GPT-3/ChatGPT

Jingfeng Yang
Applied Scientist, Amazon

Outline

- Why did all of the public reproduction of GPT-3 fail?
- In which tasks should we use/not use GPT-3.5/ChatGPT?
- Some suggestions on using GPT3(.5)
- Some mysteries remaining with LLMs
- Opportunities of research labs

Why did all of the public reproduction of GPT-3 fail?

- Compare GPT-3-175B/PaLM-540B with OPT-175B/BLOOM-176B

- Pretraining Data
  - Data Quality
  - Data Deduplication
  - Data Diversity

- Training Strategy
  - Training Framework
  - Modifications During Training Procedure
  - Model Architecture/Training Setup
  - Training Duration

Pretraining Data – Data Quality

- GPT-3/PaLM
  - A well-performed classifier to filter out the low-quality data
- OPT/BLOOM
  - No such classifier
- Conclusion: Pretraining a model with less high-quality data could outperforms those with much more mixed-quality data.

Pretraining Data – Data Deduplication

- GPT-3/PaLM
  - Document-level Deduplication

- OPT
  - After document-level deduplication, there are still many duplications in the Pile pretraining corpus.

- Conclusion: Data duplication makes a pretrained model see the same data point for many times and overfit/memorize them, leading to its worse generalization. (Note: some recent papers are against this point.)

Pretraining Data – Data Diversity

- GPT-3
  - More proportion of diverse data from Common Crawl

- BLOOM
  - ROOTS pretraining corpus used by BLOOM has too many existing academic datasets, leading to its lower diversity (in terms of distribution).

- Task-specific performance also depends on pretraining data composition:
  - More multilingual data used in BLOOM and PaLM -> better performance on multilingual tasks and Machine Translation.
  - Large portion of social media conversations used in PaLM -> better performance on QA
  - Code data in PaLM/GPT-3.5 -> better coding and CoT ability (Interesting observation: Poor CoT ability of BLOOM despite its code pretraining data)

Training Strategy – Training Framework

- PaLM
  - TPU based Pathways System (Data Parallel and model Parallel): bfloat16/BF16 (bfloat16 can represent larger ranges of floating numbers than float16/FP16, handling large values during training loss spikes)
- OPT
  - FSDP implementation of ZeRO (Distributed Optimizer)
  - Megatron-LM implementation of model parallel
  - Used float16, which is the only option for mixed-precision training in V100, thus widely being used
- BLOOM
  - Deepspeed implementation of ZeRO
  - Megatron-LM implementation of model parallel
  - Used bfloat16, but additional use of layer normalization after the first embedding leads to is poor zero-shot generalization.

Training Strategy – Modifications During Training

- PaLM
  - No any midflight adjustment
  - Restarted training from a checkpoint roughly 100 steps before the spike started, and skipped roughly 200–500 data batches. This could succeed because its Bitwise determinism, and modifications to the models architecture and training setup for better stability.
- OPT
  - OPT did a lot of midflight adjustment after loss spike
  - Changing clip gradient norm and learning rate
  - Switching to vanilla SGD and then back to Adam.
  - Resetting the dynamic loss scalar
  - Switching to a newer version of Megatron

