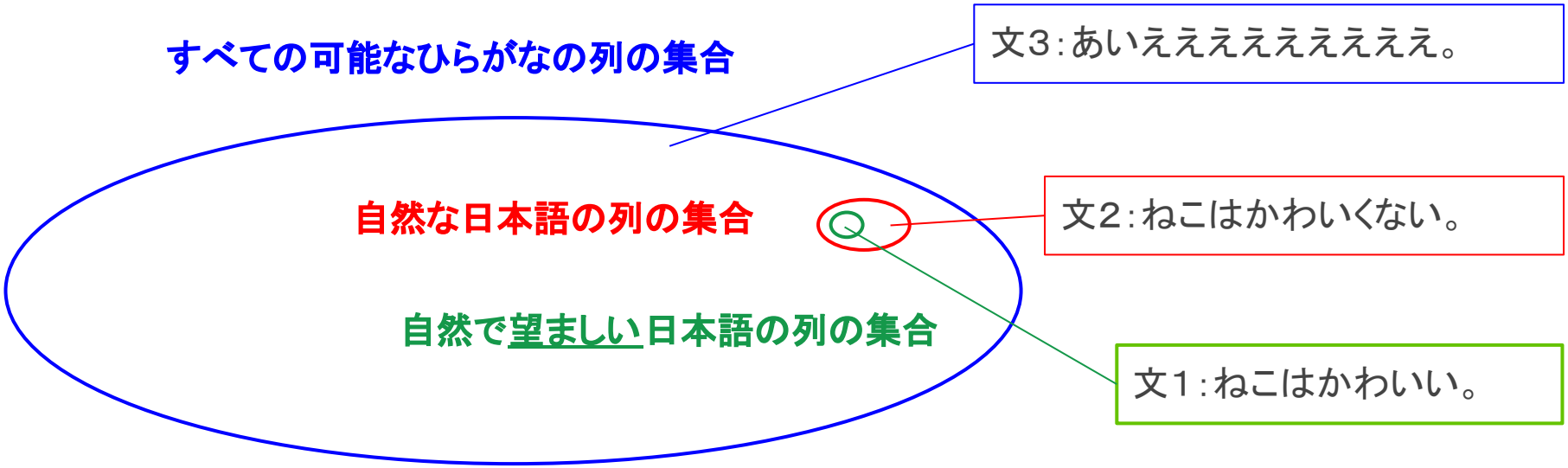
P(自然で望ましい日本語の列) >> P(自然な日本語の列)

キャット・ネコスキー氏

言語モデルは、「望ましい自然な言語」が何かを表現することができる

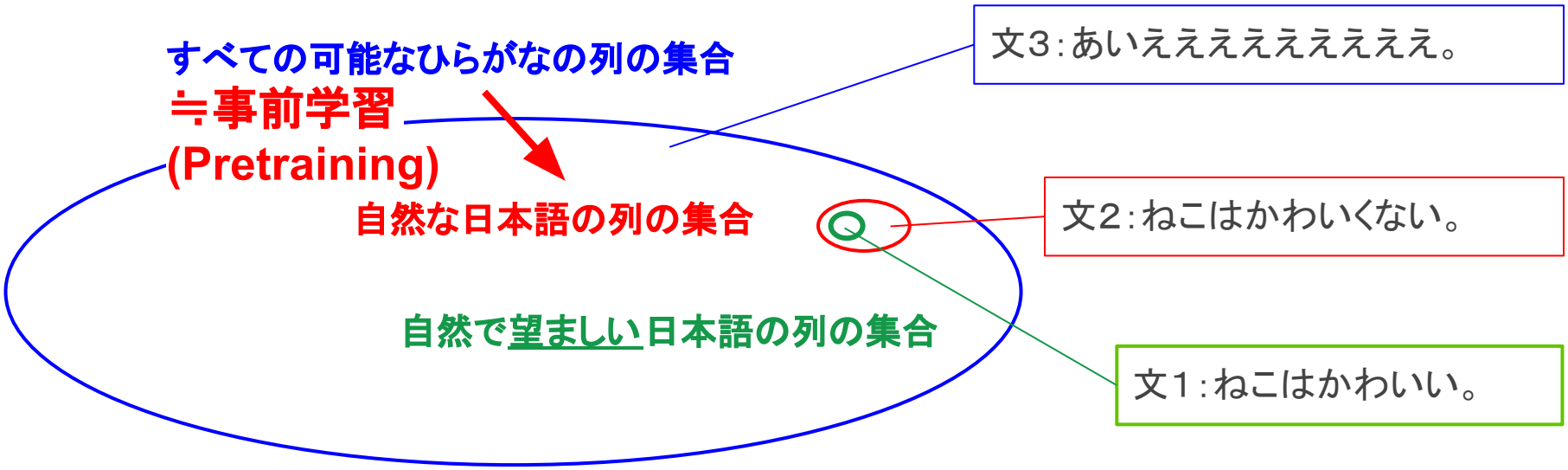


言語モデルは、「望ましい自然な言語」が何かを表現することができる



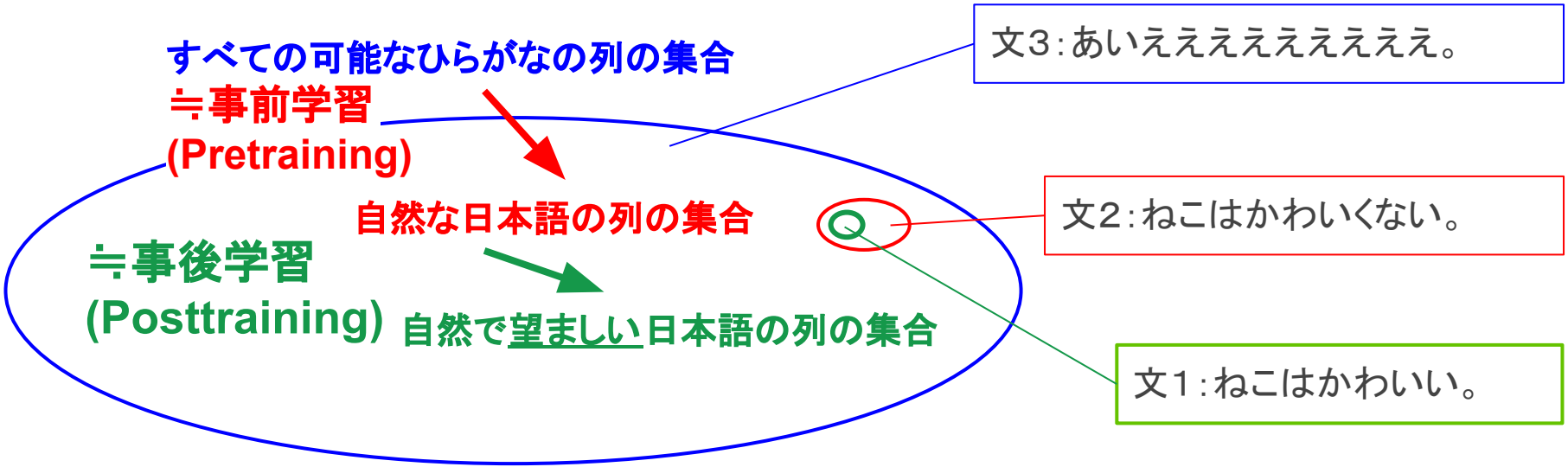
自然で望ましい日本語の列に対して高い確率を与え、それ以外の列に対して（比較的）低い確率を与えることで、「自然で望ましい日本語の言語モデル」が得られる

事前学習と事後学習の役割



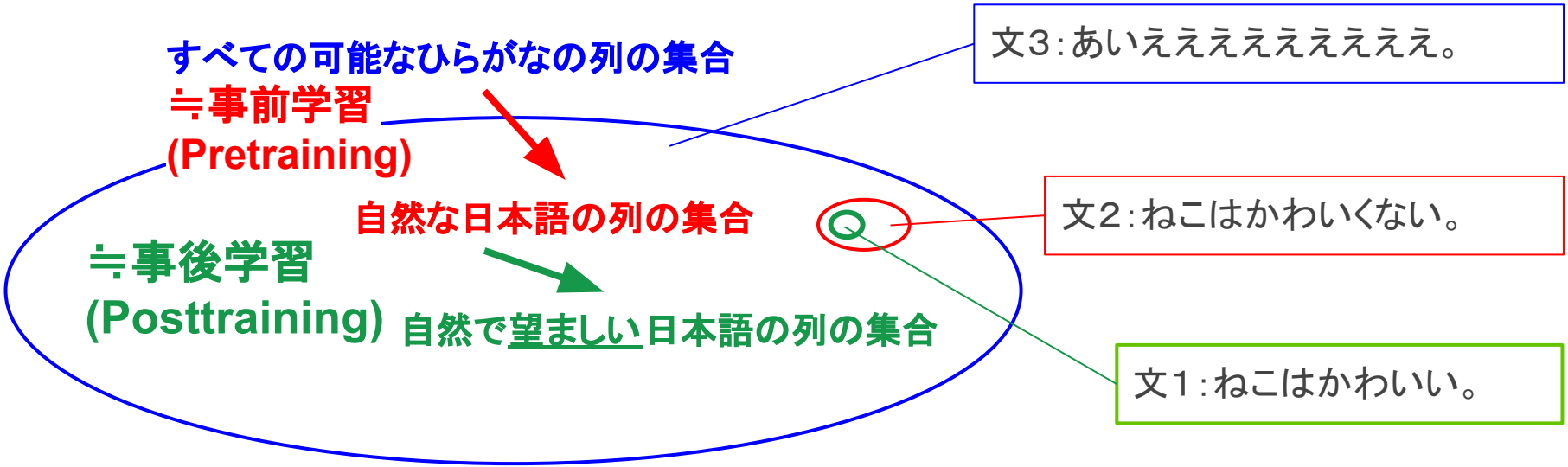
自然で望ましい日本語の列に対して高い確率を与え、それ以外の列に対して（比較的）低い確率を与えることで、「自然で望ましい日本語の言語モデル」が得られる

