大语言模型和提示范式

常宝宝
北京大学计算语言学研究所
chbb@pku.edu.cn

主要内容

- **模型规模和模型性能**
- 提示范式
- 提示和提示工程
- 涌现能力
- 指令微调
- 对齐微调
- 参数高效微调方法

模型参数规模

- 模型的参数规模越来越大
  - ELMo: 93M params, 2-layer biLSTM
  - GPT: 117M params, 12-layer Transformer
  - BERT-base: 110M params, 12-layer Transformer
  - BERT-large: 340M params, 24-layer Transformer
  - GPT-3 175B params, 96-layer transformer
  - PaLM 540B params, 118-layer transformer
  - …

训练数据规模

- 训练数据的规模越来越大
  - ELMo: 1B training tokens
  - BERT: 3.3B training tokens
  - RoBERTa: 30B training tokens
  - GPT-3: 300B training tokens
  - PaLM: 780B training tokens
  - ….

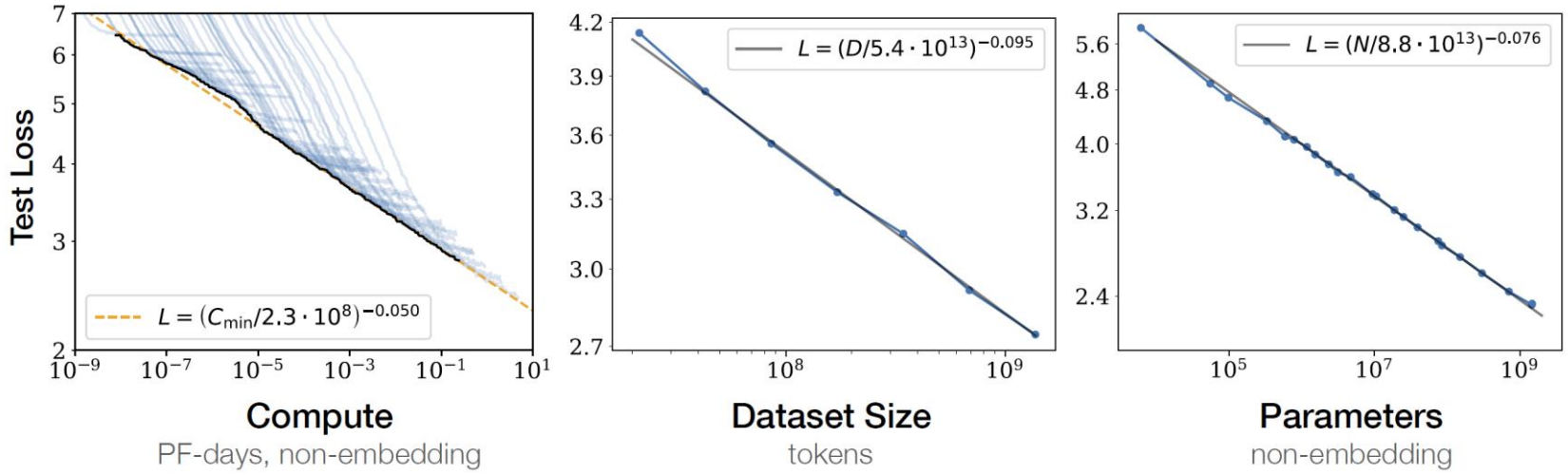
- 实践中，常常基于同样的数据规模，训练出参数规模不同的模型，如GPT-3系列
  - GPT-3 125M/350M/760M/1.3B/2.7B/6.7B/13B/175B

模型算力成本

- 训练模型算力投入也在持续增加
  - 以训练所需浮点运算数衡量 (FLOPs)
  - GPT-3 125M **2.25E+20**
  - GPT-3 350M **6.41E+20**…
  - GPT-3 175B **3.14E+23**
  - PaLM 8B **3.74E+22**
  - PaLM 62B **2.90E+23**
  - PaLM 540B **2.53E+24**
  - **……**

KM scaling law

- 模型性能较少收到模型结构的影响
- 模型性能主要依赖模型的规模(性能和规模关系符合幂律)



Jared Kaplan, Sam McCandlish et al. Scaling Laws for Neural Language Models. 2020.

KM scaling law

- 模型性能较少收到模型结构的影响
- 模型性能主要依赖模型的规模(性能和规模关系符合幂律)

$$L(N) = \left(\frac{N_c}{N}\right)^{\alpha_N}, \alpha_n \sim 0.076, N_c \sim 8.8 \times 10^{13}$$

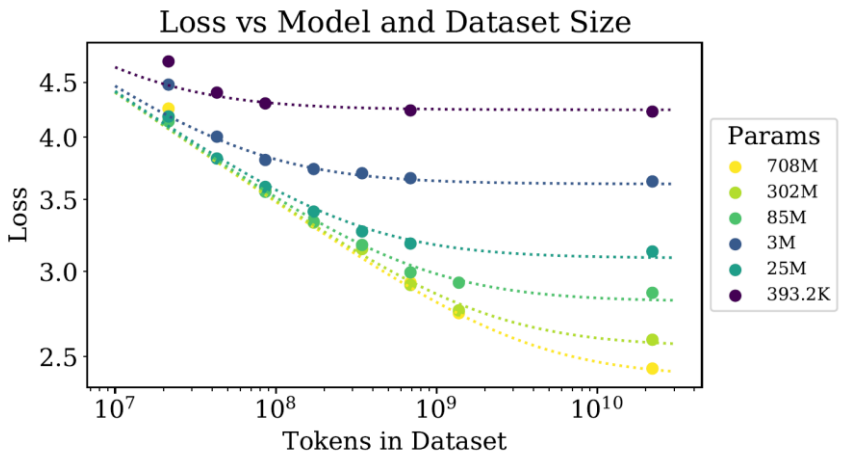
$$L(D) = \left(\frac{D_c}{D}\right)^{\alpha_D}, \alpha_D \sim 0.095, N_D \sim 5.4 \times 10^{13}$$

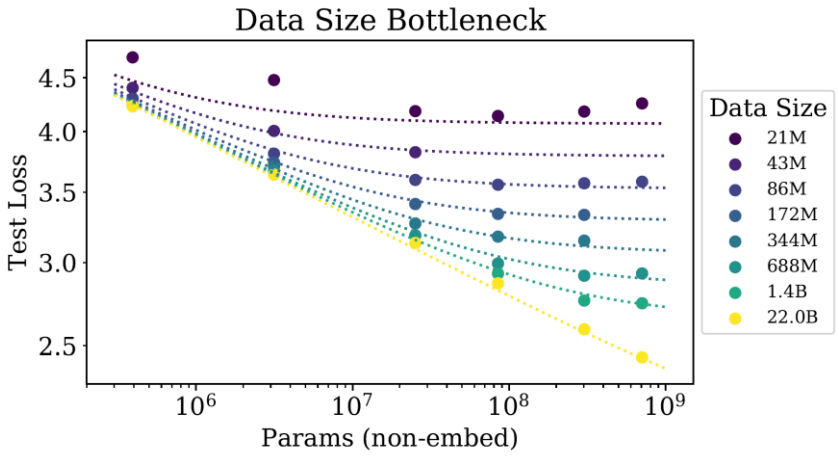
$$L(C) = \left(\frac{C_c}{C}\right)^{\alpha_C}, \alpha_C \sim 0.050, N_C \sim 3.1 \times 10^{8}$$