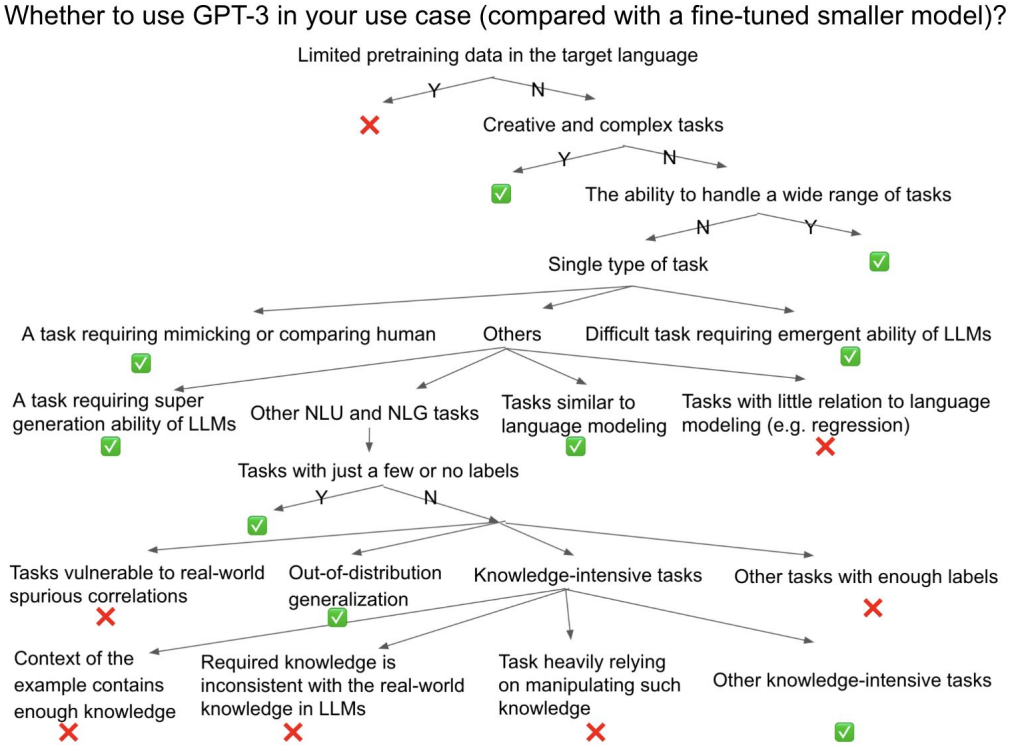
Training Strategy – Model Architecture/Training Setup

- OPT/BLOOM:
  - No much specialized optimization for better stability
- PaLM:
  - A modified version of Adafactor as the optimizer
  - Scaling pre-softmax output logits
  - Using an auxiliary loss to encourage the softmax normalizer to be close to 0
  - Using different initialization for kernel weights and embeddings
  - Using no bias in dense kernel and lay norms
  - Using no dropout during pretraing
- GLM-130B
  - DeepNorm based Post-LN instead of Pre-LN
  - Embedding Layer Gradient Shrink

Training Strategy – Training Duration

	Pretraining Corpus Tokens	Tokens Seen During Pretraining
PaLM	780B	770B
GPT3	500B	300B
OPT	180B	300B
BLOOM	341B	366B

In which tasks should we use/not use GPT-3.5/ChatGPT?



Comparing prompting best performed LLMs (e.g. FLAN-PaLM, InstructGPT) with finetuned smaller SOTA (e.g. FLAN-T5)

Use Cases - Creative and complex tasks

- Tasks which are difficult/“impossible to label”
  - Coding (code completion, natural language instructions to code, code translation, bug fixing):
    - Despite Codex/PaLM’s overall much better performance then previous models, we should still allow LLMs to sample many (k) times to pass the test case (use pass@k as metric).
  - Summarization: previous human labeled summarization has been surpassed by LLM generated summarization (Zhang et al, 2022.)
  - Translation: prompting PaLM 540B could surpass fine-tuned models in some machine translation tasks: translating from low/mid resource languages to English
    - Reason: English data typically accounts for a large portion in pretraining corpus, and thus LLMs are good at generating English sentences
  - Creative writing (e.g. writing stories, articles, emails, reports, writing improvement etc.)

Use Cases - Tasks with just a few or no labels

- GPT3 is designed for those tasks which are “expensive for labeling”
- Finetuning a smaller model is typically impossible to match GPT3’s zero/one/few-shot performance.

Use Cases - Out-of-distribution (OOD) generalization

- Traditional finetuning might overfit training set and have poor OOD generalizability, while few-shot in-context learning could have better out-of-distribution generalizability
- Example 1: Prompting PaLM could outperform finetuned SOTA on Adversarial Natural Language Inference (ANLI), while it still underperforms finetuned SOTA on normal NLI (RTE dataset).
- Example 2: Prompting LLMs has shown better compositionally OOD generalization than finetuned models
- Possible Reason: 1) No need to update parameters during in-context-learning, avoiding overfitting. 2) Those previous OOD examples are in-distribution examples for LLMs.

