Session #4: GPT models

Thursday, Sept 8

CSCI 601.771: Self-supervised Statistical Models

JOHNS HOPKINS UNIVERSITY

Motivation

- Previous work pre-trained models have been directly fine-tuned
  - Limitation: despite task-agnostic architecture, need task-specific datasets and fine-tuning
- Issues
  - Large datasets needed for every new task constrains applicability
  - Generalization in the above setting can be poor
  - Humans can generally perform a new language task from only a few examples or from simple instructions – something which current NLP systems still largely struggle to do
- As a result, we would want a universal model trained for diverse skills which might contain many parameters to adapt to different tasks.
- GPT-3, an autoregressive language model with 175 billion parameters, 10x more than any previous non-sparse language model.

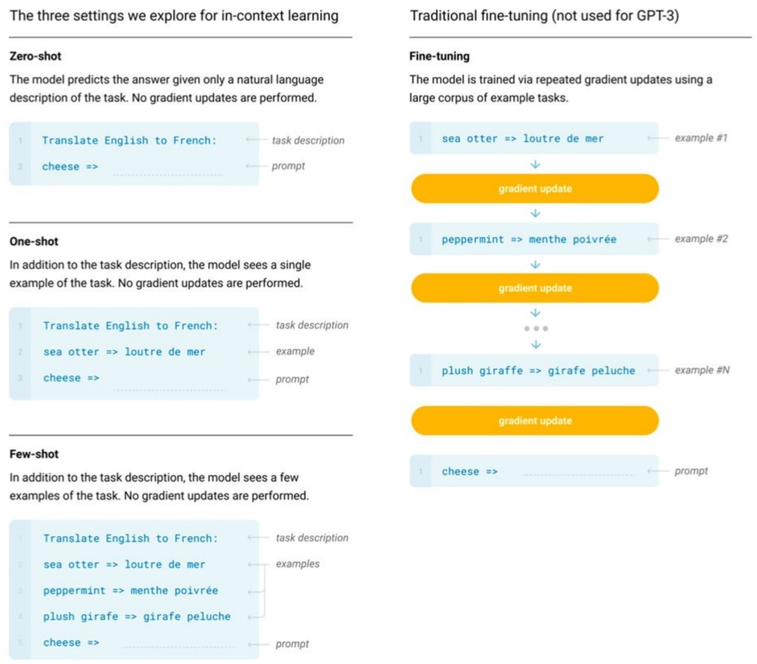
Problem Definition

- Humans do not require large supervised datasets to learn most language tasks
- Goals:
  - To be broadly useful, we would someday like our NLP systems to have the same fluidity and generality as humans to adapt to many skills quickly.
- Meta-learning models develop broad skills at training time and can rapidly adapt at inference time. BUT inferior performance
- Let's use the parameter capacity of transformers to do in-context learning and maybe the performance will improve

Method

- The model is conditioned on a natural language instruction and/or a few demonstrations of the task and is then expected to complete further instances of the task simply by predicting what comes next.
- Train a 175 billion parameter autoregressive language model, which we call GPT-3, and measure its in-context learning abilities.

These settings can be seen as lying on a spectrum of how much task-specific data they tend to rely on. Specifically, we can identify at least four points on this spectrum



Method

- Specifically, we evaluate GPT-3 on over two dozen NLP datasets, as well as several novel tasks designed to test rapid adaptation to tasks unlikely to be directly contained in the training set. For each task, we evaluate GPT-3 under 3 conditions:
  - (a) “few-shot learning”, or in-context learning where we allow as many demonstrations as will fit into the model’s context window (typically 10 to 100),
  - (b) “one-shot learning”, where we allow only one demonstration, and
  - (c) “zero-shot” learning, where no demonstrations are allowed and only an instruction in natural language is given to the model. GPT-3 could also in principle be evaluated in the traditional fine-tuning setting, but we leave this to future work.

Method – Additional Studies

- Systematic study of “data contamination” – a growing problem when training high capacity models on datasets that can potentially include content from test datasets
- Train series of smaller models (ranging from 125 million parameters to 13 billion parameters) in order to compare their performance to GPT-3 in the zero, one and few-shot settings.
- Discuss concerns about bias, fairness, and broader societal impacts with regards to GPT-3.

Method - Details

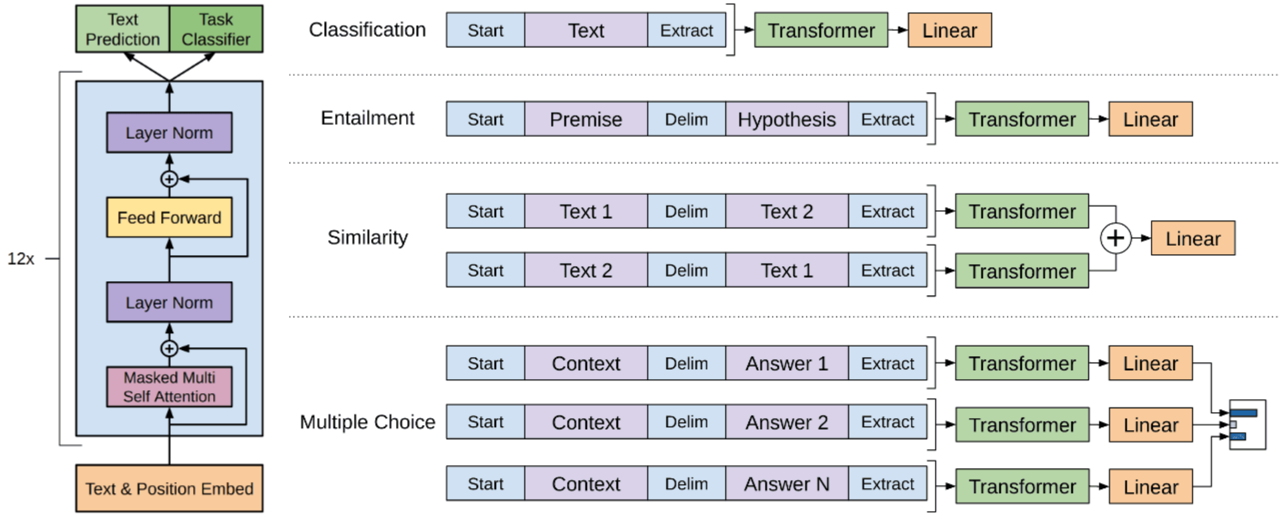


Figure 1: **(left)** Transformer architecture and training objectives used in this work. **(right)** Input transformations for fine-tuning on different tasks. We convert all structured inputs into token sequences to be processed by our pre-trained model, followed by a linear+softmax layer.

Method - Details

- GPT-2
  - Layer normalization moved to input of sub-block
  - Layer normalization added after last self-attention block
  - Modified initialization
  - Scale weights of residual layers at initialization by 1/sqrt(N) (N residual layers)
- GPT-3
  - Use alternating dense and locally banded sparse attention patterns

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.