N、D、C分别代表参数规模、训练数据量、算力

KM scaling law

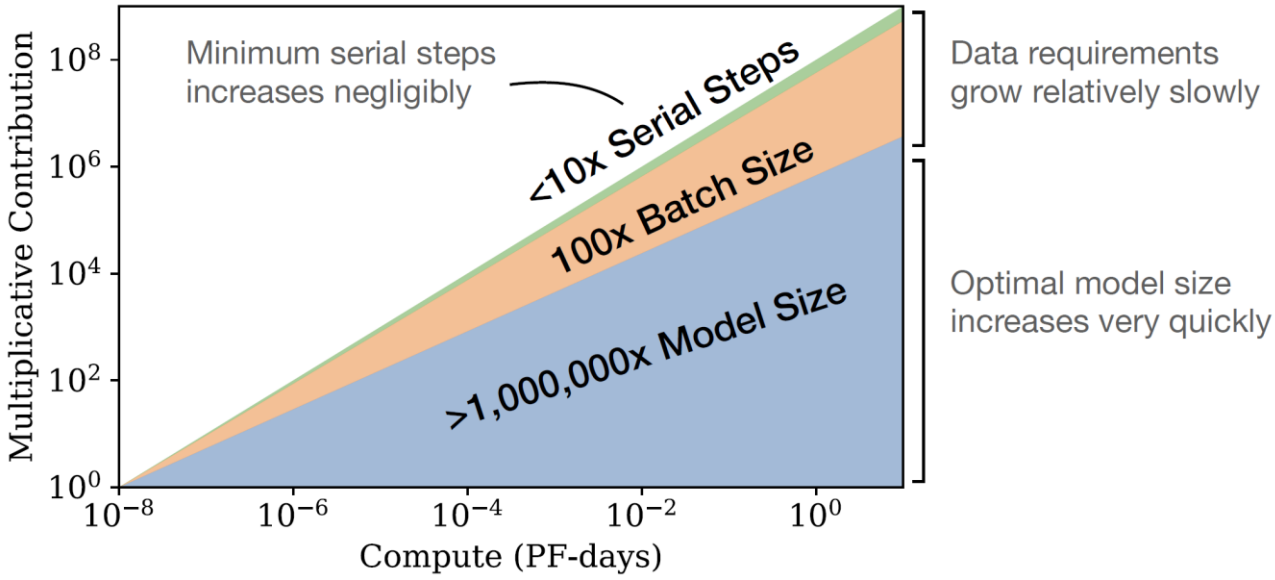
- 同时扩大参数规模(N)和数据规模(D)，性能持续提升
- 如果固定数规模和数据规模中的一个，而增加另一个，性能改进会逐渐消失，出现过拟合
- 经验显示，性能提升依赖参数规模和数据规模的比列关系($N^{0.74}/D$)，意味着每当模型参数规模扩大8倍，数据量需要相应扩大5倍





KM scaling law

- 相比于小模型，大模型更具样本效度(sample efficiency)，达到同样的性能，需要更少的训练数据和训练步数(step)
- 当算力固定，而参数规模和数据不受限制时，获得最优性能的方法是训练大模型，无需等待模型收敛。算力增加的最优选择是训练大模型，并适度增加数据规模。



主要内容

- 模型规模和模型性能
- **提示范式**
- 提示和提示工程
- 涌现能力
- 指令微调
- 对齐微调
- 参数高效微调方法

GPT与Zero-shot learning

- 除了预训练+微调的建模范式，随着模型规模的扩大，GPT等模型展现出0样本行为和少样本行为
- **0样本行为** (zero-shot learning)

  不再利用目标任务标注数据进行参数微调，直接利用预训练模型完成目标任务
- 需要解决目标任务和预训练任务不匹配的问题
- GPT预训练任务 --- 语言生成(language generation)

  根据给定的语境生成文本

  context → completion

GPT与Zero-shot learning

- 将目标任务转写为条件序列生成任务 input→output
  $P(output|input)$
- 通过计算模型生成序列的概率确定结果
  - per-token log likelihood
- 例如：
  - 情感分析--序列分类
    输入input sequence + very ___?
    比对模型生成positive和negative的概率大小

    I really like this movie. very *positive*

GPT与Zero-shot learning

- CoLA任务(Linguistic Acceptability)
  计算例子概率，通过设定阈值确定结果
  The book was written by John. (√)
  Books were sent to each other by the students. (×)
- QA任务：计算给定document及question前提下生成每个answer的概率，取其大者作为正确选项。
- 指代消解：将候选先行词分别替换代词生成不同的句子，分别计算概率，取其大者为正确结果。
  the trophy doesn't fit in the brown suitcase because ***it*** *is too big*. (the trophy, the suitcase)

GPT与task conditioning

- GPT2继续探究了0样本学习能力，展示出GPT模型在不微调的前提下具有一定多任务处理能力。
- 例如
  - 阅读理解类任务**CoQA:**

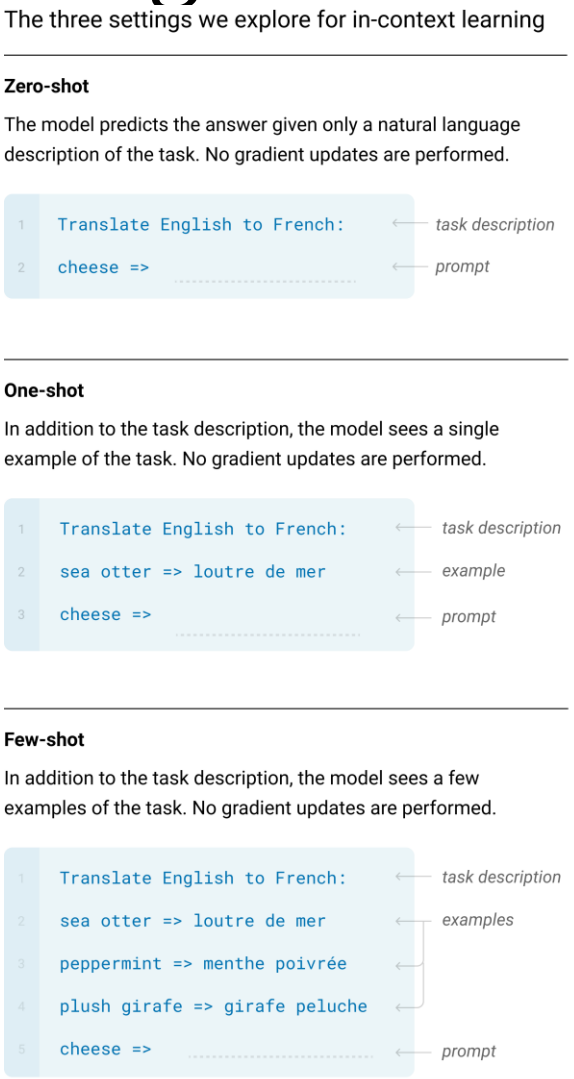
    document, *conversation history*, **A: _____**
  - 摘要任务