Use Cases - The ability to handle a wide range of tasks

- Handling various tasks instead of focusing on superior performance on a specific task
- Example 1: Chatbot - GPT3
- Example 2: Reasonable performance of GPT3(.5)/PaLM on various benchmarks

Use Cases - Knowledge-intensive tasks

- Knowledge-intensive tasks where retrieval is not feasible
- Use world knowledge stored in LLMs
- Example: Closed-book Question Answering, MMLU (a benchmark dataset including multiple-choice questions from 57 subjects across STEM, the humanities, the social sciences, and more, which tests LLMs' world knowledge and problem solving ability)
    - On MMLU benchmark, PaLM-540B has better performance than any finetuned models
- Instruction-tuning a smaller model can also achieve similar effect to scaling
- However, if a retrieval step could be added for retrieval augmented generation, this use case is questionable.
    - Atlas is better than both PaLM and the most recent InstructGPT in both closed-book NaturalQuestions and TrivialQA datasets

Use Cases - Using emergent abilities of LLMs for difficult tasks

- Emergent ability:
  - Discontinuous improvements from 8B to 62B to 540B models, which exceeds the scaling law.
- Example 1: BIG-Bench (including logical reasoning, translation, question answering, mathematics tasks)
  - PaLM-540B also outperforms the average human performance on aggregate.
- Example 2: PaLM has shown that, among 7 multi-step reasoning tasks, including arithmetic and commonsense reasoning, 8-shot CoT is better than finetuned SOTA on 4 tasks, and could nearly match fintuned-SOTA in remaining 3 tasks. (scaling and CoT are both important).

Use Cases - AGI with human-level performance/mimicking humans

- “Humans do not require large supervised datasets to learn most language tasks. Humans could seamlessly mix together or switch between many tasks and skills through at most a few examples. Thus traditional finetuning methods lead to an unfair comparison with humans, despite their claimed human-level performance in many benchmark datasets.”

- Example: ChatBot – ChatGPT

Use Cases – NLP tasks similar to language modeling

- Example 1: Ending Sentence or Ending Word Cloze – LAMBADA
- Example 2: Anaphora Resolution – Winogrande

Use Cases – NLP tasks similar to language modeling

- Example 1: Ending Sentence or Ending Word Cloze – LAMBADA
- Example 2: Anaphora Resolution – Winogrande

No Use Cases – Issues of Calling OpenAI GPT3 API or Hosting Similar-size Models

- Calling OpenAI GPT3 API is out of the budget (e.g. for new startups without much money)
- Calling OpenAI GPT3 API could raise some security issues (e.g. data leakage to OpenAI, or potentially generated harmful contents).
- No enough engineering or hardware resources to host a similar-size model, as well as eliminate the inference latency issue.
    - Using Alpa naively to host OPT-175B on 16 40G A100 GPUs requires 10 seconds to finish one-instance inference

No Use Cases – Some NLU tasks which neither require knowledge nor require generation abilities of LLMs

- Test data points are mostly in the same distribution as training data points in hand. Such tasks are already performed well enough by previous finetuned models.
- Suggestion: Finetuning FLAN-T5-11B first, instead of prompting GPT3
- Example 1: SuperGLUE, a difficult NLU benchmark, including reading comprehension, textual entailment, word sense disambiguation, coreference resolution, and causal reasoning tasks, all of the PaLM (540B)'s few-shot prompting results are still worse than a finetuned T5-11B, with significant gaps in most tasks.
- Example 2: In sentiment analysis, ChatGPT is still worse than finetuned smaller models according to a recent paper.

No Use Cases – Do not require much extra knowledge from LLMs

- In machine reading comprehension tasks, there is still a large gap between the best few-shot prompting LLMs and finetuned SOTA. In most datasets, the gaps are extremely large.
- That’s probably because all the knowledge required to answer a question is already available in the given passage, and do not require extra knowledge in LLMs.