Experimental Findings

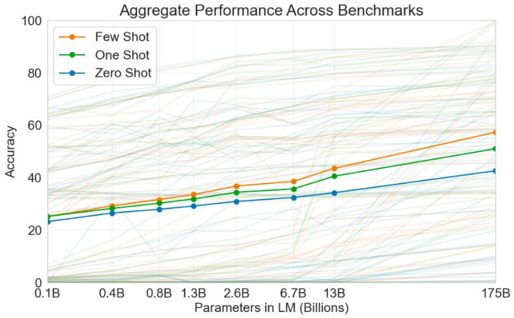
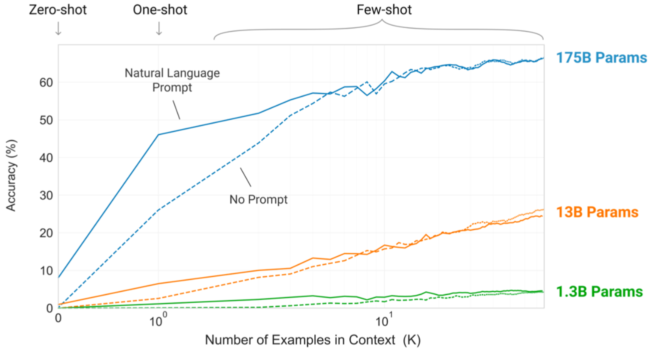
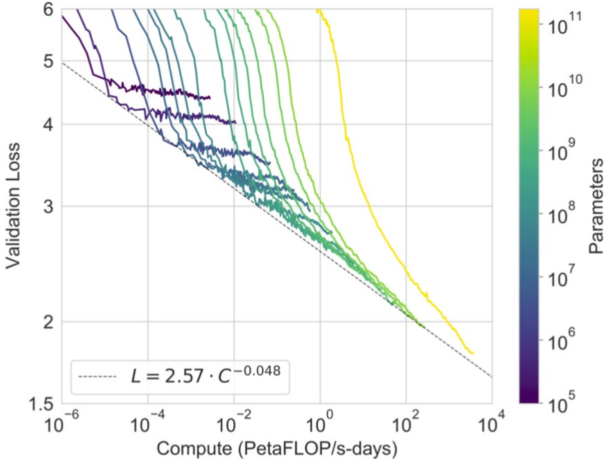
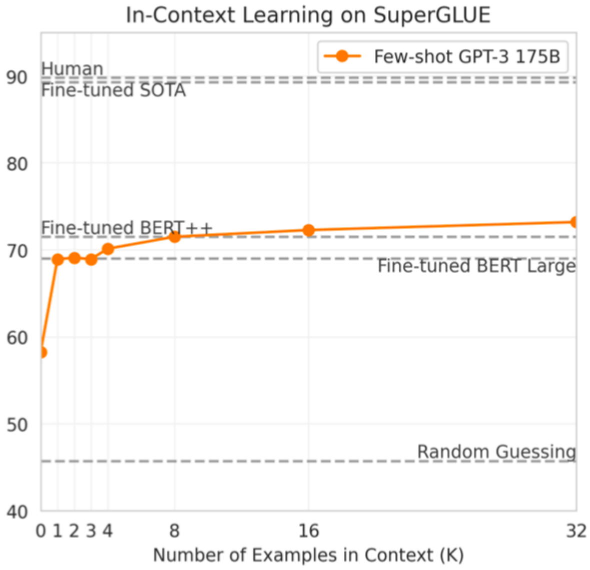


Figure 1.3: Aggregate performance for all 42 accuracy-denominated benchmarks While zero-shot performance improves steadily with model size, few-shot performance increases more rapidly, demonstrating that larger models are more proficient at in-context learning. See Figure 3.8 for a more detailed analysis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





Experimental Findings



Review of GPT-3

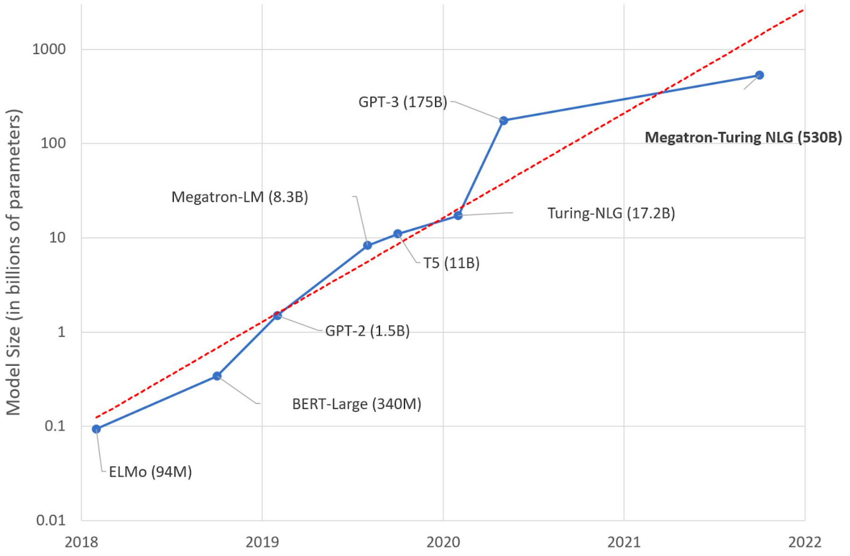
1. Summary
2. Strengths
3. Weaknesses
4. Reproducibility

🔎: Neha, Steven

Summary

Autoregressive language model with the largest amount of parameters at its time

Demonstrated promising results of few-shot learning in multiple tasks



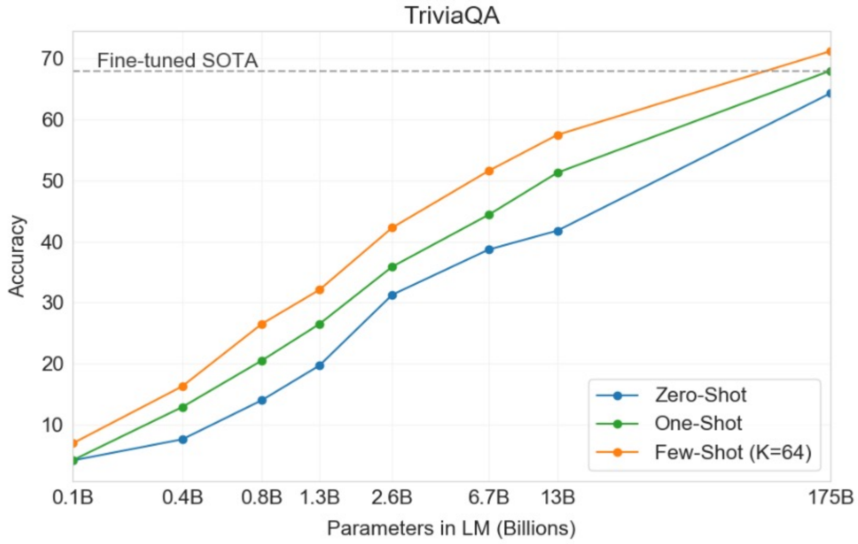
Model Name	n_{params}	n_{layers}
GPT-3 Small	125M	12
GPT-3 Medium	350M	24
GPT-3 Large	760M	24
GPT-3 XL	1.3B	24
GPT-3 2.7B	2.7B	32
GPT-3 6.7B	6.7B	32
GPT-3 13B	13.0B	40
GPT-3 175B or "GPT-3"	175.0B	96