事前学習と事後学習の役割



自然で望ましい日本語の列に対して高い確率を与え、それ以外の列に対して（比較的）低い確率を与えることで、「自然で望ましい日本語の言語モデル」が得られる

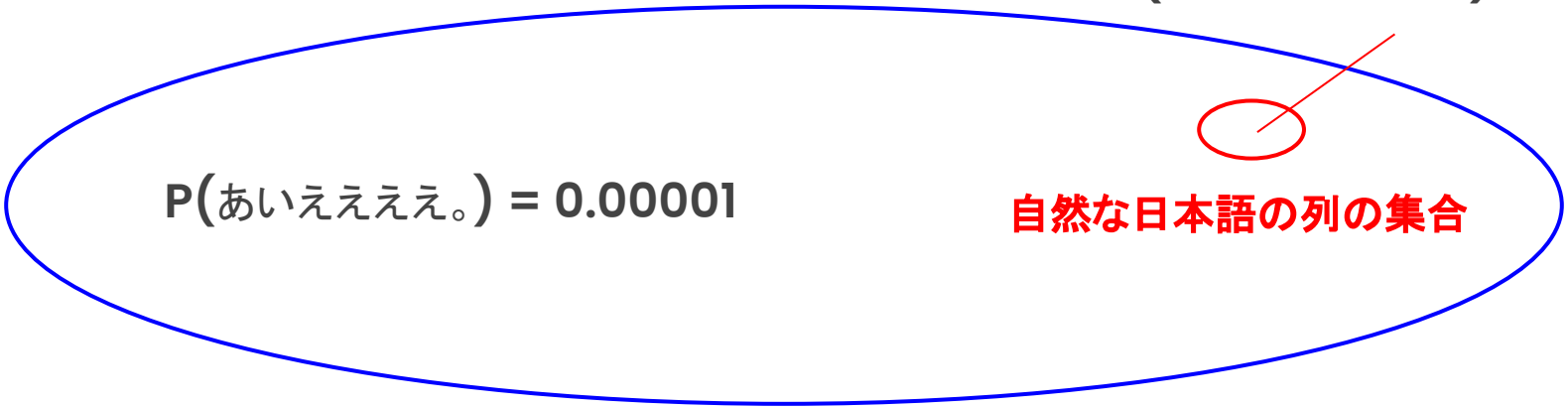
事前学習と事後学習の役割



|すべての可能なひらがなの列の集合 | >>>> |自然で望ましい日本語の列の集合 |

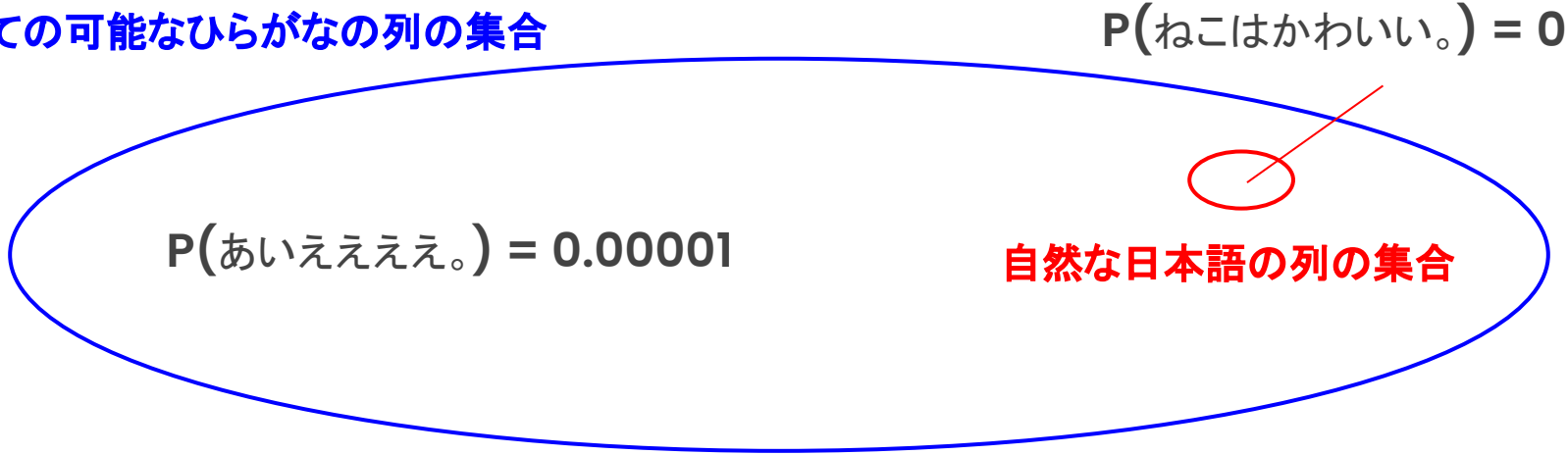
なので一気に学習するのは非常に難しい

言語モデル (Language Model) はどのように学習できるか？



自然な日本語の列に対して高い確率を与え、それ以外の列に対して（比較的）低い確率を与えることで、「自然な日本語の言語モデル」が得られる

言語モデル (Language Model) はどのように学習できるか？



すべての可能なひらがなの列の集合 は膨大（無限集合）だが、

|すべての可能なひらがなの列の集合 | >> |自然な日本語の列の集合 |

自己回帰モデル（Autoregressive model）

自然な日本語は、ひらがなの列に**構造**が存在する

文 = ね こ は か わ い ＿＿ 。

$y_{0..7}$ = y_0 y_1 y_2 y_3 y_4 y_5 y_6 y_7

*Autoregressive modelは古典的には線形のものを指すが、NLPの文脈では非線形モデルも含まれる

自己回帰モデル（Autoregressive model）

自然な日本語は、ひらがなの列に**構造**が存在する

文 = ね こ は か わ い ＿＿＿ 。

$y_{0..7}$ = y_0 y_1 y_2 y_3 y_4 y_5 y_6 y_7

自然な日本語では、y_6が何であるかが前後のトークンから推測できる

*Autoregressive modelは古典的には線形のものを指すが、NLPの文脈では非線形モデルも含まれる

自己回帰モデル（Autoregressive model）

自然な日本語は、ひらがなの列に**構造**が存在する

文 = ね こ は か わ い ＿＿ 。

$\mathbf{y}_{0..7}$ = $\mathbf{y}_0$ $\mathbf{y}_1$ $\mathbf{y}_2$ $\mathbf{y}_3$ $\mathbf{y}_4$ $\mathbf{y}_5$ $\mathbf{y}_6$ $\mathbf{y}_7$

自然な日本語では、$\mathbf{y}_6$が何であるかが前後のトークンから推測できる

$$\mathbf{P}(\mathbf{y}_{0..7}) = \prod_{t=0}^{7} \mathbf{P}(y_t | y_{<t})$$

自己回帰モデル（Autoregressive model）

自然な日本語は、ひらがなの列に**構造**が存在する

文 = ね こ は か わ い ＿＿ 。

$\mathbf{y}_{0..7}$ = $\mathbf{y}_0$ $\mathbf{y}_1$ $\mathbf{y}_2$ $\mathbf{y}_3$ $\mathbf{y}_4$ $\mathbf{y}_5$ $\mathbf{y}_6$ $\mathbf{y}_7$

自然な日本語では、$\mathbf{y}_6$が何であるかが前後のトークンから推測できる

$$\mathbf{P}(\mathbf{y}_{0..7}) = \prod_{t=0}^{7} \mathbf{P}(y_t|y_{<t})$$

自己回帰モデル
（Autoregressive model）

自己回帰モデル（Autoregressive model）

すなわち、

T-1個目のトークンまでが与えられたときに t個目のトークンを予測するモデル（next-token prediction model）があれば言語が表現できる

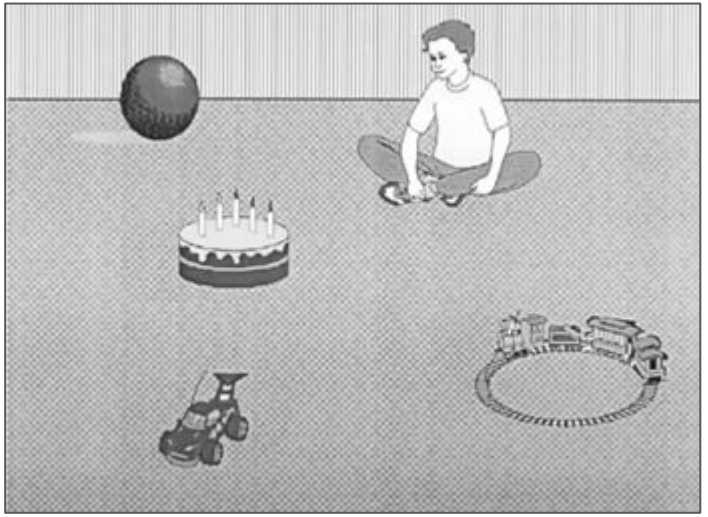
$$\mathbf{P}(\mathbf{y}_{0..7}) = \prod_{t=0}^{7} \mathbf{P}(y_t | y_{<t})$$

自己回帰モデル
（Autoregressive model）

言語モデルは「知能」になるのか？

英単語の列からなる言語モデルを考えよう

- The boy will eat ___ ___.