    document, **TL;DR: ____**
  - 翻译任务

    english sentence = French sentence english sentence = _____

GPT与few-shot learning

- GPT 3验证了预训练语言模型具备few-shot学习能力。
- 在模型输入中给定若干示例，模型解决任务的能力得到提升。
- 对模型输入的设计
  - task description
  - demonstration
  - prompt
- 提出了in-context-learning的概念
  无需参数微调，模型从context中所给的示例中学习的能力



GPT

- 随着模型规模的扩大，模型展现出
  - zero-shot/one-shot/few-shot learning能力
  - 多任务处理能力
- 不再需要针对特定任务标注数据进行模型微调
- 模型中不再需要面向特定任务进行架构设计
- 预训练语言模型成为通用语言任务模型(AGI)
- 几乎所有任务都转写为条件序列生成任务

$$contexts \rightarrow completion$$

$$prompt \rightarrow response$$

预训练+提示范式

- 自然语言处理的新范式　预训练+提示

- 提示(prompt)
  对语言模型需要完成任务的描述

- 语言模型理解提示，并按照提示的要求完成任务，生成任务输出(response)

主要内容

- 模型规模和模型性能
- 提示范式
- **提示和提示工程**
- 涌现能力
- 指令微调
- 对齐微调
- 参数高效微调方法

提示类型

- 自回归型预训练语言模型 **--- prefix prompt**
  Translate English to French: *cheese* => _____
  document, TL;DR: _____
  …
- 自编码式预训练语言模型 **--- cloze prompt**
  I like the Disney films very much. It was _____.
  下面是_____新闻。*中国女排再夺冠！*
- 输出标签的语言化(verbalize)
  positive → great, fantastic
  negative → terrible, bad
  I like the Disney films very much. It was *great*

提示工程

- 提示工程(prompt engineering)
  设计和优化提示使语言模型更好地完成目标任务的过程
- 原始任务描述： $x \to y$
- 提示设计

$$x \to x' = f_{\text{prompt}}(x)$$

- 输出设计

$$y \to y' = f_{\text{verbalize}}(y)$$

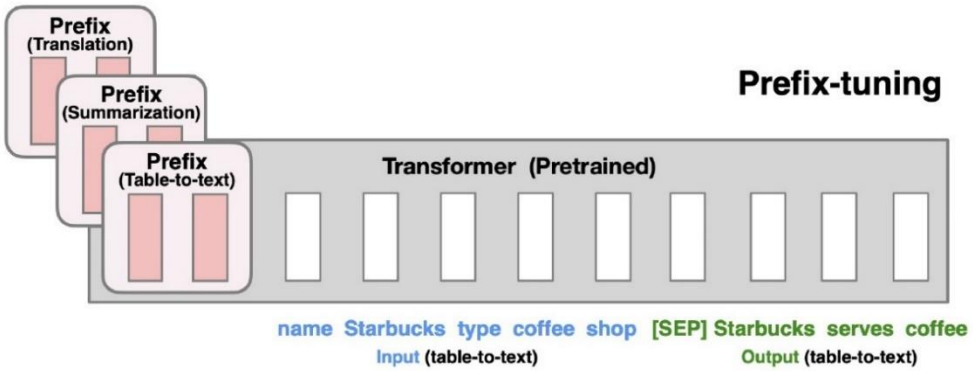
- 提示工程就是寻求最佳$f_{\text{prompt}}(x)$和$f_{\text{verbalize}}(y)$的过程

提示工程

- 人工设计
  - 专家基于经验设计，通常不是最优选择
- 自动构建
  - 设计算法，寻求更好的提示方案
    - 在可能的提示空间中搜索表现最佳的提示
  - 需要基于标注数据进行评价或者训练
  - 离散型提示(硬提示--- hard prompt)
    - 自然语言形式，可理解和解释
  - 连续型提示(软提示--- soft prompt)
    - 直接学习参数化的提示、向量形式的提示

提示工程---prefix tuning

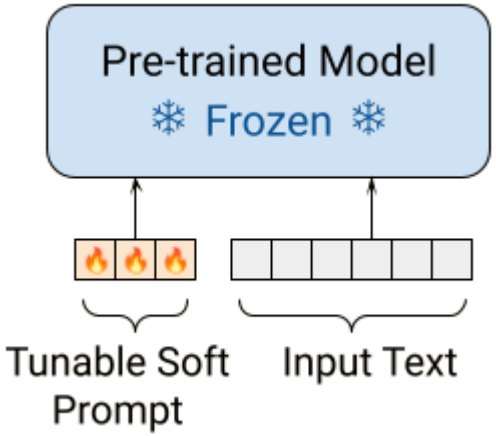
- 模型前增加若干个前缀向量作为前缀
- 针对具体任务训练前缀向量，预训练模型参数保持不变



- 针对不同任务寻求目标任务上的最佳前缀向量，解码时，不同的目标任务使用不同的前缀向量
- 前缀向量被作为各层激活值之前，预训练模型以注意力机制将前缀向量纳入预训练语言模型
- 预训练模型可以是decoder架构(GPT)或者encoder-decoder架构

提示工程---prompt tuning

- 与prefix tuning类似，但只在输入层学习提示向量
- 提示向量与常规文本词向量以同样方式输入预训练模型
- 训练时，针对不同目标任务，固定预训练模型参数不变，只学习提示向量
- 不同目标任务使用不同的提示向量
- 对同一任务可以学习多组提示向量，以集成方式(ensemble)工作