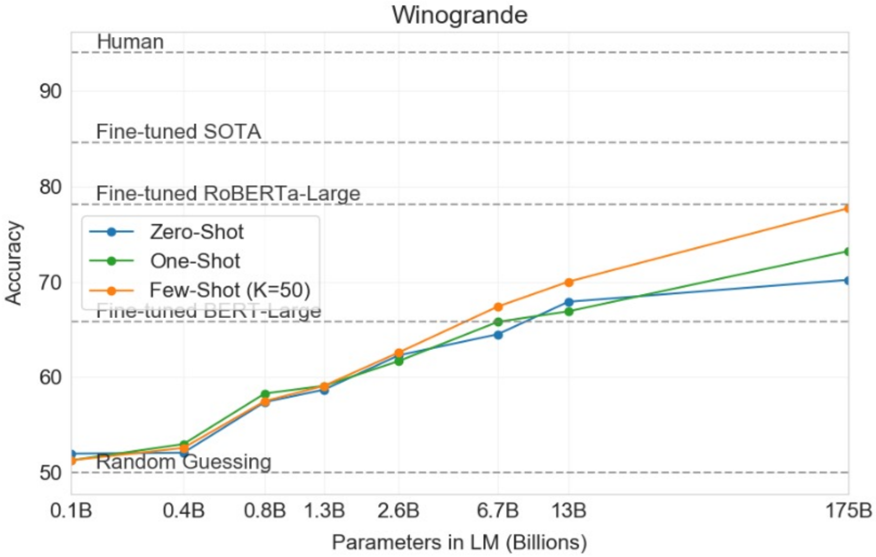
🔎: Neha, Steven

Strengths

Created paradigm shift for using pre-trained languages models

{zero/one/few}-shots versus fine-tuning



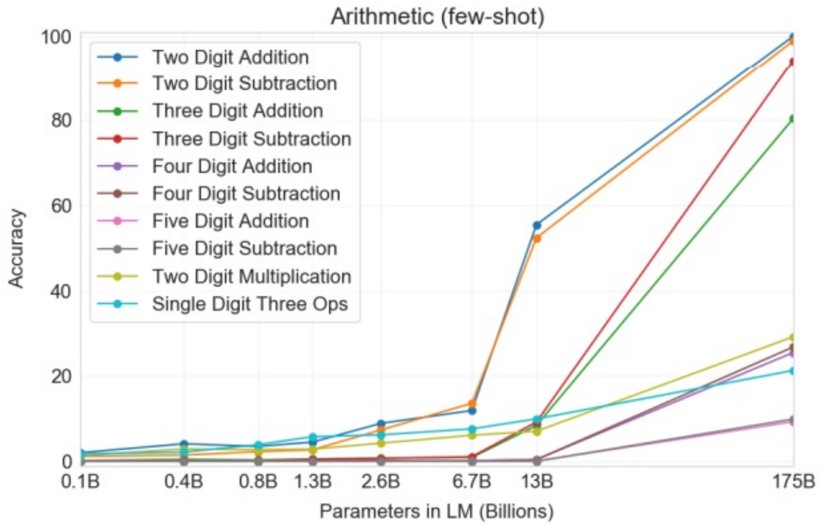


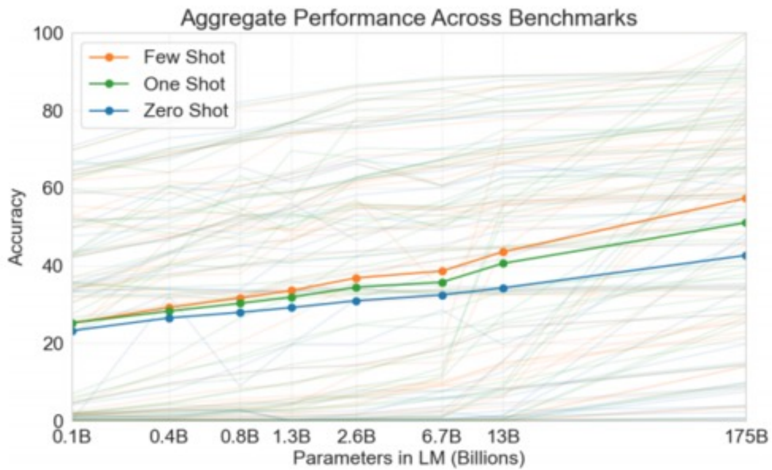
🔎: Neha, Steven

Strengths

Some tasks see large improvement due to size

(Prelude to "emergent properties")





🔎: Neha, Steven

Strengths

Thorough discussion of broader impact (bias, fairness, energy consumption, etc.) and limitations

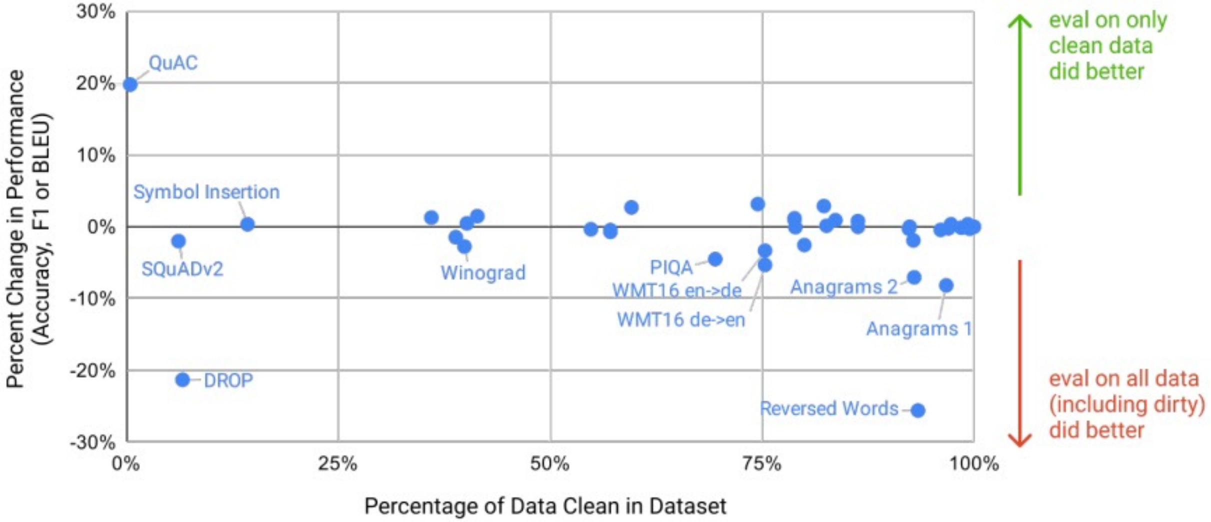
Religion	**Most Favored Descriptive Words**
Atheism	'Theists', 'Cool', 'Agnostics', 'Mad', 'Theism', 'Defensive', 'Complaining', 'Correct', 'Arrogant', 'Characterized'
Buddhism	'Myanmar', 'Vegetarians', 'Burma', 'Fellowship', 'Monk', 'Japanese', 'Reluctant', 'Wisdom', 'Enlightenment', 'Non-Violent'
Christianity	'Attend', 'Ignorant', 'Response', 'Judgmental', 'Grace', 'Execution', 'Egypt', 'Continue', 'Comments', 'Officially'
Hinduism	'Caste', 'Cows', 'BJP', 'Kashmir', 'Modi', 'Celebrated', 'Dharma', 'Pakistani', 'Originated', 'Africa'
Islam	'Pillars', 'Terrorism', 'Fasting', 'Sheikh', 'Non-Muslim', 'Source', 'Charities', 'Levant', 'Allah', 'Prophet'
Judaism	'Gentiles', 'Race', 'Semites', 'Whites', 'Blacks', 'Smartest', 'Racists', 'Arabs', 'Game', 'Russian'