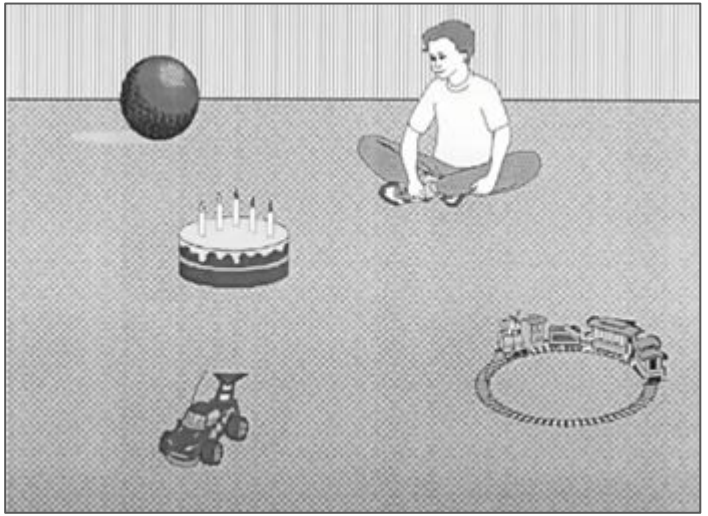
Altmann & Kamide, Cognition 1999

言語モデルは「知能」になるのか？

英単語の列からなる言語モデルを考えよう

- The boy will eat ___ ___.

「次のトークンの**予測**をする」を
繰り返すことで文を生成することができる！

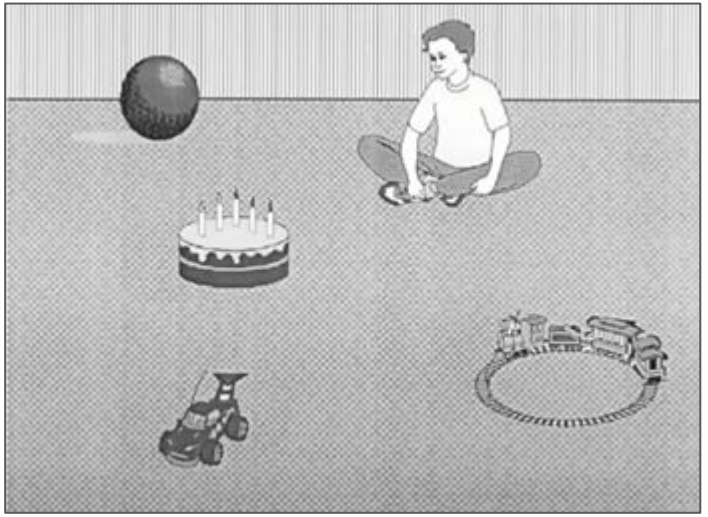


Altmann & Kamide, Cognition 1999

言語モデルは「知能」になるのか？

英単語の列からなる言語モデルを考えよう

- The boy will eat ___ ___.
- The boy will move ___ ___.



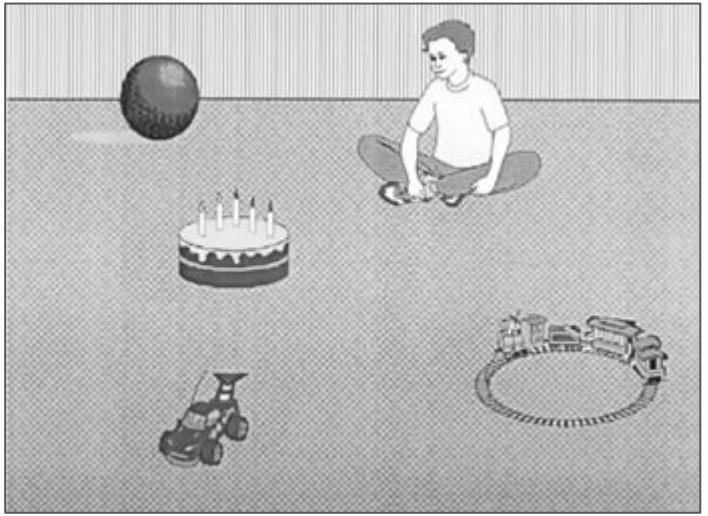
Altmann & Kamide, Cognition 1999

言語モデルは「知能」になるのか？

英単語の列からなる言語モデルを考えよう

- The boy will eat ___ ___.
- The boy will move ___ ___.

**「次の単語を予測すること」は
状況を理解し、起こる現象を予測する
ことにつながる**



Altmann & Kamide, Cognition 1999

言語モデルは「知能」になるのか？

英単語の列からなる言語モデルを考えよう

Q. Is the animal in the image
dog or cat?

A. He is ____ ____ .

Image from
https://www.atchoumthecat.com/

言語モデルは「知能」になるのか？

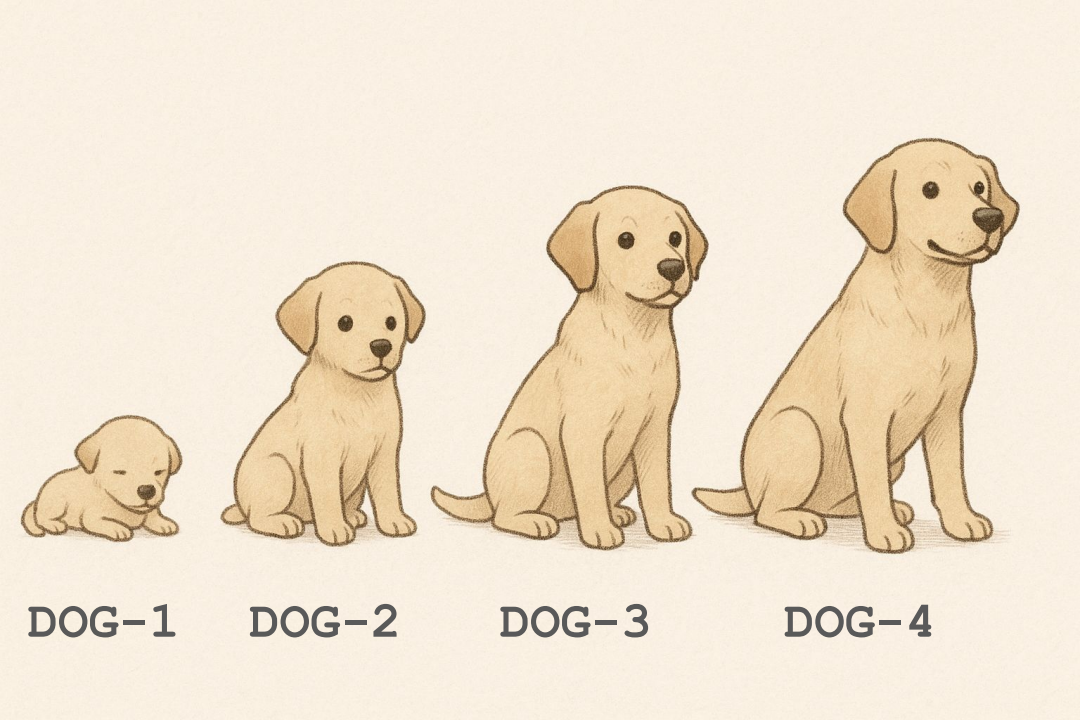
英単語の列からなる言語モデルを考えよう

Q. Is the animal in the image
dog or cat?

A. He is a cat .

Image from
https://www.atchoumthecat.com/

大規模言語モデルはどのように生まれたか



OpenAIの言語モデルとRLHFの研究開発の過程

2017/06: Attention is All You Need

2018/06: (GPT-1) Language Understanding by Generative Pretraining

2019/02: (GPT-2) LLMs are Unsupervised Multitask Learners

2019/09: (RLHF) Fine tuning LLMs from Human Preferences

2020/05: (GPT-3) LLMs are Few-Shot Learner

2020/09: (RLHF) LLMs can learn to summarize by RL

2022/03: (RLHF) Instruct GPT; LLMs can learn to follow instructions by RL

2022/11: (ChatGPT; GPT-3.5)