提示的构成

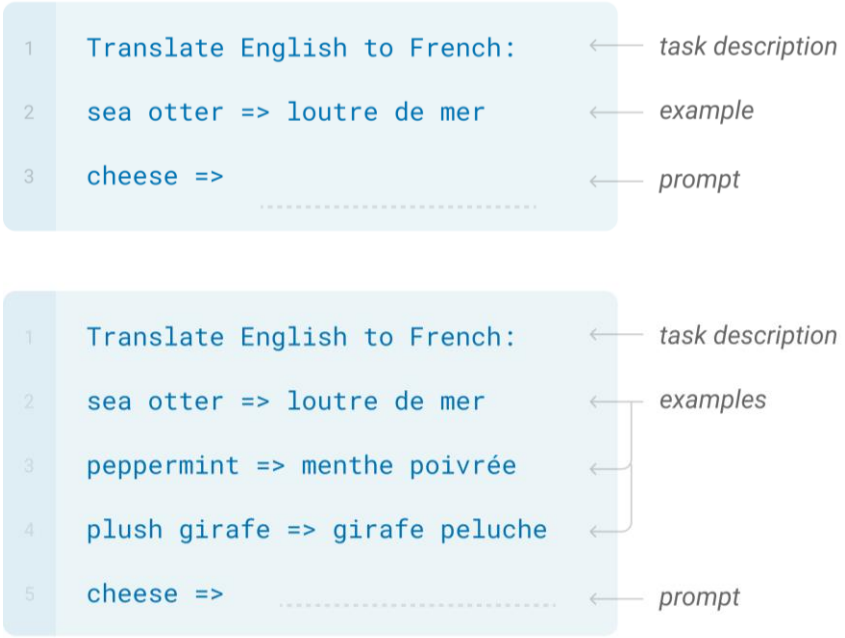
- 存在不同的提示形式，为了适应预训练语言模型的训练目标，提示通常为自然语言形式
- 提示通常需要体现如下要素
  - 任务描述(task description) T
  - 0~k个任务示例(demonstrations) (x_i, y_i)

    示例中可以增加任务求解的中间过程
  - 任务输入 x

$$T, [x_1 \rightarrow y_1, x_2 \rightarrow y_2, \cdots, x_k \rightarrow y_k], x \Rightarrow \hat{y}$$

上下文学习(In context learning)

- 研究发现，在提示中纳入一个或少量任务示例，有助于模型性能的提升，模型展示出从输入的context中学习的能力，被称作In context learning



- 选择几个示例？
- 选择哪些示例？
- 示例正确性是否重要？
- 如何安排顺序？
- 内部工作机理？

思维链(Chain of Thought)

- 研究发现，在许多任务中，通过在提示中纳入 诱导模型输出求解中间过程的要求 或 在提示的示例中纳入求解中间过程，有助于模型性能的提升，被称作chain-of-thought
- 人们认为，思维链技术有助于模型应对复杂推理型任务

$$T, [x_1 \rightarrow i_1 \rightarrow y_1, \cdots, x_k \rightarrow i_k \rightarrow y_k], x \Rightarrow \hat{i} \rightarrow \hat{y}$$

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?
A: ***Let's think step by step.***

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

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?
A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

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

(Output) The juggler can juggle 16 balls. Half of the balls are golf balls. So there are 16 / 2 = 8 golf balls. Half of the golf balls are blue. So there are 8 / 2 = 4 blue golf balls. The answer is 4. ✓

主要内容

- 模型规模和模型性能
- 提示范式
- 提示和提示工程
- **涌现能力**
- 指令微调
- 对齐微调
- 参数高效微调方法

涌现能力(emergent ability)

- 按照scaling law，模型性能是模型规模的函数
  - 扩大模型规模 ⇒ 更好的模型性能(Loss)
  - 性能的持续改善可以预估(幂律关系)
- 但对很多任务而言，随着模型规模的扩大，模型性能一直维持在随机水平(采用具体任务的评价指标)
- Jason Wei等人研究模型规模和任务性能之间的关系，认为在大模型中具有能力涌现现象

Jason Wei et al. Emergent Abilities of Large Language Models.2022

涌现能力

- 涌现能力一般定义

  Emergence is when quantitative changes in a system result in qualitative changes in behavior.

  系统量变引起系统行为质变现象。

- 大模型中的能力涌现
  - 小规模模型中没有的能力
  - 在模型扩大到一定规模时突然具有的能力
  - 涌现能力无法通过小模型能力进行推测

涌现能力

- 模型规模的衡量

  (1) 模型参数数量

  (2) 训练模型所投入的算力大小(training FLOPs)