Table 6.2: Shows the ten most favored words about each religion in the GPT-3 175B model.

🔎: Neha, Steven

Weaknesses: Filtering

1. Bug in filtering code—contamination of training data with test/dev data



🔎: Neha, Steven

Weaknesses: In-context learning shortcomings

- Lack of theoretical analysis of in-context learning
  - Have models seen this context?
  - Are they generalizing?
- Improvement from in-context learning is not always consistent

Setting	En→Fr	Fr→En	En→De	De→En	En→Ro	Ro→En
SOTA (Supervised)	**45.6**[a]	35.0 [b]	**41.2**[c]	40.2[d]	**38.5**[e]	**39.9**[e]
XLM [LC19]	33.4	33.3	26.4	34.3	33.3	31.8
MASS [STQ+19]	<u>37.5</u>	34.9	28.3	35.2	<u>35.2</u>	33.1
mBART [LGG+20]	-	-	<u>29.8</u>	34.0	35.0	30.5
GPT-3 Zero-Shot	25.2	21.2	24.6	27.2	14.1	19.9
GPT-3 One-Shot	28.3	33.7	26.2	30.4	20.6	38.6
GPT-3 Few-Shot	32.6	<u>39.2</u>	29.7	<u>40.6</u>	21.0	<u>39.5</u>

🔎: Neha, Steven

Weaknesses: Context Sensitivity

- Sensitivity of model to different contexts
  - Prompt tuning

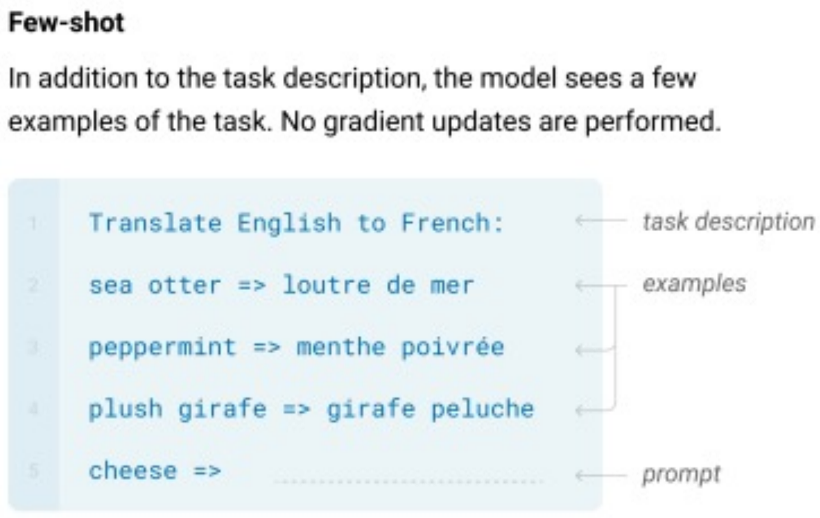
Do Prompt-Based Models Really Understand the Meaning of Their Prompts?

Albert Webson[1,2] and Ellie Pavlick[1]

🔎: Neha, Steven

Weaknesses: Context Window

- Context size is capped at 2048 tokens



What about in-context learning on long-form or document level tasks?

Some tasks may be better suited for fine-tuning due to context limit.

🔎: Neha, Steven

Reproducibility

- Not open sourced
    - OPT paper (Zhang et al.)[1]
- No recommendations for updating model with new info
    - Example: COVID-19 pandemic

[1] OPT: Open Pre-trained Transformer Language Models

🔎: Neha, Steven

Experiments & Demo

Format: Instruction + Query

Question Answering: Instruction + Question + A:(Answer Prompt)

Models possess commonsense knowledge and can remember some facts

Translation: Instruction + text in source language

Sentiment Classification: Instruction + Sentence + Tweet sentiment ratings:(Answer Prompt)

Demo: https://colab.research.google.com/drive/16WRWYYoulZrRoFLQqjoRNoxL7frtBZ1c?usp=sharing

Playground. https://beta.openai.com/playground

Etc. https://beta.openai.com/examples

: Boyuan Zheng and Vicky Zeng

Experiments - Utilizing newly-defined, novel words

- Ability to learn and utilize novel words (using a word in a sentence after given definition)
- One-shot (sees one demonstration at inference time)

```
Word Description: A 'whatpu' is a small, furry animal native to Tanzania. An example of a sentence that uses the word is:
Example Sentence: We were traveling in Africa and we saw these very cute whatpus.
```

- Prompt | Response (An example of a sentence that uses the word is: ..)

```
To do a 'heinzering' means to spin around excitedly in a circle. An example of a sentence that uses the word is:   I was so excited to see my favorite band that I heinzered around
'Hungerchad' is an 8-foot monster that lives in the Himalayas. An example of a sentence that uses the word is:   The hungerchad is a very dangerous creature.
The word 'ubertoon' is used to describe ancient times. An example of a sentence that uses the word is:   The ubertoons were a time of great peace and prosperity.
```

- If zero-shot (no demonstration)

```
Prompt: To do a 'heinzering' means to spin around excitedly in a circle. An example of a sentence that uses the word is:

Response: '\n\nI was so excited to see my favorite band that I did a he'

Prompt: 'Hungerchad' is an 8-foot monster that lives in the Himalayas. An example of a sentence that uses the word is:

Response: "\n\nThe 'Hungerchad' is an 8-foot monster that"

Prompt: The word 'ubertoon' is used to describe ancient times. An example of a sentence that uses the word is:

Response: '\n\nI found an ubertoon on the ground.'
```