No Use Cases – Unlikely to get enough knowledge, or see similar distribution from LLMs

- Low-resource language where LLMs have limited pretraining data in such language:
  - PaLM and ChatGPT is still worse than MT-finetuned smaller models when translating from English to other languages, and translating high-resource languages to English.
  - There is still a large gap between few-shot PaLM 540B and fine-tuned smaller models for multilingual Question Answering.
  - For multilingual text generation (including summarization and data-to-text generation), there is still a large gap between few-shot PaLM 540B and fine-tuned smaller models.
- Interesting point: even with very limited pretraining data in some languages, LLMs could still show reasonable performance

No Use Cases – Required knowledge is inconsistent with the knowledge in LLMs

- Some difficult tasks which are not grounded to real-world data:
  - Inverse scaling prize: Redefine-math (e.g. a prompt might “Redefine π as 462”). In such cases, the real-world prior in LLM is too strong to be overridden by prompts, while finetuning a smaller model could probably better adapt to counterfactual knowledge.
  - Inverse scaling prize: “modified quote repetition”, where LLMs is asked to repeat a modified famous quote in the prompt. In this case, LLMs is inclined to repeat the original famous quote, instead of the modified version.
  - Many difficult tasks in BigBench for LLMs:
    - In mathematical induction, PaLM makes many errors when assumption in the prompt is incorrect (e.g. “2 is an odd integer”)

No Use Cases – Heavily relying on manipulating required knowledge

- Some tasks that require knowledge from LMs, but also heavily rely on manipulating such knowledge, where such manipulation could not be easily solved by the LLM next-token prediction objective.
- Some commonsense reasoning tasks
  - Hypothesis of CoT and least-to-most prompting: call out those continuous pretraining texts, which happens to mimic the process of planning and decomposing/composing knowledge.
  - CoT and least-to-most prompting perform well in some mathematical reasoning, coding and other simple natural language reasoning tasks, but still struggles in many commonsense reasoning (e.g. deductive reasoning as explained in inverse scaling law) and wired symbolic reasoning tasks, which are not encompassed by most real-world continuous sequences in natural language data, and require manipulating distributed knowledge.

No Use Cases – Tasks vulnerable to spurious correlations in in-context learning examples or real-world data

- Example: Negation Question Answering from the Inverse Scaling Prize.
- If a LLM is asked “If a cat has a body temp that is below average, it isn’t in …”, it is inclined to generate “danger” instead of “safe ranges”.
- Because the LLMs is dominated by the relation between “body temp that is below average” and “danger”, which, instead, is a spurious correlation with a negation.