  (3) 训练模型所使用的数据规模

- Jason Wei等人选择算力大小代表模型规模

  (1) 参数数量通常和算力大小正相关

  (2) 通常基于同样的数据训练不同规模的模型

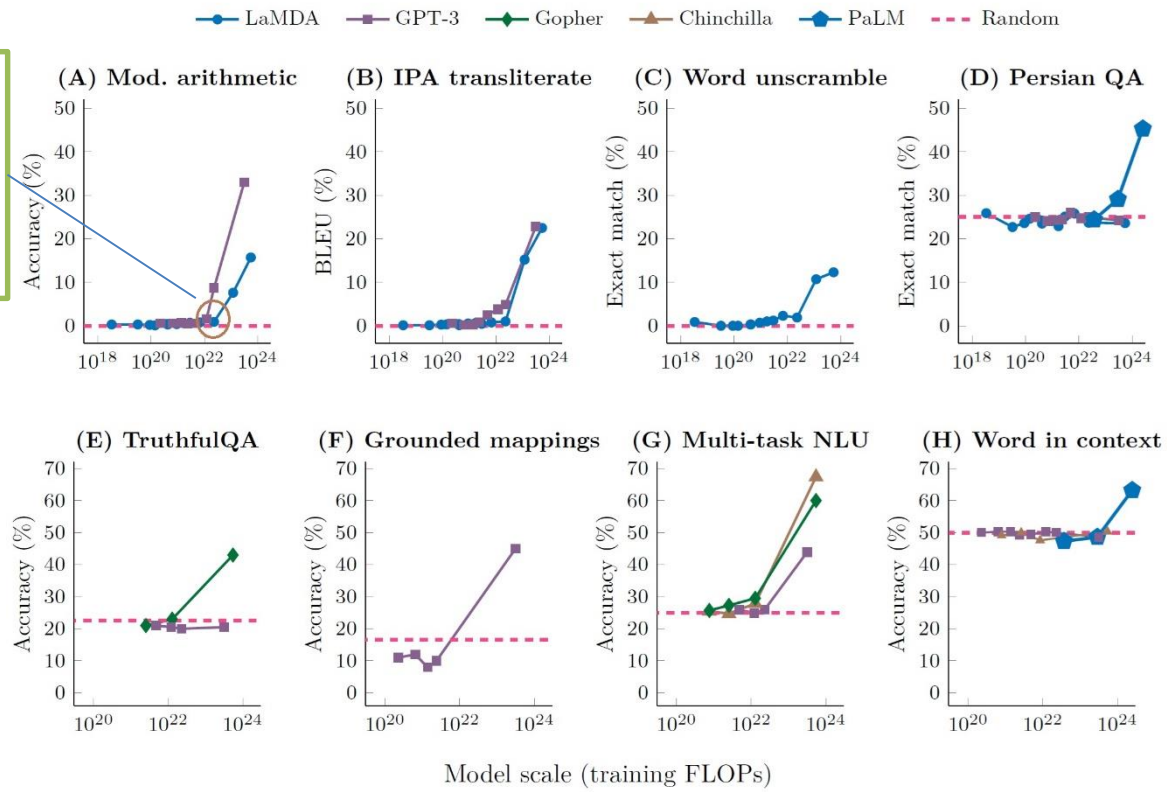
  (3) 基于参数规模和基于算力得到的结果是类似的

- 选择LaMDA、GPT-3、Gopher、Chinchilla、PaLM等不同规模的版本，考察不同任务的性能变化规律

涌现能力

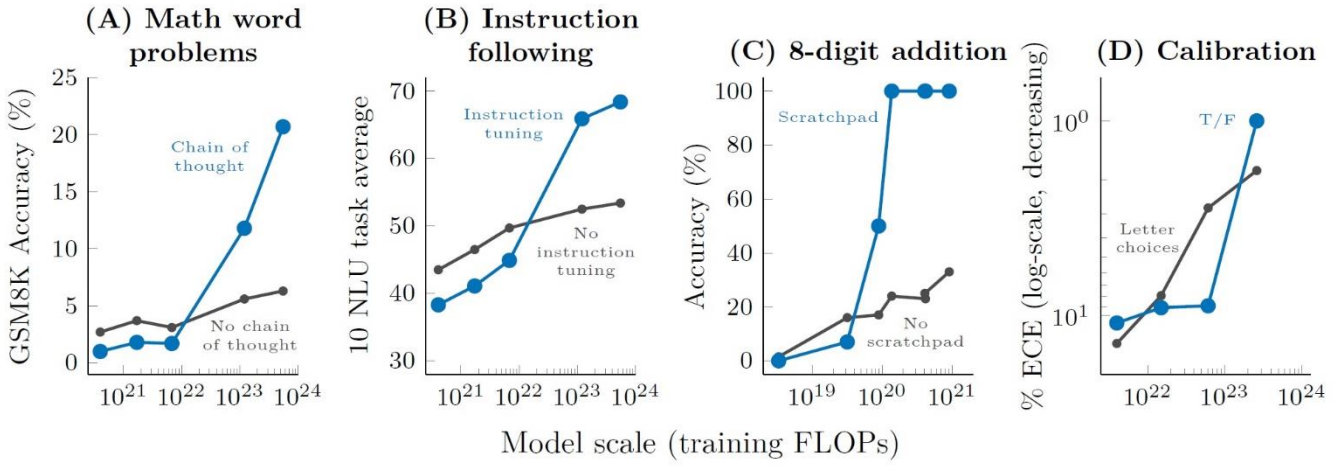
- few-shot prompting(ICL)策略中的涌现现象

$2 \cdot 10^{22}$ FLOPs
13B GPT-3
10^{23} FLOPS
68B LaMDA



涌现能力

- 针对few-shot prompting的一些改进策略也需要在模型达到一定规模时才能奏效
- (1) 思维链 (2) 指令微调 (3) 程序执行 (4) 模型校准



- 在模型未达到一定规模时，这些策略甚至起到负面作用

涌现能力

- 模型能力涌现所要求临界点不是不变的。更好的数据、模型架构，是否可以使能力更早涌现？
- 继续扩大规模，是否会出现更多的涌现能力？
- 能力涌现的同时也会伴随风险的涌现(真实性、偏见、有害的内容)

- 涌现现象的解释需要研究

  多步推理，l步计算，可能需要的网络深度至少是$O(l)$

  记忆支持型任务，更多的参数意味着更好的记忆

- 评价指标和涌现能力

主要内容

- 模型规模和模型性能
- 提示范式
- 提示和提示工程
- 涌现能力
- **指令微调**
- 对齐微调
- 参数高效微调方法

指令微调

- 大规模语言模型的实践表明
  - 大模型有初步(多任务)zero-shot learning能力(如GPT-2)
  - 大模型zero-shot learning能力有限
  - 通过在提示中提供示例(few-shot learning)，提升大模型的性能(如GPT-3)

- zero-shot learning能力为什么有限？
  - 训练和应用时数据形式差异太大
  - 增加任务示例有助于模型理解任务

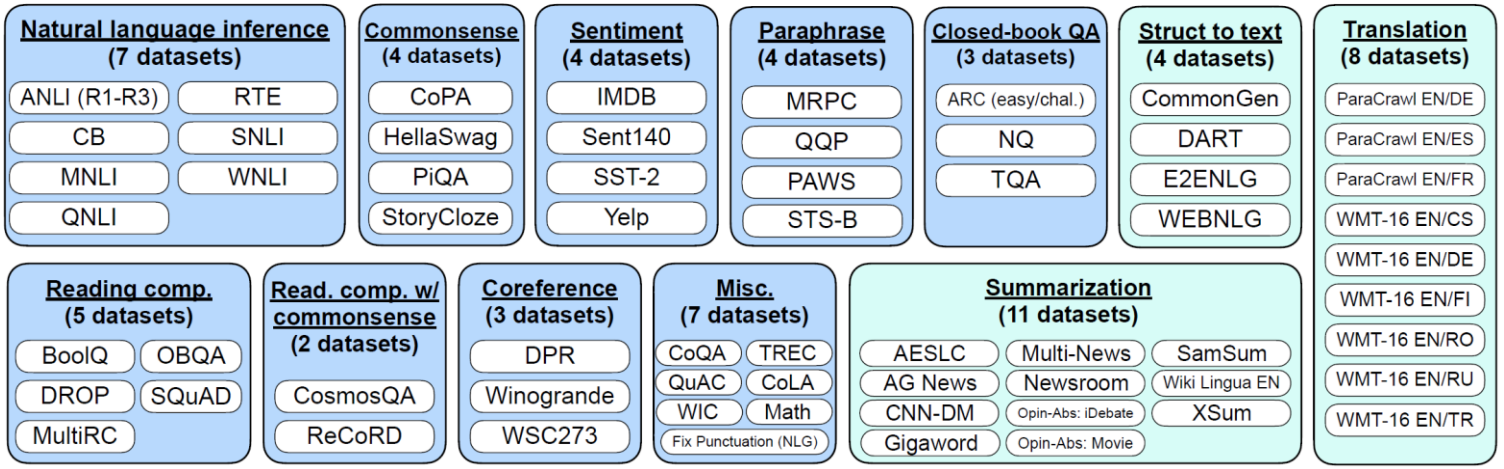
指令微调

Is the sentiment of this movie review positive or negative?
Translate 'how are you' into Chinese.