2023/03: (GPT-4) GPT-4 Technical Report

OpenAIの言語モデルとRLHFの研究開発の過程

2017/06: Attention is All You Need

2018/06: (GPT-1) Language Understanding by Generative Pretraining

2019/

2019/

2020/

2020/

2022/

2022/11: (ChatGPT; GPT-3.5)

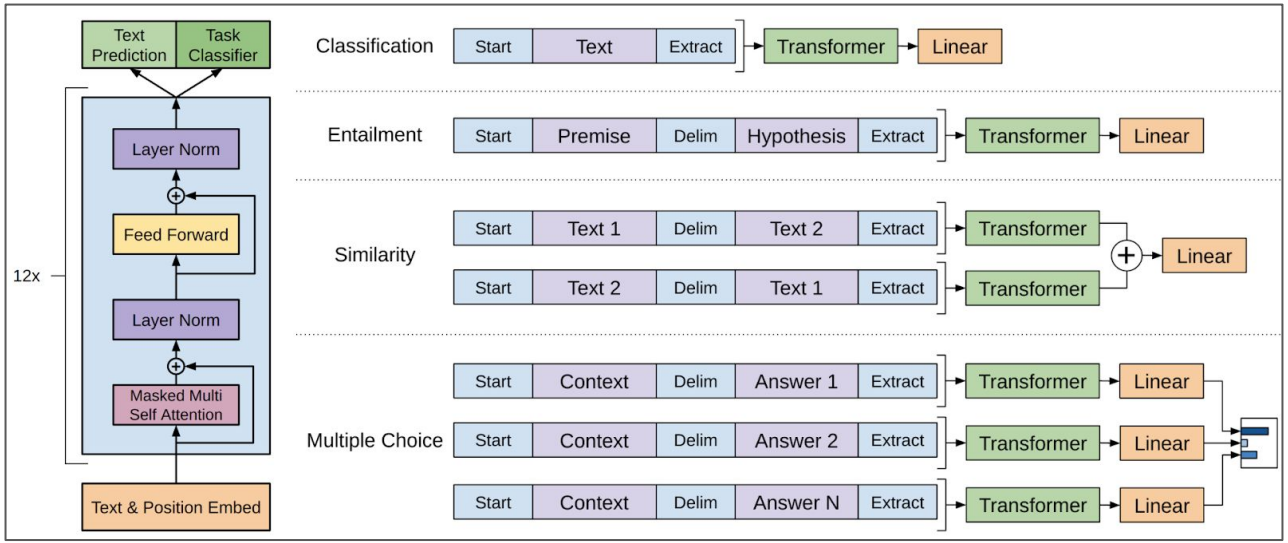
2023/03: (GPT-4) GPT-4 Technical Report

とりあえず読んでおこう！！！！
LLMがこれからの人類の基礎技術の
一つになる可能性は高い

2018/06: (GPT-1) Language Understanding by Generative Pretraining

(Radford et al. OpenAI 2018)

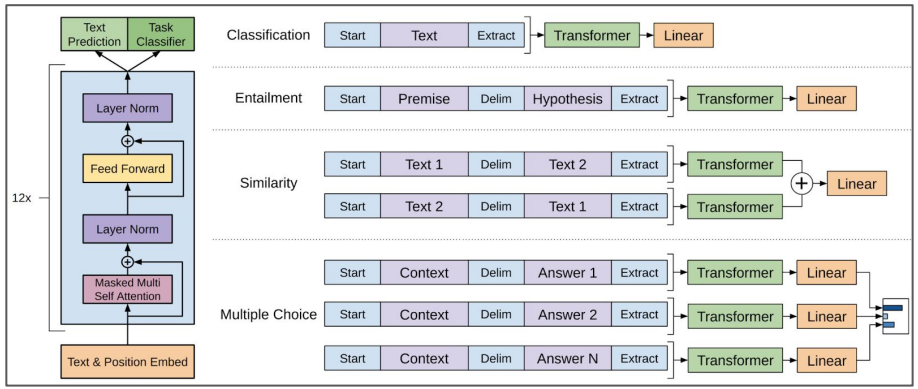
- Transformerを教師なし学習 (pretraining) させた言語モデル (LM)が幅広いタスクに適用できる基盤モデルになることを示した

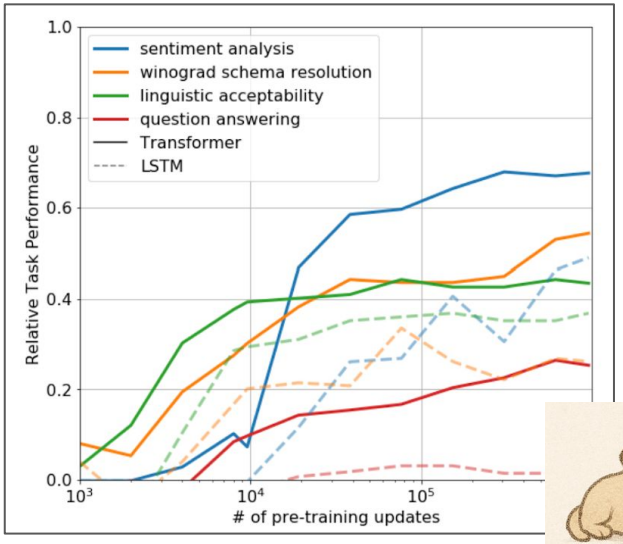


2018/06: (GPT-1) Language Understanding by Generative Pretraining

(Radford et al. OpenAI 2018)

- Transformerを教師なし学習 (pretraining) させた言語モデル (LM)が幅広いタスクに適用できる基盤モデルになることを示した
- Pretrainingの有用性自体は機械翻訳などで知られていた
- データセット: 7000冊の本; 1.2GB (BooksCorpus)
- モデル: 117 M parameters





2018/06: (GPT-1) Language Understanding by Generative Pretraining

(Radford et al. OpenAI 2018)

- Transformerを教師なし学習 (pretraining) させた言語モデル (LM)が幅広いタスクに適用できる基盤モデルになることを示した
- Pretrainingの有用性自体は機械翻訳などで知られていた
- データセット: 7000冊の本; 1.2GB (BooksCorpus)
- モデル: 117 M parameters

参考:

- AtariのRLエージェントは1-10M parameters
- 囲碁・ChessなどのRLエージェントは~50M parameters (推定)

2019/02: (GPT-2) LMs are Unsupervised Multitask Learners

(Radford et al. OpenAI 2019)

- 言語モデルは、webの文章を大量に事前学習 (教師なし学習) をすれば

  Fine-tuningなしで様々なタスクを解けることを示した

入力文

> Fixing /.ssh/ Permissions in Cygwin
>
> In Cygwin on Windows I found that I could not set the permissions of my /.ssh/ folder to be 0600, as is required for ssh to allow you to use keys. The symptom I had was that I no matter what I did, it always modfied the owner and the group

GPT-2による生成文

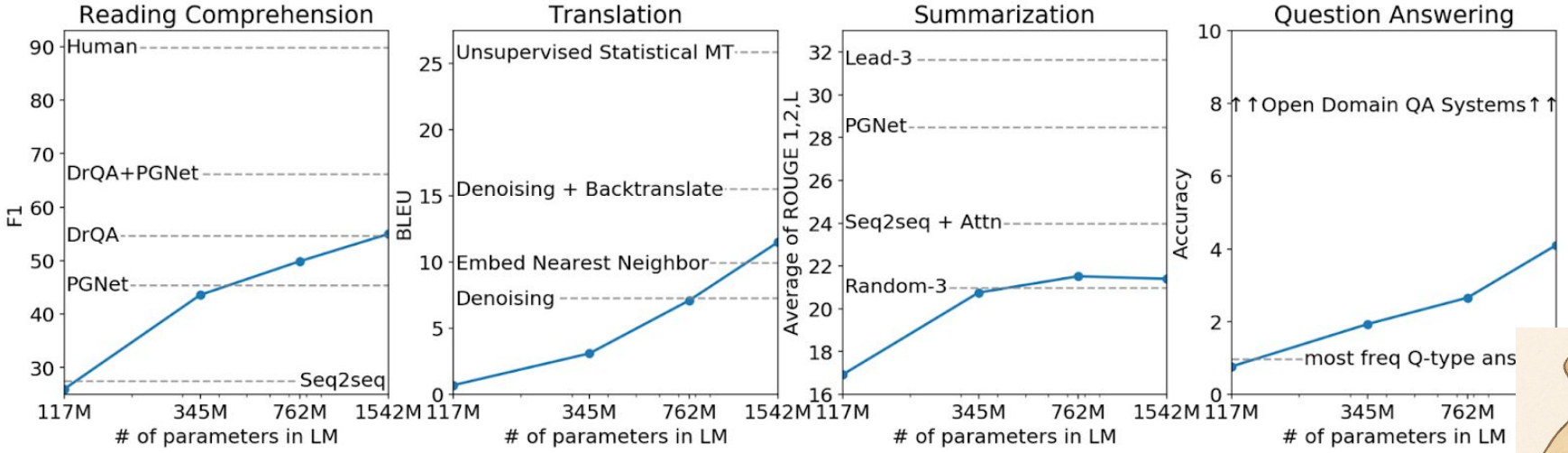
> , making these files read-only. I tried just starting ssh and using a program like passwd which modifies the owner/group to 0644 permissions, but it didn't work. It was annoying but not nearly as bad as it could be, just annoying to have to fiddle with permissions.