No Use Cases – Large divergence between objective and language data

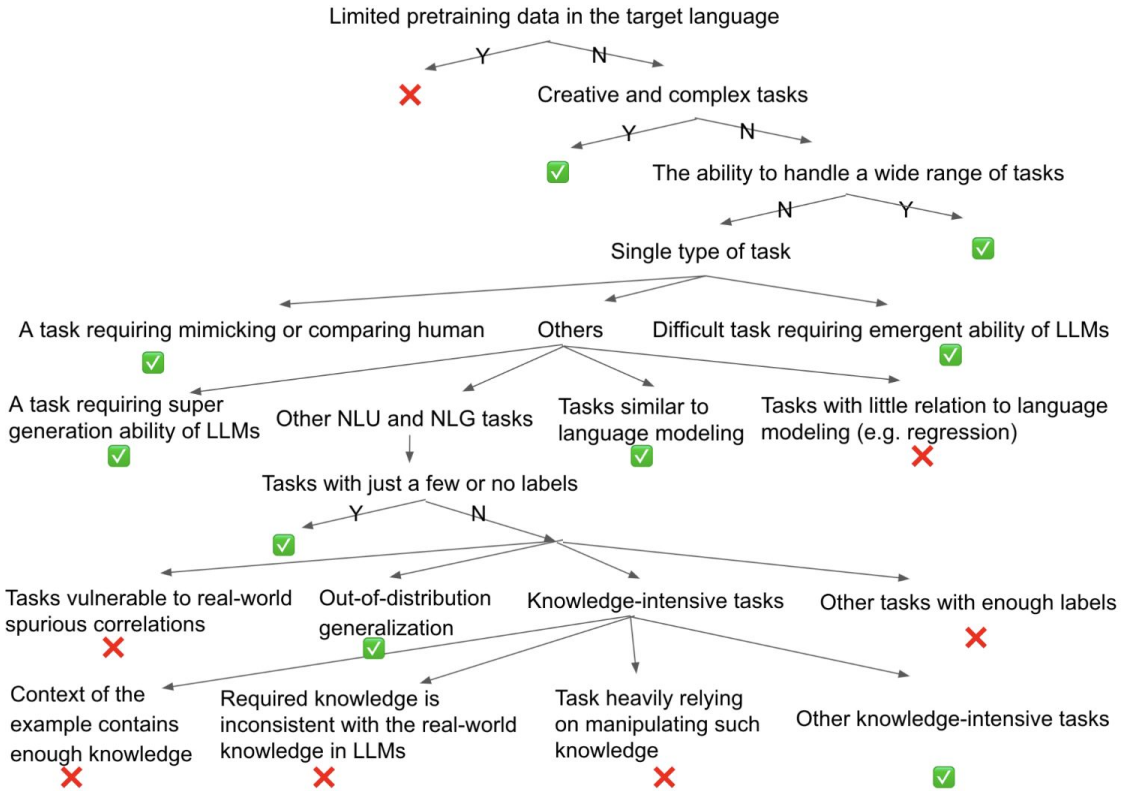
- Regression problems, where finetuned models are difficult to be replaced by LLMs.
- Multimodality tasks could not be completed by LLMs alone, but could potentially benefit from better large pretrained multimodality models.

No Use Cases – Not requiring emergent abilities of LLMs

- Need to better where such emergent abilities come from during LLM training

Summary: use and no-use cases

Whether to use GPT-3 in your use case (compared with a fine-tuned smaller model)?



Some suggestions on using GPT3(.5)

- In some tasks, few-shot demonstrations in in-context learning will not help, even worse than instructions alone, like the Ending Sentence or Ending Word Cloze task.
- Due to inference latency requirement and cost of calling OpenAI APIs
  - Try finetuning FLAN-T5 first in your task before using GPT-3
  - Use LLMs to generate/label data offline
    - Such labeled data could be looked up and provided to users online
    - Such labeled data could be used to finetune a smaller model, which could alleviate the human-annotated data required to train a model, and inject some emergent abilities (e.g. CoT) of LLMs to smaller models.

Some mysteries remaining with LLMs

- Where do emergent abilities of LLMs come from?
  - Why could CoT work?
  - Why could in-context learning work?
  - Why could LLMs perform reasonably on some tasks even with very limited pretraining data in that language?
- A more aggressive question: if a LLM based-AGI could outperform humans in some tasks, how could it learn from “worse” human feedbacks to achieve better alignment?

Opportunities of research labs

- If still have enough resources, explore instruction tuning, and human-in-the-loop learning for better alignment (Interactive Learning / Human-AI Collaboration).
  - A(G)I should help people instead of replacing people
- How to use LLMs (with inference only):
  - Decoding/Sampling strategy.
  - Generating data for augmentation
  - Learning with limited data
  - Problem Definition / Dataset Creation / Evaluation
  - Language Model as KG
  - Prompt designing
  - Fundamentally addressing issues of reasoning/logic/structure
  - Robustness, OOD generalization
  - Long document generation/understanding
  - Deal with specific use cases (e.g. ambiguity in language, mining ChatGPT failures/hard examples)
  - Personalization
  - Security/privacy
  - SocialNLP (bias, toxicity etc.)
  - Complete complex tasks (e.g. ToolFormer)

Thanks!