- 解决策略
  - 用自然语言撰写提示
  - 用自然语言提示及响应数据微调大模型
  - 混合多种任务的数据微调大模型
- 指令(instruction)：用自然语言描述的NLP任务
- 指令微调 --- Instruction Tuning
- 指令微调可以提升zero-shot learning的能力
- 多任务指令微调泛化了指令理解能力
  - 提升了unseen task的处理性能

指令微调

- Jason Wei等选择12类62个数据集，涵盖NLU和NLG任务

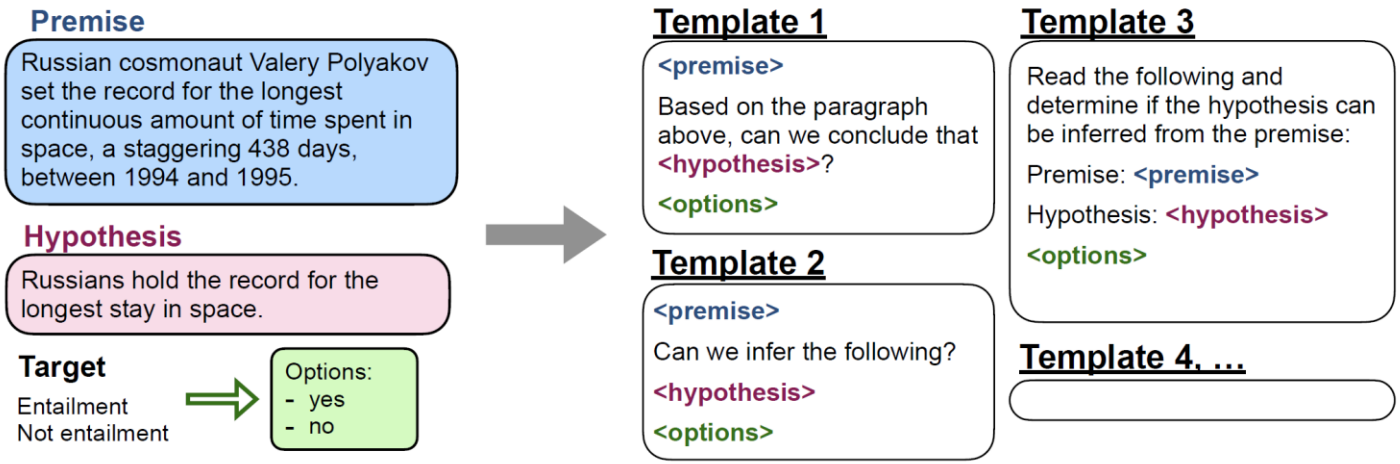


- 任务形态
  - classification (e.g. sentiment classification)
  - free text generation (e.g. summarization)