2019/02: (GPT-2) LMs are Unsupervised Multitask Learners

(Radford et al. OpenAI 2019)

- 言語モデルは、webの文章を大量に事前学習 (教師なし学習) をすれば

  Fine-tuningなしで様々なタスクを解けることを示した



2019/02: (GPT-2) LMs are Unsupervised Multitask Learners

(Radford et al. OpenAI 2019)

- 言語モデルは、webの文章を大量に学習すれば Fine-tuningなしで様々なタスクを解けることを示した
- データセット: 8M documents linked from Reddit; 40GB (WebText dataset; 自家製)
- CommonCrawlはクオリティが低くうまく行かなかった
- モデル: 1.5 B parameters
- 学習時間: H100が8日

https://x.com/karpathy/status/1811467135279104217

Q. 何故学習なしに解答出来るのか？

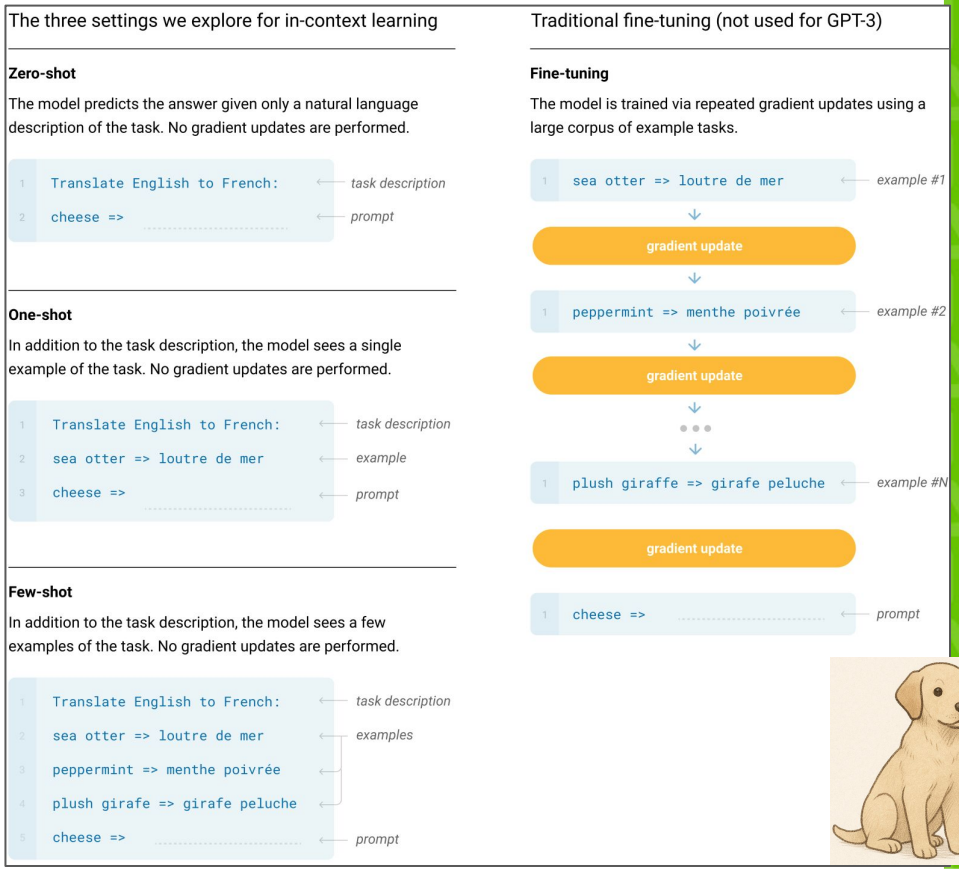
A. データセットの中に関連するタスクがあるからではないか

> ”I’m not the cleverest man in the world, but like they say in French: **Je ne suis pas un imbecile [I’m not a fool].**
>
> In a now-deleted post from Aug. 16, Soheil Eid, Tory candidate in the riding of Joliette, wrote in French: ”**Mentez mentez, il en restera toujours quelque chose**,” which translates as, ”**Lie lie and something will always remain.**”
>
> “I hate the word ‘**perfume**,”’ Burr says. ‘It’s somewhat better in French: ‘**parfum**.’

2020/05: (GPT-3) LMs are Few-Shot Learner

(Ouyang et al. NeurIPS 2020)

- 言語モデルが**Prompt文にタスクの例を入れることで性能が上がる(Few-shot learningが出来る)ことを示した**

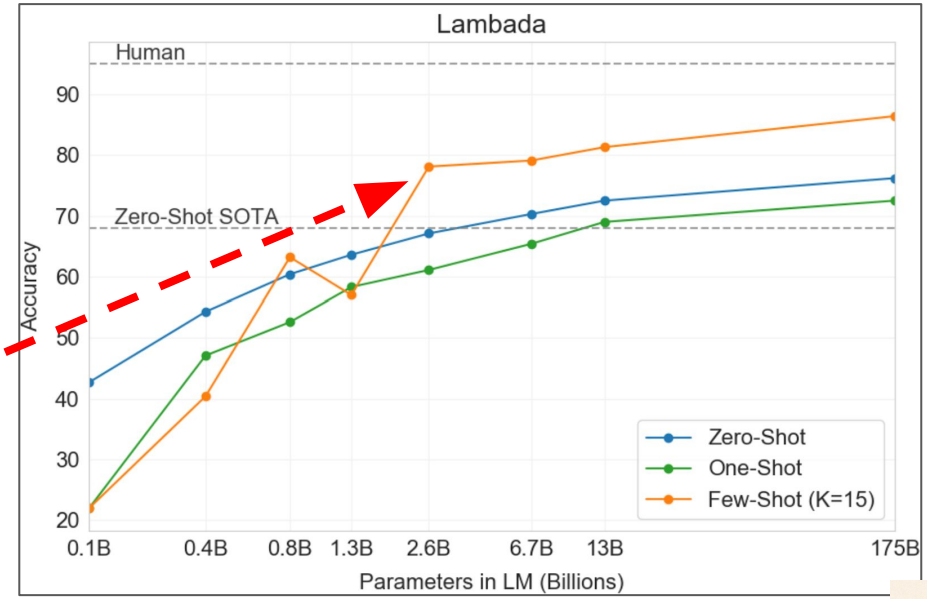


2020/05: (GPT-3) LMs are Few-Shot Learner

(Ouyang et al. NeurIPS 2020)

- 言語モデルが**Prompt文にタスクの例を入れることで性能が上がる(Few-shot learningが出来る)ことを示した**

2.6BモデルのFew-shotが175BのZero-shot, One-shotよりも高い精度を達成



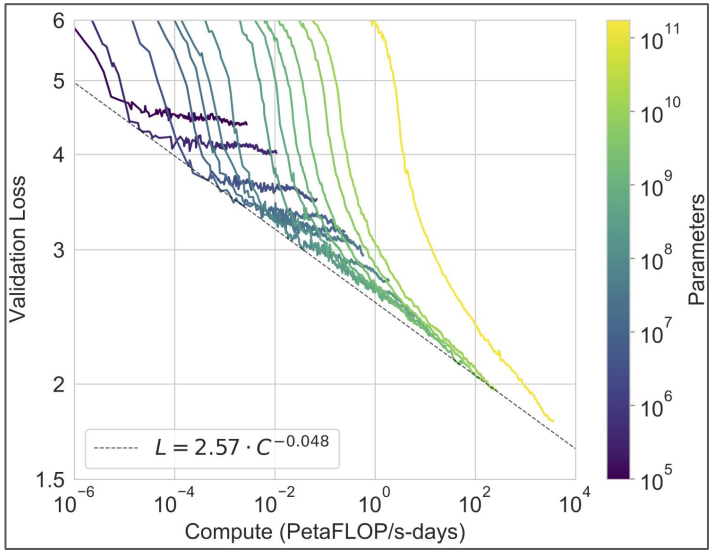
2020/05: (GPT-3) LMs are Few-Shot Learner

(Ouyang et al. NeurIPS 2020)