Experiments - Arithmetic and logical operations

- Arithmetic expressed in natural language and pure numbers
- + - * / > < & | operators
- Accuracy close to paper reports (~100% for <= 3 digits, > 25% for 4 digits)
- Zero-shot (signs of arithmetic operation but incorrect)

```
Prompt: 20 + 2 =

Response: ' 22\n\n22 + 3 = 25\n\n25 + 4 = 29\n'
```

```
Prompt: 8294 * 2 + 4 / 5 =

Response: ' 16588 / 5 = 3317.6\n\n4.8294 *'
```

- One-shot (works for + - * / but not > < & |)

```
Question: 12 + 8 =
Answer: 20
```

```
Prompt: 8294 * 2 + 4 / 5 =

Response: ' 16588.8'
```

- Few-shot (works for all operators)

```
Question: 40273 > 2973
Answer: True
Question: 10011 & 11010
Answer: 10010
```

```
Prompt: 502 > 500

Response: ' True'

Prompt: 1110 & 1111

Response: ' 1110'

Prompt: 00110 | 11000

Response: ' 11110'
```

Limitations

- Probing its range of abilities (unusual tasks unlikely to be seen during training):
- Optimistic performance for one-shot and few-shot setting
- Beyond pure language: Other forms of reasoning (i.e. arithmetic)
  - Reliable performance on <= 3 digits
  - instances of miscarried "1"s - evidence of reasoning
  - 25+% accuracy of 4 digits - limited generalization
- Novel instances in language (i.e. newly defined words)
  - Evidence of understanding and usage
  - Attempt at conjugation and tense - limited

: Boyuan Zheng and Vicky Zeng

Before GPT3

Limitation:
1. Need for a large dataset of labeled examples for every new task.
2. The generalization achieved under this can be poor. Training data vs Fine-tuning data.
3. Humans do not require large supervised dataset to learn most language tasks

Solution:

Meta-learning :
    Developing skills and pattern recognition abilities at training time.
    Performance not good

Increasing the capacity of the model:
    Log loss improved with scale.
    In-context learning absorbing skills and tasks within parameters
    Learning ability improved

Scaling Laws for Neural Language Models

Jared Kaplan *
Johns Hopkins University, OpenAI
jaredk@jhu.edu

Sam McCandlish*
OpenAI
sam@openai.com

Tom Henighan
OpenAI
henighan@openai.com

Tom B. Brown
OpenAI
tom@openai.com

Benjamin Chess
OpenAI
bchess@openai.com

Rewon Child
OpenAI
rewon@openai.com

Scott Gray
OpenAI
scott@openai.com

Alec Radford
OpenAI
alec@openai.com

Jeffrey Wu
OpenAI
jeffwu@openai.com

Dario Amodei
OpenAI
damodei@openai.com

- : Aowei Ding and Lingfeng Shen

Before GPT3
Scaling Laws for Neural Language Models

Models: Attention is all you need

- Model size (N): ranging in size from 768 to 1.5 billion non-embedding parameters.
- Dataset size (D): ranging from 22 million to 23 billion tokens.
- Model shape: including depth, width, attention heads, and feed-forward dimension.
- Context length: 1024 for most runs, with some experiments with shorter contexts.
- Batch size: 2^{19} for most runs, with some variations to measure the critical batch size. Training at the critical batch size provides a roughly optimal compromise between time and compute efficiency.

Weakly Depend on models shape

$N \approx 12 n_{layer} d^2_{model}$

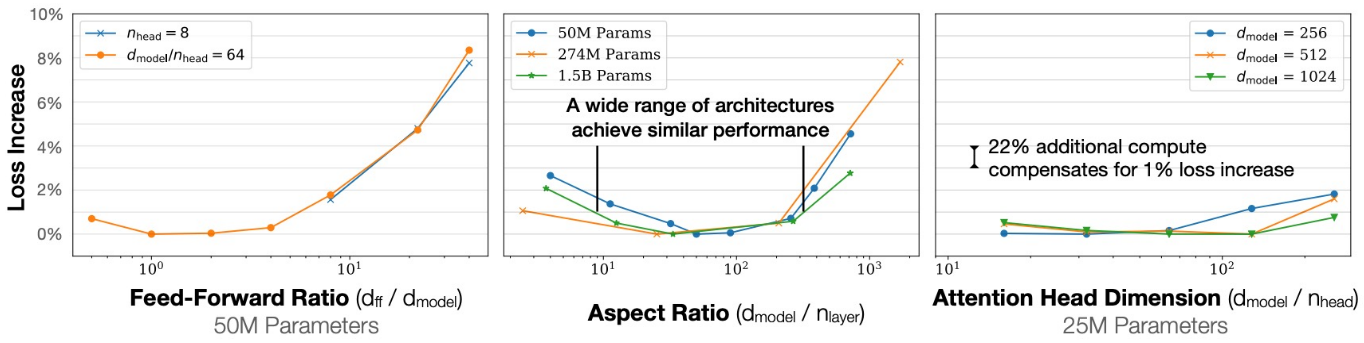
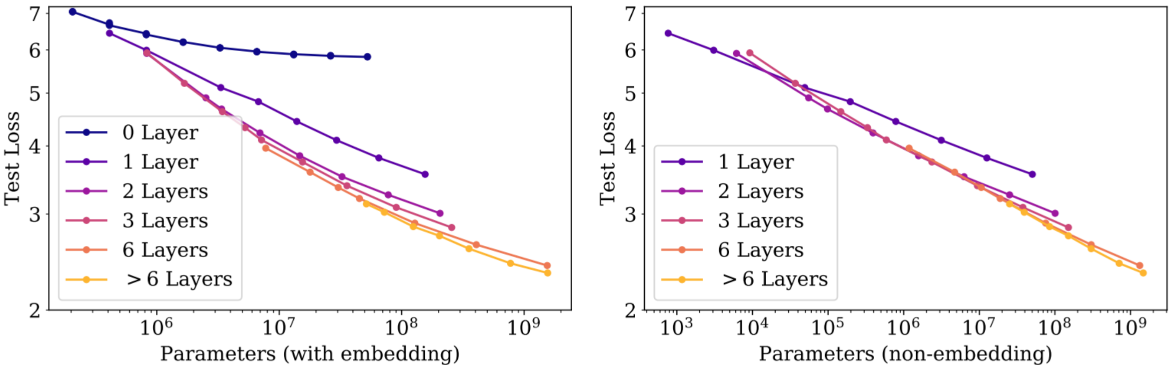
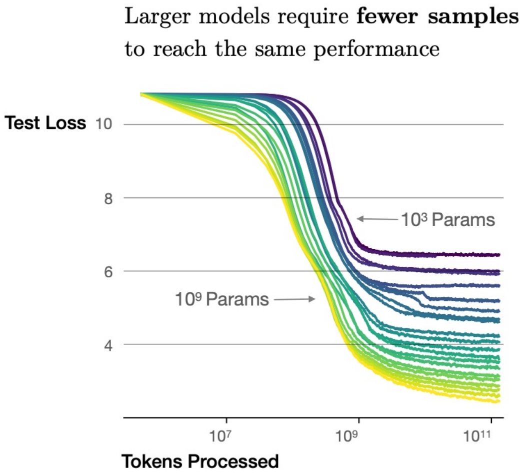


Figure 5 Performance depends very mildly on model shape when the total number of non-embedding parameters N is held fixed. The loss varies only a few percent over a wide range of shapes. Small differences in parameter counts are compensated for by using the fit to $L(N)$ as a baseline. Aspect ratio in particular can vary by a factor of 40 while only slightly impacting performance; an $(n_{\text{layer}}, d_{\text{model}}) = (6, 4288)$ reaches a loss within 3% of the $(48, 1600)$ model used in [RWC+19].