Jason Wei et al. Fine-tuned language models are zero-shot learners.2022

指令微调

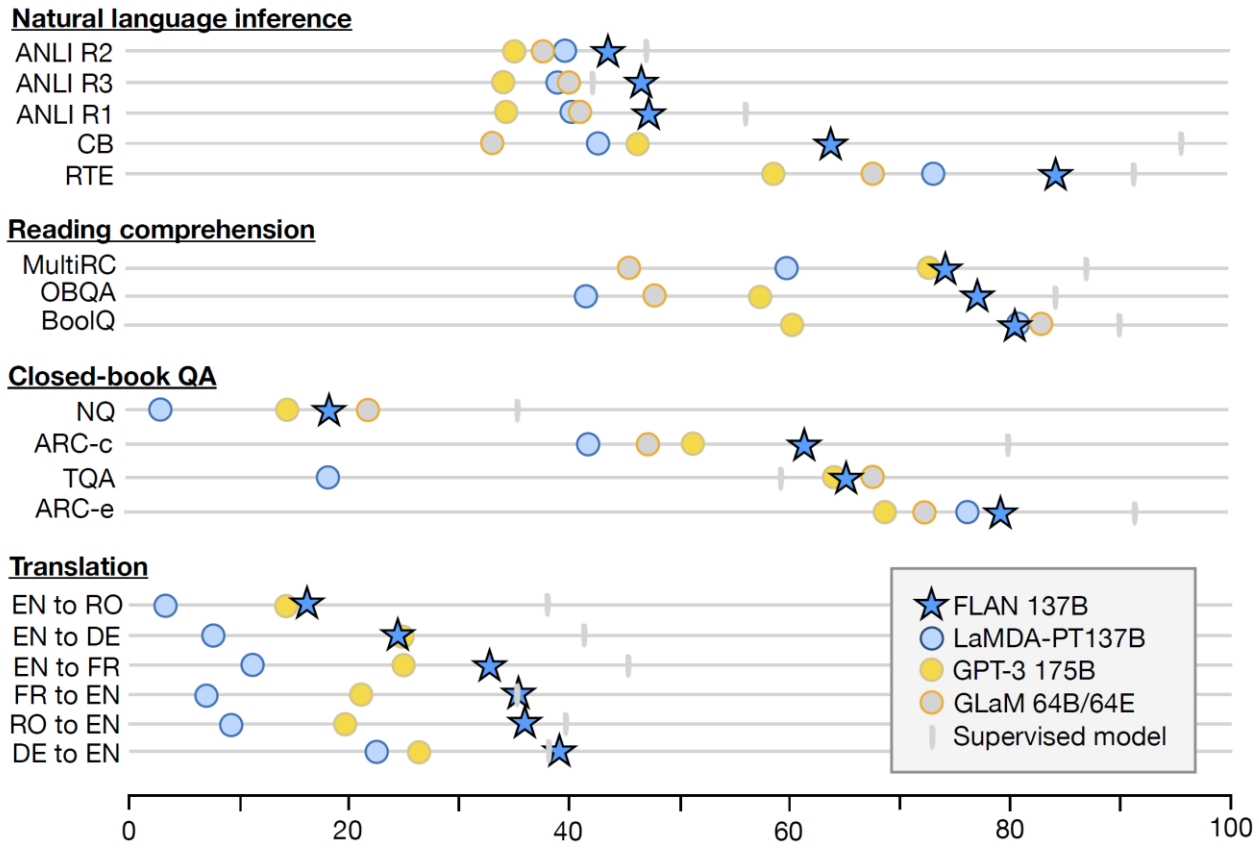
- 用人工定义的模板把每个任务的数据转写为指令形式



- 混合不同任务的指令，微调预训练模型LaMDA-PT
  - LaMDA-PT 137B 只经过预训练
- 微调版本称作FLAN 137B

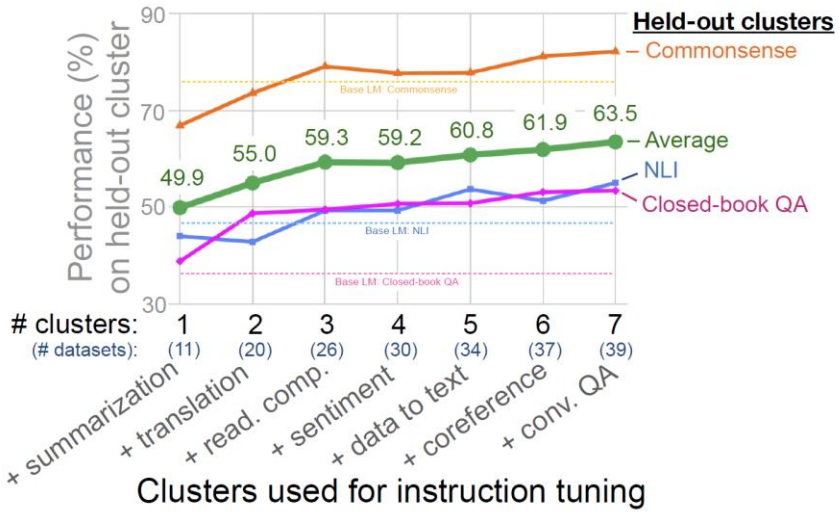
指令微调

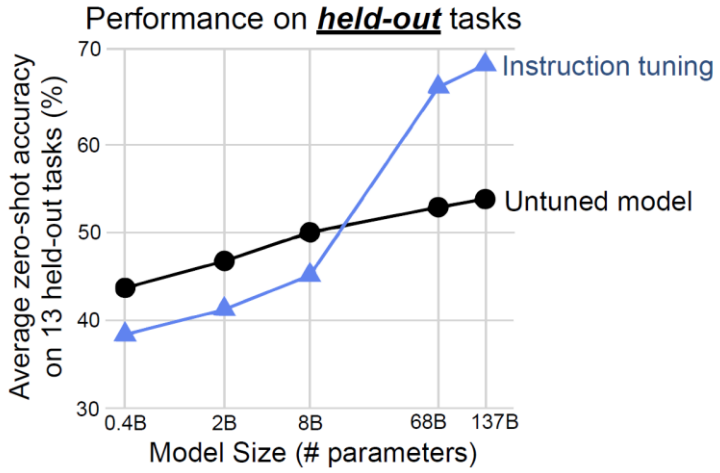
- zero-shot 性能提升



指令微调

- 微调任务越多，在unseen task上性能越好





- Instruction tuning对模型规模有要求，是一种涌现能力
- 使用时加入示例会进一步提升性能(few-shot learning)

主要内容

- 模型规模和模型性能
- 提示范式
- 提示和提示工程
- 涌现能力
- 指令微调
- **对齐微调**
- 参数高效微调方法

对齐微调

- 大模型预训练目标是 — **给定语境，预测下一个词例**
- 用户期望的模型目标 — **以安全有益的方式遵循用户的指令**
- 大模型常见问题
  不遵循指令、虚假信息、社会偏见、毒性内容
- 用户对模型的期望 — 符合人类价值观
  - 有助(helpful)
  - 诚实(honest)
  - 无害(harmless)
- 对齐(alignment): 使模型的行为符合人类期望和人类价值观

对齐微调

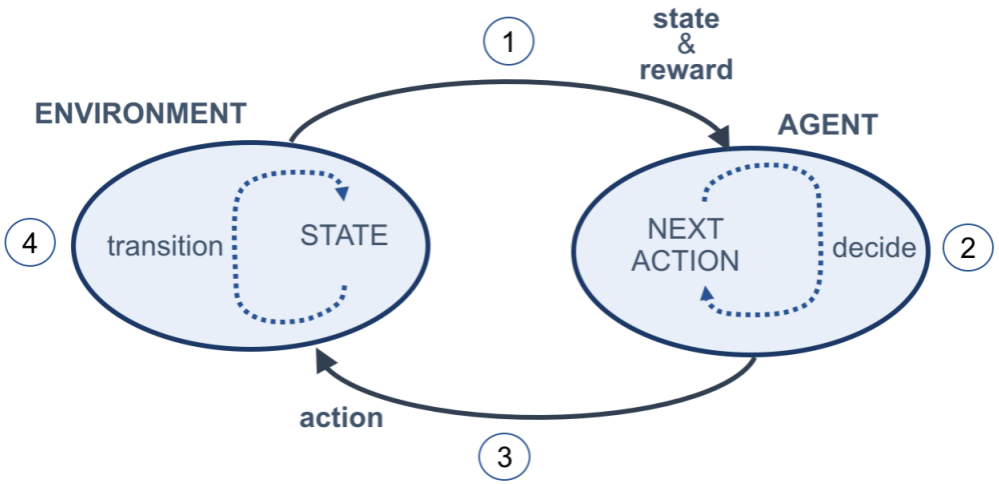
- 困难：人类价值、偏好很难形式化为优化目标
- 解决策略 --- 微调 (fine-tuning)
  - 利用符合人类价值、偏好的数据微调模型
  - 基于人工标注数据
  - 基于人工对模型数据的评价反馈

- 对齐微调(alignment tuning)
- 对齐税 (alignment tax)

  对齐微调可能造成模型在一些处理任务上的性能下降

强化学习简介

- 强化学习：智能体通过与环境的交互进行学习
- 智能体通过执行动作改变环境的状态。
- 环境作为回应，会回馈智能体奖励。



- 强化学习是一种目标导向的学习，智能体目的是在交互过程中最大化交互过程中的总奖励。为此，智能体需要通过探索试错，学习出指导动作选择的最佳策略。
- 策略(policy): 给定状态s，智能体所能进行的动作的概率分布。即$\pi(a|s)$

对齐微调

- Long Ouyang等采用RLHF的方式对GPT-3进行对齐微调，微调后的模型称为InstructGPT.
- RLHF(Reinforcement Learning from Human Feedback)
- 基于人类反馈用强化学习的方法微调模型

- 关键步骤

  (1) 利用人工标注提示数据微调GPT-3，得到一个有指导微调版模型(SFT模型)

  (2) 利用人工标注了优劣差异的对比数据集，微调GPT-3模型，得到一个奖励模型(RM模型)

  (3) 将RM模型用作奖励函数，用强化学习的方法微调SFT模型

Long Ouyang et al. Training language models to follow instructions with human feedback.2022

对齐微调

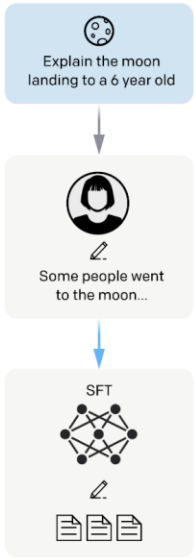
Step 1

Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3 with supervised learning.



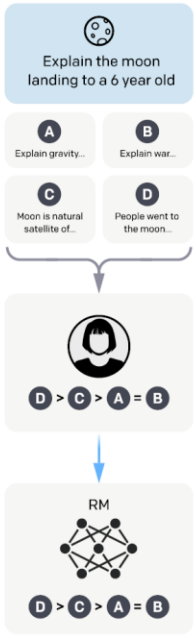
Step 2

Collect comparison data, and train a reward model.

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.