GPT-3の学習

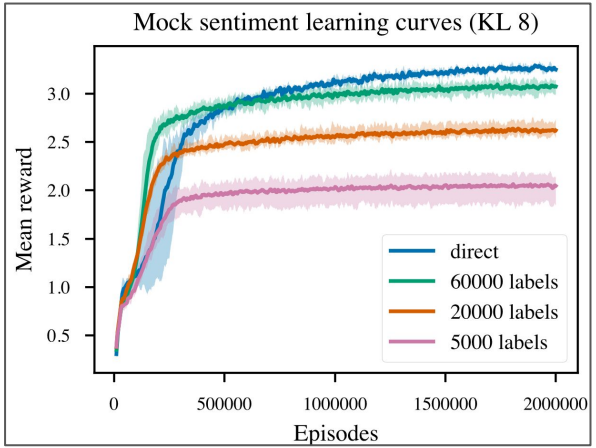
- データセット: Common Crawlを綺麗にしたデータセット等; 570GB (自家製; 非公開)
- モデル: 175 B parameters
- 学習時間: A100が5800日分

Dataset	Quantity (tokens)	Weight in training mix	Epochs elapsed when training for 300B tokens
Common Crawl (filtered)	410 billion	60%	0.44
WebText2	19 billion	22%	2.9
Books1	12 billion	8%	1.9
Books2	55 billion	8%	0.43
Wikipedia	3 billion	3%	3.4



2019/09: Fine tuning LLMs from Human Preferences

- LMを「2つの文章のうちどちらがより好ましいか？」をアノテートしたデータを使って強化学習することによって文章要約タスクを効率的に性能を改善することができることを示した
- データセット: 20k/60kのHuman Feedback
- モデル: 774 M parameters (GPT-2)

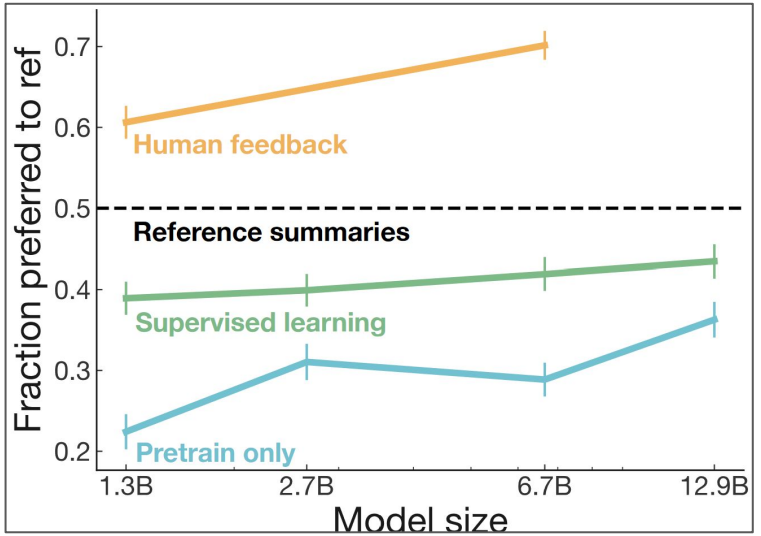


	TL;DR				CNN/Daily Mail			
	R-1	R-2	R-L	R-AVG	R-1	R-2	R-L	R-AVG
SOTA	22*	5*	17*	14.7*	41.22	18.68	38.34	32.75
lead-3	17.435	3.243	14.575	11.751	**40.379**	**17.658**	36.618	31.552
zero-shot	15.862	2.325	13.518	10.568	28.406	8.321	25.175	20.634
supervised baseline	17.535	3.124	14.969	11.877	39.525	16.992	36.728	31.082
supervised + 60k fine-tune	**18.434**	**3.542**	**15.457**	**12.478**	40.093	17.611	**37.104**	**31.603**
60k fine-tune	16.800	2.884	14.011	11.232	37.385	15.478	33.330	28.731
30k fine-tune	16.410	2.920	13.653	10.994	35.581	13.662	31.734	26.992
15k fine-tune	15.275	2.240	12.872	10.129	38.466	15.960	34.468	29.631
60k offline fine-tune	16.632	2.699	13.984	11.105	33.860	12.850	30.018	25.576

2020/09: LLMs can learn to summarize by RL

(Stienno et al. NeurIPS 2020)

- 「2つの文章のうちどちらがより好ましいか？」をアノテートしたデータを使って強化学習することによって文章要約タスクでSoTAの性能を達成できることを示した

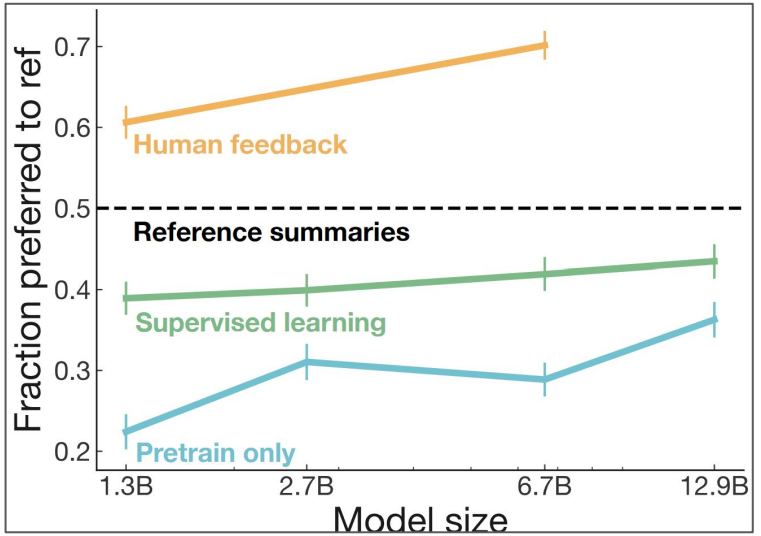


2020/09: LLMs can learn to summarize by RL

(Stienno et al. NeurIPS 2020)

- 「2つの文章のうちどちらがより好ましいか？」をアノテートしたデータを使って強化学習することによって文章要約タスクでSoTAの性能を達成できることを示した
- エキスパートデータを使った教師あり学習よりも強化学習の方が効率が良かった

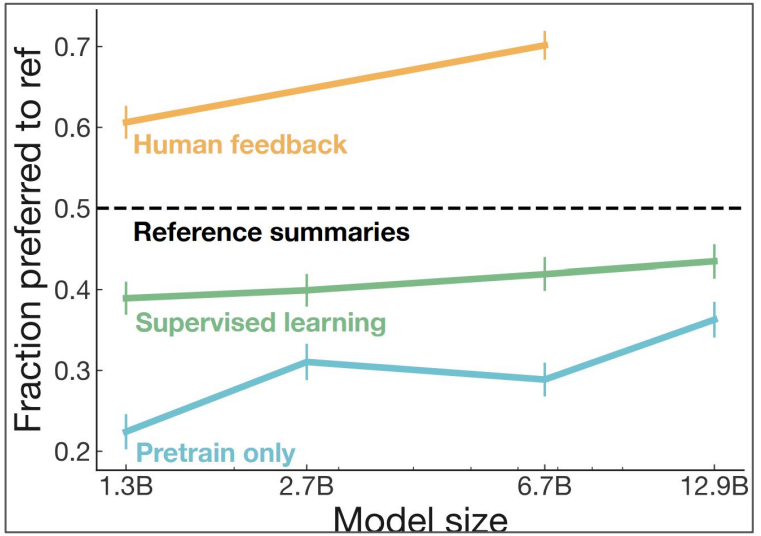
エキスパート=BBCのWriter



2020/09: LLMs can learn to summarize by RL

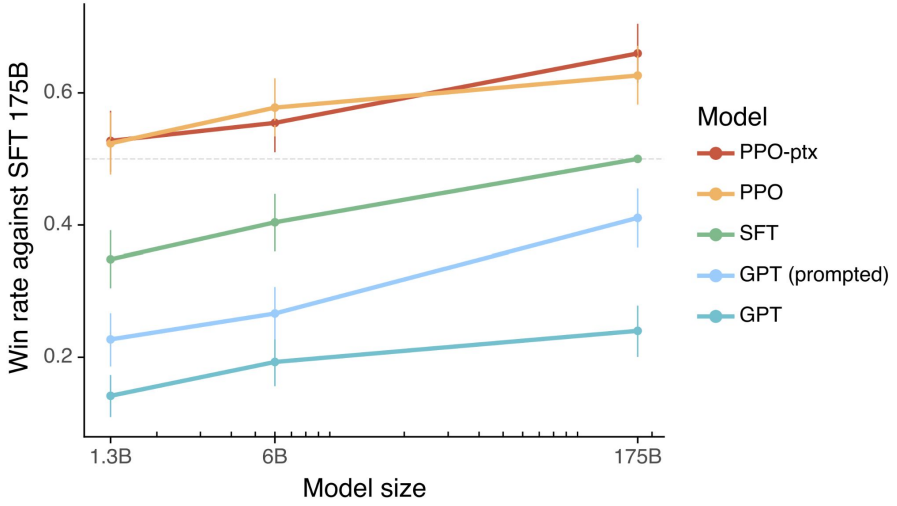
(Stienno et al. NeurIPS 2020)

- 「2つの文章のうちどちらがより好ましいか？」をアノテートしたデータを使って強化学習することによって文章要約タスクでSoTAの性能を達成できることを示した
- エキスパートデータを使った教師あり学習よりも強化学習の方が効率が良かった
- データセット: 数千時間による人手アノテーション
- モデル: 6.7 B parameters
- 計算時間: 320 GPU days