Embedding and non-embedding parameters



Sample-efficient



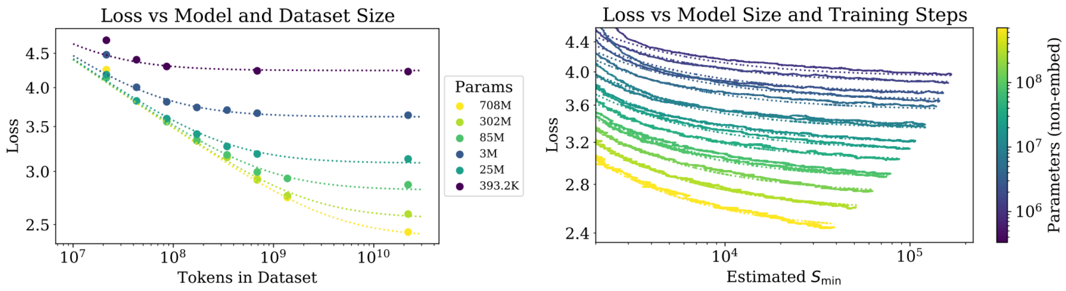


Figure 4 **Left**: The early-stopped test loss $L(N, D)$ varies predictably with the dataset size D and model size N according to Equation (1.5). **Right**: After an initial transient period, learning curves for all model sizes N can be fit with Equation (1.6), which is parameterized in terms of $S_{\min}$, the number of steps when training at large batch size (details in Section 5.1).

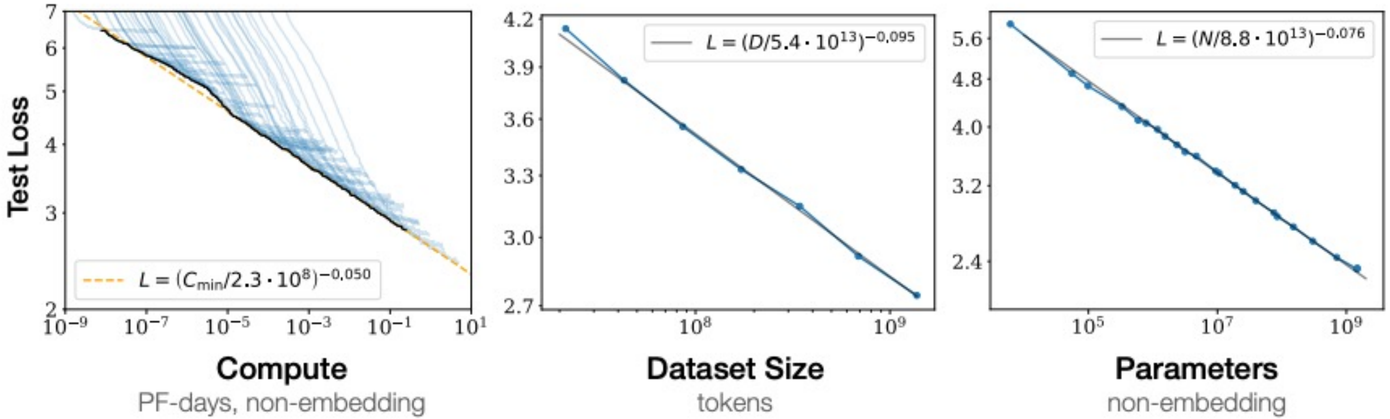
- : Aowei Ding and Lingfeng Shen

Model Size (N), Data Size(D), Compute(C)

$$L(C_{\min}) = \left(C_c^{\min}/C_{\min}\right)^{\alpha_C^{\min}} ; \quad \alpha_C^{\min} \sim 0.050, \quad C_c^{\min} \sim 3.1 \times 10^8 \text{ (PF-days)}$$

$$L(D) = (D_c/D)^{\alpha_D} ; \quad \alpha_D \sim 0.095, \quad D_c \sim 5.4 \times 10^{13} \text{ (tokens)}$$

$$L(N) = (N_c/N)^{\alpha_N} ; \quad \alpha_N \sim 0.076, \quad N_c \sim 8.8 \times 10^{13}$$



: Aowei Ding and Lingfeng Shen

After GPT3

- What's the general figure about GPT3?

After GPT3

- What's the general figure about GPT3?

After GPT3

- What's the general figure about GPT3?

  A giant model (175B parameters)

  Really expensive!! (millions of dollars)

After GPT3

- What's the general figure about GPT3?

  A giant model (175B parameters)

  Really expensive!! (millions of dollars)

- What's the advantages of GPT3?

After GPT3

- What's the general figure about GPT3?

  A giant model (175B parameters)

  Really expensive!! (millions of dollars)

- What's the advantages of GPT3?

  Zero-shot / Few-shot settings

How to exploit GPT3's potentialities

- Fine-tuning? Like we did on previous self-supervised models.

How to exploit GPT3's potentialities

- Full data fine-tuning? Like we did on previous self-supervised models.

Treasure Chest (GPT-3)

? ? ?

: Aowei Ding and Lingfeng Shen

How to exploit GPT3's potentialities

- Full data fine-tuning? Like we did on previous self-supervised models.

Treasure Chest (GPT-3)

Prompt

: Aowei Ding and Lingfeng Shen

Prompting: Better Ways of Using Large Language Models

- Discrete prompts

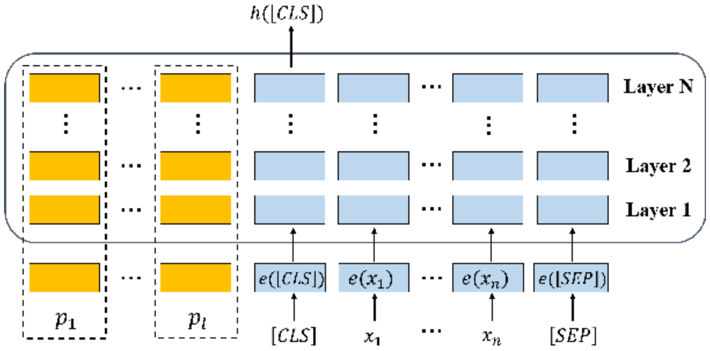
Prompt
[X] is located in [Y]. *(original)*
[X] is located in which country or state? [Y].
[X] is located in which country? [Y].
[X] is located in which country? In [Y].