Step 3

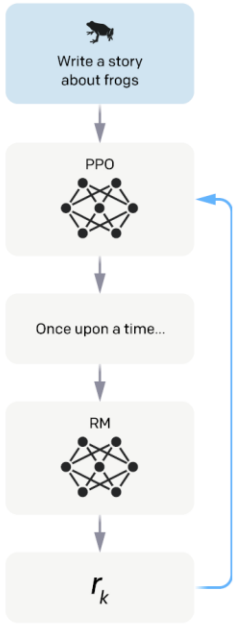
Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.

The policy generates an output.

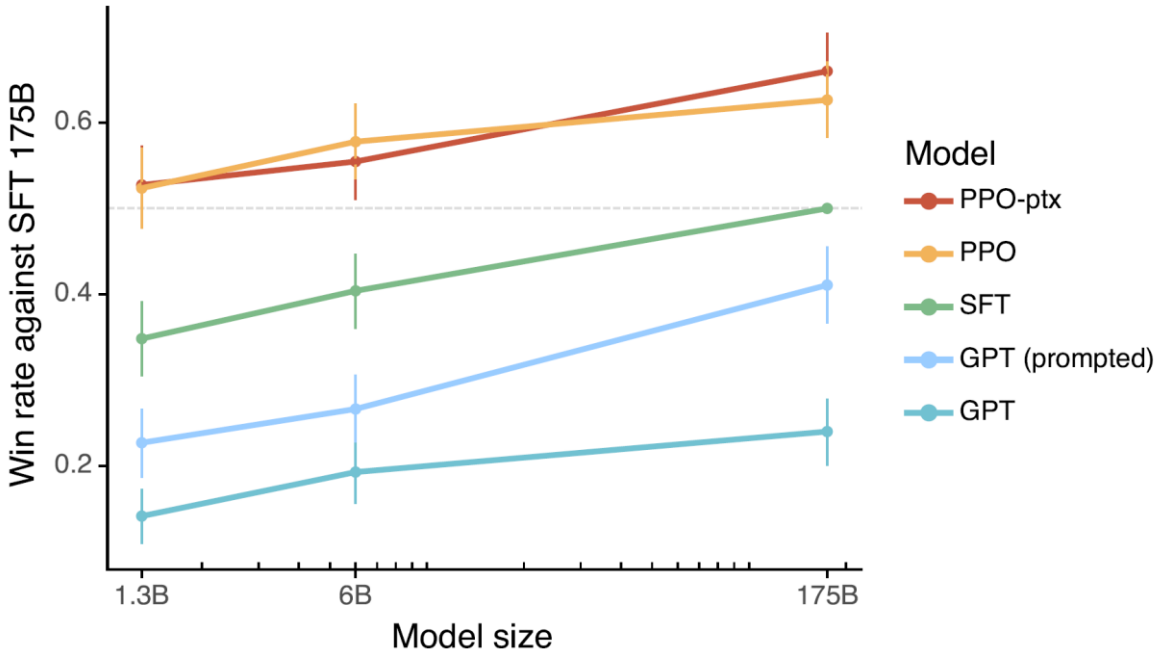
The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.



对齐微调

- 训练了多个不同规模的版本(1.3B/6B/175B)
- 在测试集上人工评价结果是否符合人类偏好



对齐微调

- 在真实性(truthfulness)方面, 和GPT-3相比，InstructGPT有明显改善
- 在有毒内容(toxicity)方面，和GPT-3相比，InstructGPT有改善(有害内容少了25%)
- 在偏见(bias)方面，InstructGPT没有显著改善
- 改进优化目标，可以有效降低alignment tax
- 模型对齐的是标注人员(labeler)的偏好
- 但实验显示对齐也泛化到held-out labeler的偏好
- ChatGPT与InstructGPT采用相同的对齐微调方法
- 人们还在研究其他微调对齐方法

主要内容

- 模型规模和模型性能
- 提示范式
- 提示和提示工程
- 涌现能力
- 指令微调
- 对齐微调
- **参数高效微调方法**

参数高效微调方法--LoRA

- 随着模型规模的持续增加，微调全部参数代价很大
- 基本原理
  - 初始模型参数为W_0, 微调后模型参数为W
  - 微调过程实现了$W \leftarrow W_0 + \Delta W$
  - LoRA的做法是固定W_0，通过微调得到ΔW
  - 对ΔW再参数化，将其进行低秩分解

$$\Delta W = BA$$

  - 将微调全部模型参数转换为求解参数A和B
- 需要训练的模型参数数量大大减少

$$W \in \mathbb{R}^{m\times n} \Rightarrow A \in \mathbb{R}^{k\times n}, B \in \mathbb{R}^{m\times k}, k \ll \min(m,n)$$

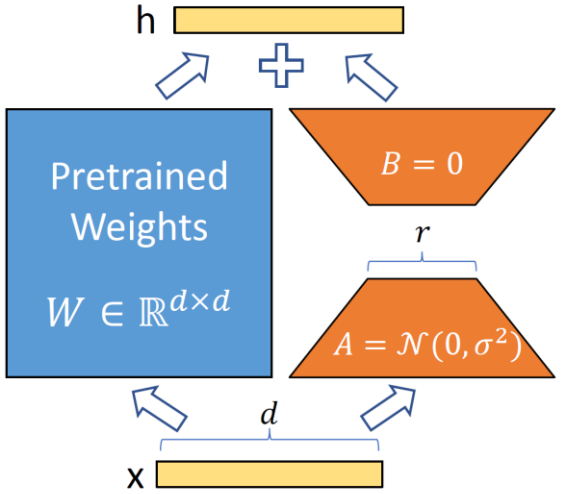
Edward Hu et al. LoRA: Low-Rank Adaptation of Large Language Models, 2022

参数高效微调方法--LoRA

- 训练过程中的前向计算

$$h = (W_0 + \Delta W)x$$
$$= W_0 x + \Delta W x$$
$$= W_0 x + BAx$$

- 训练过程中，对W_0不做梯度更新只对B和A做梯度更新
- 对矩阵A做随机初始化(Gaussian)矩阵B初始化为0



- 利用大模型参数内在低秩特征，增加旁路矩阵模拟全参数微调

参数高效微调方法--LoRA

- 全参数微调方法优化目标

$$\max_{\Phi} \sum_{(x,y)\in Z} \sum_{t=1}^{|y|} \log(P_{\Phi}(y_t|x, y_{<t})$$

- LoRA微调优化目标

$$\max_{\Theta} \sum_{(x,y)\in Z} \sum_{t=1}^{|y|} \log(P_{\Phi_0+\Delta\Phi(\Theta)}(y_t|x, y_{<t})$$

- 实践中，可以选择对哪些参数矩阵应用LoRA微调
  例如，transformer架构中$W_q, W_k, W_v, W_o \ldots$

参数高效微调方法--LoRA

- 极大缩减微调参数数量
- 极大减少了内存需求
- 可以提升微调的效率
- 部署时，将参数合并，不会增加解码时间

- 大模型微调可以应用LoRA，例如：Alpaca-Lora