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

- 強化学習によって、特定のタスク (文章要約) だけではなく指示された幅広い様々なタスクが解けることを示した



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

- 強化学習によって、特定のタスク (文章要約) だけではなく指示された幅広い様々なタスクが解けることを示した
- データセット: ~120kのHuman Feedback
- モデル: 175 B parameters (GPT-3)
- 学習時間:
  A100を100日ほど

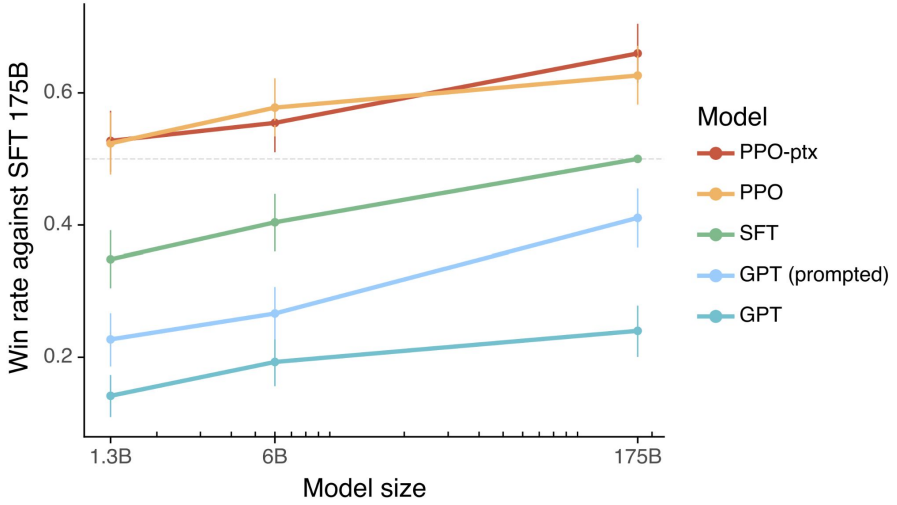


Table 6: Dataset sizes, in terms of number of prompts.

SFT Data			RM Data			PPO Data		
split	source	size	split	source	size	split	source	size
train	labeler	11,295	train	labeler	6,623	train	customer	31,144
train	customer	1,430	train	customer	26,584	valid	customer	16,185
valid	labeler	1,550	valid	labeler	3,488			
valid	customer	103	valid	customer	14,399			

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

- 強化学習によって、特定のタスク (文章要約) だけではなく指示された幅広い様々なタスクが解けることを示した
- データセット: ~120kのHuman Feedback
- モデル: 175 B parameters (GPT-3)
- 学習時間:
  A100を100日ほど
  (ベースモデルは 5800日)

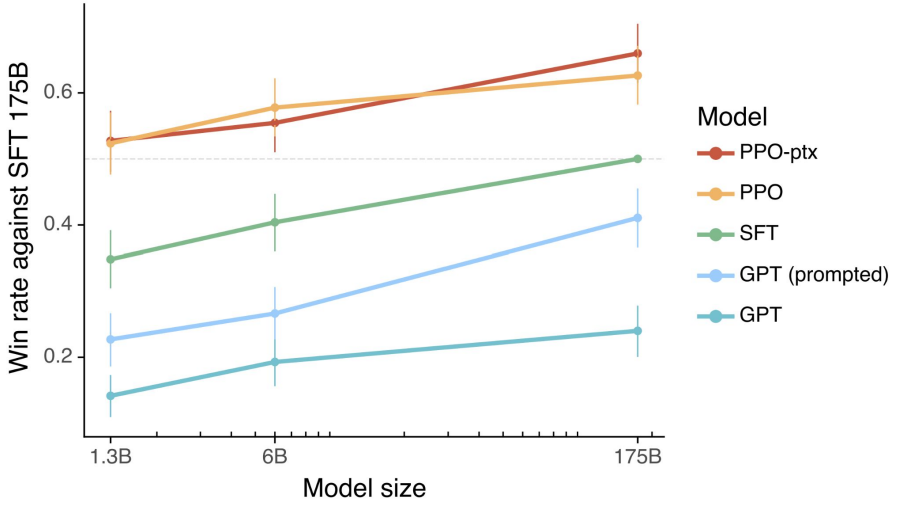
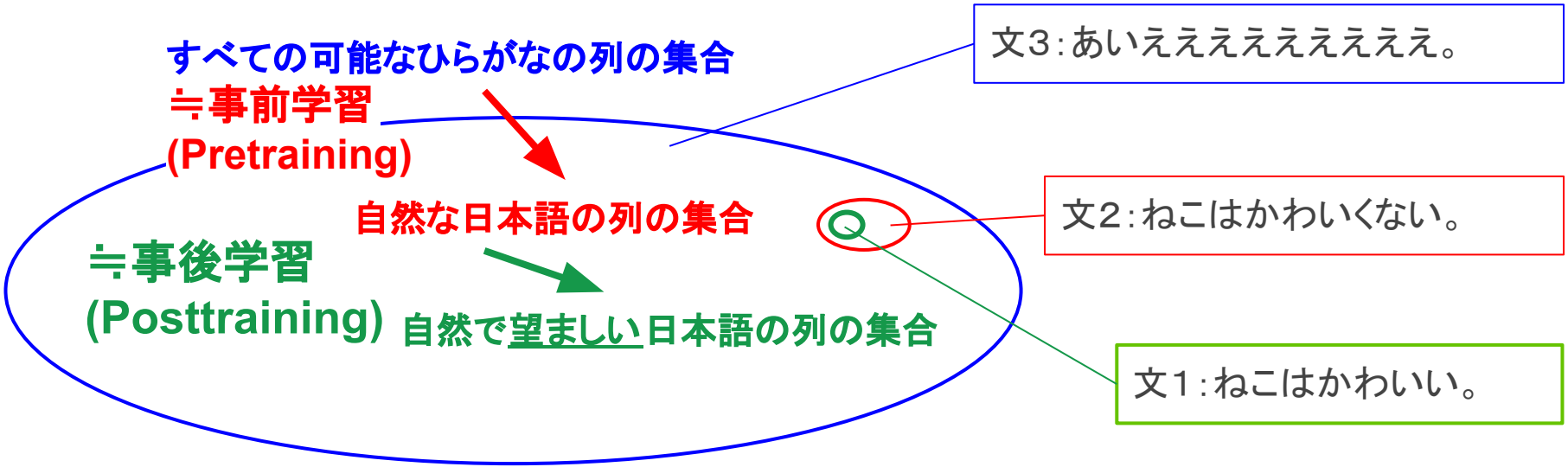


Table 6: Dataset sizes, in terms of number of prompts.

SFT Data			RM Data			PPO Data		
split	source	size	split	source	size	split	source	size
train	labeler	11,295	train	labeler	6,623	train	customer	31,144
train	customer	1,430	train	customer	26,584	valid	customer	16,185
valid	labeler	1,550	valid	labeler	3,488			
valid	customer	103	valid	customer	14,399			