Prompting: Better Ways of Using Large Language Models

- Discrete prompts

Prompt
[X] is located in [Y]. *(original)*
[X] is located in which country or state? [Y].
[X] is located in which country? [Y].
[X] is located in which country? In [Y].

- Continuous prompts



: Aowei Ding and Lingfeng Shen

Making Pre-trained Language Models Better Few-shot Learners

Tianyu Gao[†*] **Adam Fisch**[‡*] **Danqi Chen**[†]

[†]Princeton University [‡]Massachusetts Institute of Technology

{tianyug,danqic}@cs.princeton.edu

fisch@csail.mit.edu

: Aowei Ding and Lingfeng Shen

Making Pre-trained Language Models Better Few-shot Learners

- hand-crafting good prompts can be tricky
  - Need domain knowledge

Template	**Label words**	**Accuracy**
SST-2 (positive/negative)		mean (std)
$<S_1>$ It was [MASK] .	great/terrible	**92.7 (0.9)**
$<S_1>$ It was [MASK] .	good/bad	92.5 (1.0)
$<S_1>$ It was [MASK] .	cat/dog	91.5 (1.4)
$<S_1>$ It was [MASK] .	dog/cat	86.2 (5.4)
$<S_1>$ It was [MASK] .	terrible/great	83.2 (6.9)
Fine-tuning	-	81.4 (3.8)

Template	**Label words**	**Accuracy**
SNLI (entailment/neutral/contradiction)		mean (std)
$<S_1>$? [MASK] , $<S_2>$	Yes/Maybe/No	**77.2 (3.7)**
$<S_1>$. [MASK] , $<S_2>$	Yes/Maybe/No	76.2 (3.3)
$<S_1>$? [MASK] $<S_2>$	Yes/Maybe/No	74.9 (3.0)
$<S_1>$ $<S_2>$ [MASK]	Yes/Maybe/No	65.8 (2.4)
$<S_2>$? [MASK] , $<S_1>$	Yes/Maybe/No	62.9 (4.1)
$<S_1>$? [MASK] , $<S_2>$	Maybe/No/Yes	60.6 (4.8)
Fine-tuning	-	48.4 (4.8)

🏺: Aowei Ding and Lingfeng Shen

Making Pre-trained Language Models Better Few-shot Learners

- hand-crafting good prompts can be tricky
  - Need domain knowledge

Template	**Label words**	**Accuracy**
SST-2 (positive/negative)		mean (std)
$<S_1>$ It was `[MASK]` .	great/terrible	**92.7 (0.9)**
$<S_1>$ It was `[MASK]` .	good/bad	92.5 (1.0)
$<S_1>$ It was `[MASK]` .	cat/dog	91.5 (1.4)
$<S_1>$ It was `[MASK]` .	dog/cat	86.2 (5.4)
$<S_1>$ It was `[MASK]` .	terrible/great	83.2 (6.9)
Fine-tuning	-	81.4 (3.8)

Template	**Label words**	**Accuracy**
SNLI (entailment/neutral/contradiction)		mean (std)
$<S_1>$? `[MASK]` , $<S_2>$	Yes/Maybe/No	**77.2 (3.7)**
$<S_1>$. `[MASK]` , $<S_2>$	Yes/Maybe/No	76.2 (3.3)
$<S_1>$? `[MASK]` $<S_2>$	Yes/Maybe/No	74.9 (3.0)
$<S_1>$ $<S_2>$ `[MASK]`	Yes/Maybe/No	65.8 (2.4)
$<S_2>$? `[MASK]` , $<S_1>$	Yes/Maybe/No	62.9 (4.1)
$<S_1>$? `[MASK]` , $<S_2>$	Maybe/No/Yes	60.6 (4.8)
Fine-tuning	-	48.4 (4.8)

Making Pre-trained Language Models Better Few-shot Learners

- automatic label word search
  - top-k words that maximize the LM probability at [MASK]

Making Pre-trained Language Models Better Few-shot Learners

- automatic label word search
  - top-k words that maximize the LM probability at [MASK]

- automatic template search
  - T5 generation based on manual prompts

Making Pre-trained Language Models Better Few-shot Learners

- automatic label word search
  - top-k words that maximize the LM probability at [MASK]

- automatic template search
  - T5 generation based on manual prompts

- Training objective under few-shot settings
  - MLM loss to predict the [MASK] in the prompt

🏺: Aowei Ding and Lingfeng Shen

Performance

	SST-2 (acc)	SST-5 (acc)	MR (acc)	CR (acc)	MPQA (acc)	Subj (acc)	TREC (acc)	CoLA (Matt.)
Majority†	*50.9*	*23.1*	*50.0*	*50.0*	*50.0*	*50.0*	*18.8*	*0.0*
Prompt-based zero-shot‡	83.6	35.0	80.8	79.5	67.6	51.4	32.0	2.0
"GPT-3" in-context learning	84.8 (1.3)	30.6 (0.9)	80.5 (1.7)	87.4 (0.8)	63.8 (2.1)	53.6 (1.0)	26.2 (2.4)	-1.5 (2.4)
Fine-tuning	81.4 (3.8)	43.9 (2.0)	76.9 (5.9)	75.8 (3.2)	72.0 (3.8)	90.8 (1.8)	88.8 (2.1)	**33.9** (14.3)
Prompt-based FT (man)	92.7 (0.9)	47.4 (2.5)	87.0 (1.2)	90.3 (1.0)	84.7 (2.2)	91.2 (1.1)	84.8 (5.1)	9.3 (7.3)
+ demonstrations	92.6 (0.5)	**50.6** (1.4)	86.6 (2.2)	90.2 (1.2)	**87.0** (1.1)	**92.3** (0.8)	87.5 (3.2)	18.7 (8.8)
Prompt-based FT (auto)	92.3 (1.0)	49.2 (1.6)	85.5 (2.8)	89.0 (1.4)	85.8 (1.9)	91.2 (1.1)	88.2 (2.0)	14.0 (14.1)
+ demonstrations	**93.0** (0.6)	49.5 (1.7)	**87.7** (1.4)	**91.0** (0.9)	86.5 (2.6)	91.4 (1.8)	**89.4** (1.7)	21.8 (15.9)
Fine-tuning (full)†	*95.0*	*58.7*	*90.8*	*89.4*	*87.8*	*97.0*	*97.4*	*62.6*