まとめ：大規模言語モデルの学習工程



「次のトークンを予測する」という言語モデルを学習（事前学習）をしたのちに
目的となるタスクのために望ましい文章を学習（事後学習）することが現代のパラダイム

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。

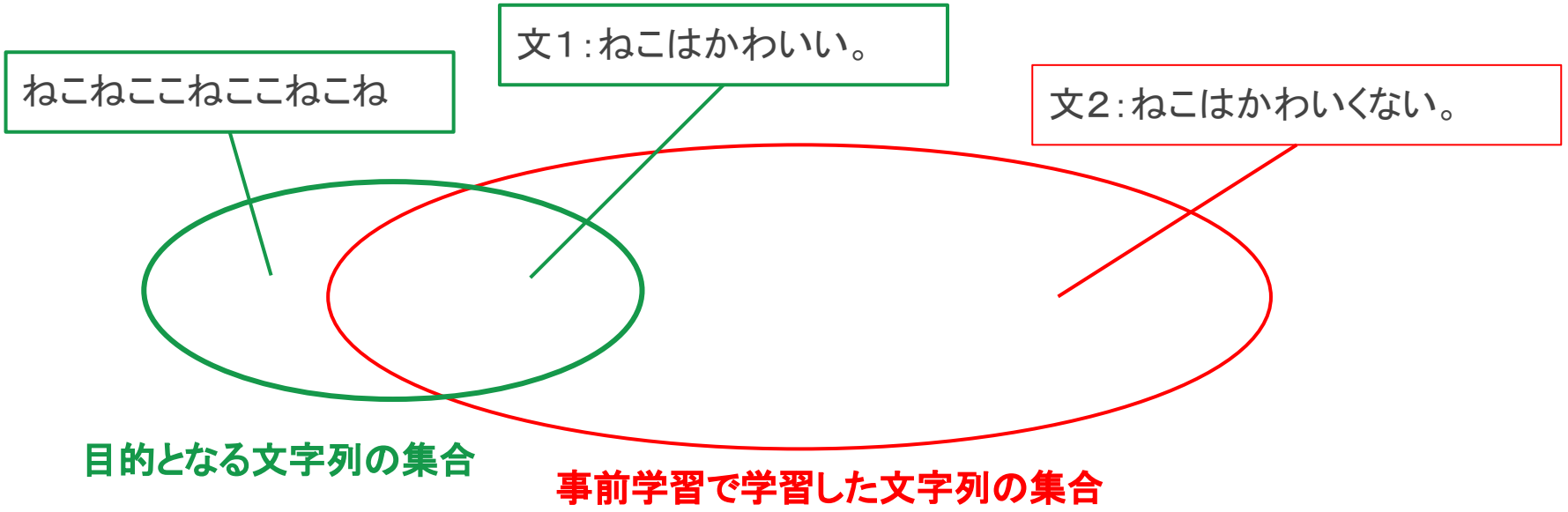
自然で望ましい日本語の列の集合

自然な日本語の列の集合

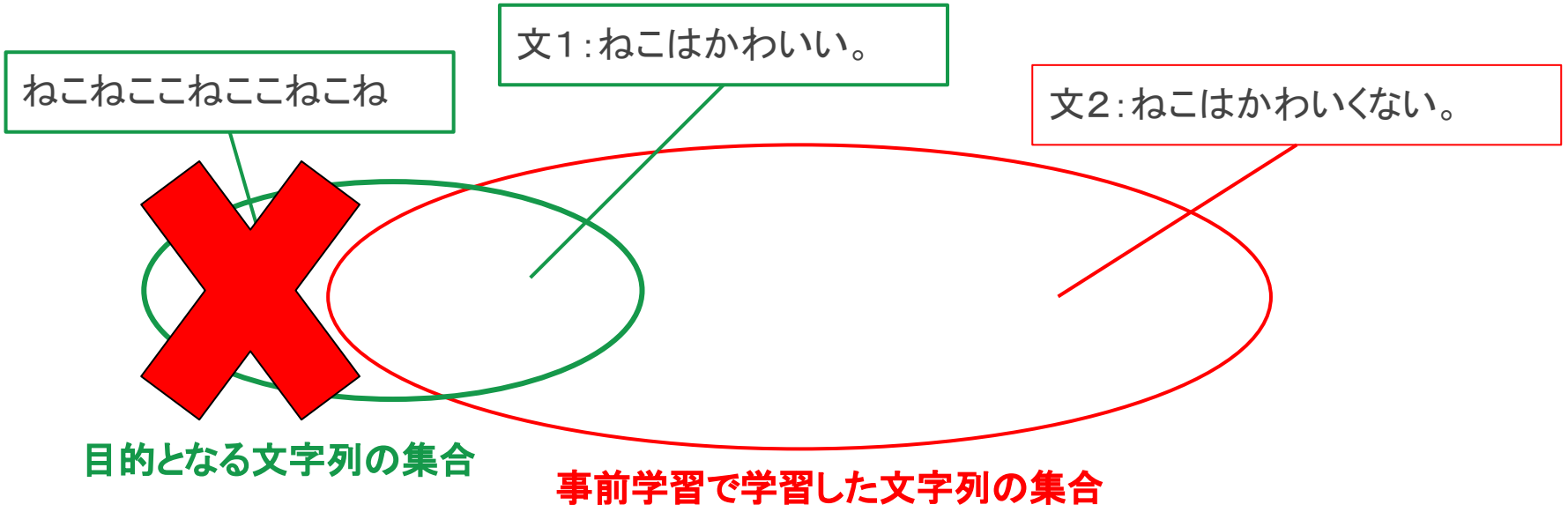
ねこはすきですか？

キャット・ネコスキー氏

強化学習では出来ないことは？



強化学習では出来ないことは？



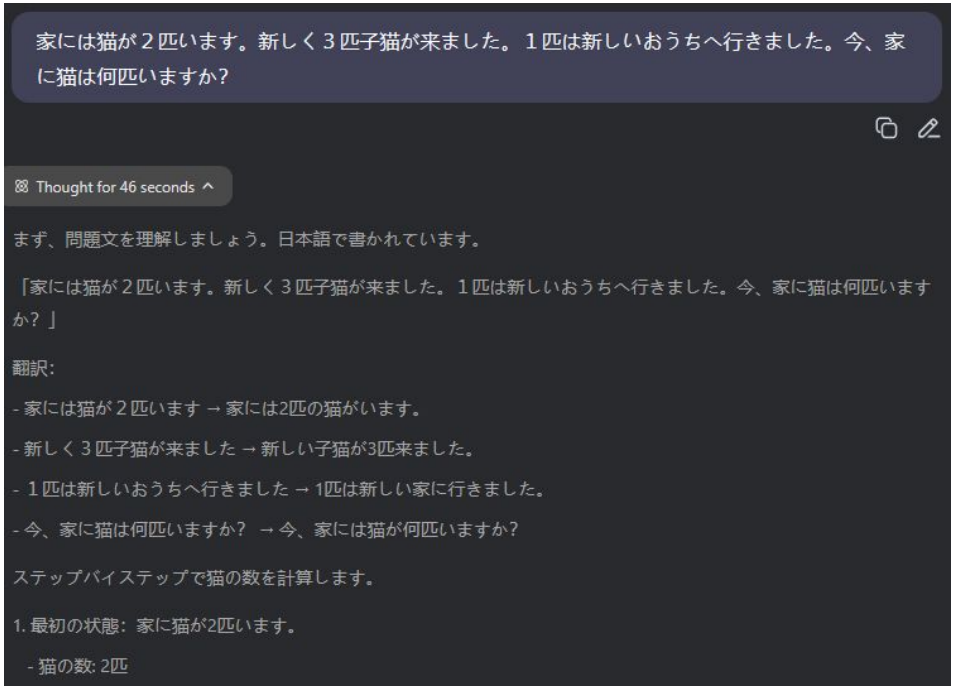
AtariやAlphaGoなどとは目的が異なる

Priorが存在すること
元々強化学習の研究は「何も前提知識がないところから知識を獲得させたい」という目標があった (Sutton&Barto)

LLMのファインチューニング（微調整）とは大きく目的が異なる
元々Prior仮説は言われていたが、LLMでそれが表出した

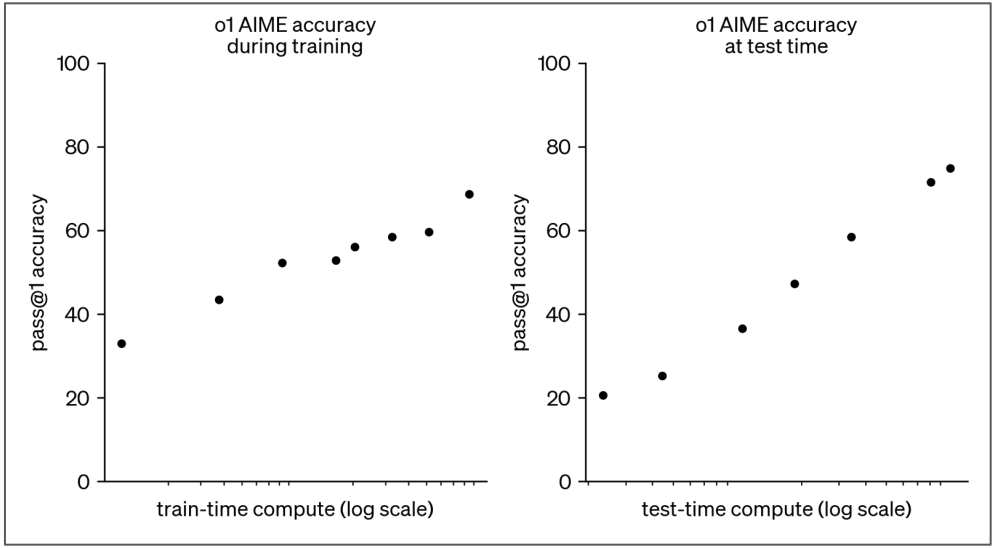
今、Inference-Time Scalingがアツい！

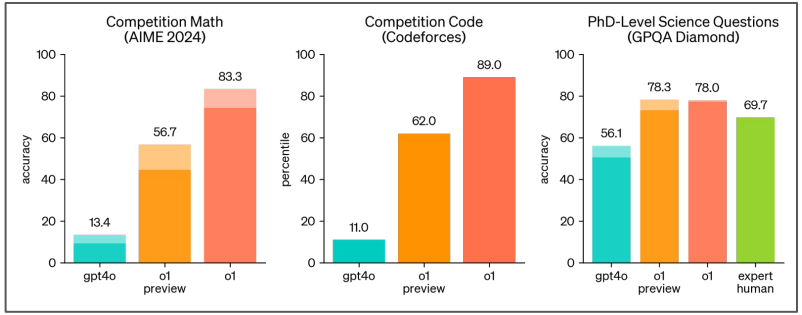
Inference-Time Scaling:
回答を生成する際に計算時間をかけることでより良い回答を得る手法の総称



2024/9: (o1) Inference-Time Scaling

- 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

- **Reasoning**を考慮した強化学習によって性能が向上
- 「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
- 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**モデルが得られた