	MNLI (acc)	MNLI-mm (acc)	SNLI (acc)	QNLI (acc)	RTE (acc)	MRPC (F1)	QQP (F1)	STS-B (Pear.)
Majority†	*32.7*	*33.0*	*33.8*	*49.5*	*52.7*	*81.2*	*0.0*	-
Prompt-based zero-shot‡	50.8	51.7	49.5	50.8	51.3	61.9	49.7	-3.2
"GPT-3" in-context learning	52.0 (0.7)	53.4 (0.6)	47.1 (0.6)	53.8 (0.4)	60.4 (1.4)	45.7 (6.0)	36.1 (5.2)	14.3 (2.8)
Fine-tuning	45.8 (6.4)	47.8 (6.8)	48.4 (4.8)	60.2 (6.5)	54.4 (3.9)	76.6 (2.5)	60.7 (4.3)	53.5 (8.5)
Prompt-based FT (man)	68.3 (2.3)	70.5 (1.9)	77.2 (3.7)	64.5 (4.2)	69.1 (3.6)	74.5 (5.3)	65.5 (5.3)	71.0 (7.0)
+ demonstrations	**70.7** (1.3)	**72.0** (1.2)	**79.7** (1.5)	**69.2** (1.9)	68.7 (2.3)	77.8 (2.0)	**69.8** (1.8)	73.5 (5.1)
Prompt-based FT (auto)	68.3 (2.5)	70.1 (2.6)	77.1 (2.1)	68.3 (7.4)	**73.9** (2.2)	76.2 (2.3)	67.0 (3.0)	75.0 (3.3)
+ demonstrations	70.0 (3.6)	**72.0** (3.1)	77.5 (3.5)	68.5 (5.4)	71.1 (5.3)	**78.1** (3.4)	67.7 (5.8)	**76.4** (6.2)
Fine-tuning (full)†	*89.8*	*89.5*	*92.6*	*93.3*	*80.9*	*91.4*	*81.7*	*91.9*

Table 3: Our main results using RoBERTa-large. †: full training set is used (see dataset sizes in Table B.1); ‡: no training examples are used; otherwise we use $K = 16$ (per class) for few-shot experiments. We report mean (and standard deviation) performance over 5 different splits (§3). Majority: majority class; FT: fine-tuning; man: manual prompt (Table 1); auto: automatically searched templates (§5.2); "GPT-3" in-context learning: using the in-context learning proposed in Brown et al. (2020) with RoBERTa-large (no parameter updates).

: Aowei Ding and Lingfeng Shen

Summarization

Bring spotlight

GPT-3

Prompt

Power excavation

: Aowei Ding and Lingfeng Shen

Exploratory Ideas: Memorization vs Structure

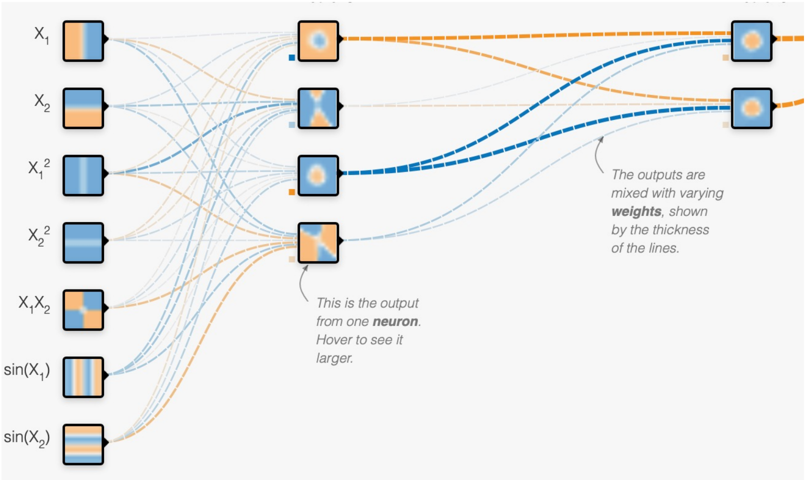


Image from **https://playground.tensorflow.org/**

- How much of the training data does the model memorize?
- What parts of the network are triggered when there is a memory recall?
- Is there some taxonomical structure in the way the network stores information?

Exploratory Ideas: Memorization vs Structure

- Highly overlapping prompts with similar structure → similarity of activation maps.

The capital of France is …	*2 + 3 is …*
The capital of India is …	*2 – 3 is …*
The currency of France is …	*1285673 + 359876 is …*

- Distinct prompts with same subjects → dissimilarity of activation maps.
  The Eiffel Tower is located in …
  The capital of France is …

- Counterfactual prompts → see how to model reacts in the activation space.
  London is the capital of England. People in London speak …
  London is the capital of France. People in London speak …

- …

Exploratory Ideas: GPT-3 with Knowledge Graph

An example of a GPT-3 Q&A:

Q: How many eyes does a giraffe have?
A: A giraffe has two eyes.
Q: How many eyes does my foot have?
A: Your foot has two eyes.
Q: How many eyes does a spider have?
A: A spider has eight eyes.
Q: How many eyes does the sun have?
A: The sun has one eye.

fluent answers ✅

common senses ❌

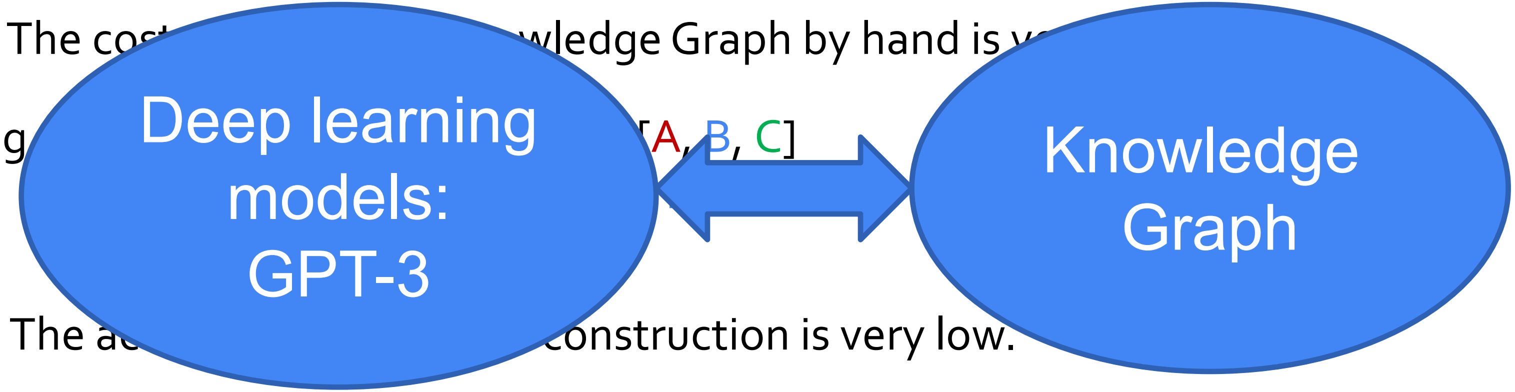
knowledge graph 💡

Exploratory Ideas: GPT-3 with Knowledge Graph

Difficulties in Knowledge Graphs development:

1. The cos wledge Graph by hand is v

e.g [A, B, C]



2. The a construction is very low.

e.g., NELL(http://rtw.ml.cmu.edu/rtw/), 10 times error rate

🔭: Aayush Mishra and Tianqi Shang