

# Introduction to Deep Learning

GPT Siblings

Sungjoon Choi, Korea University

# Content

- GPT
- GPT-2
- BERT
- RoBERTa
- ALBERT
- GPT-3
- InstructGPT
- GPT-4

# GPT

"Improving Language Understanding by Generative Pre-Training," 2018



# Generative Pre-Training (GPT)

The goal is to learn a **universal representation** that transfers with little adaptation to a wide range of tasks.

# GPT



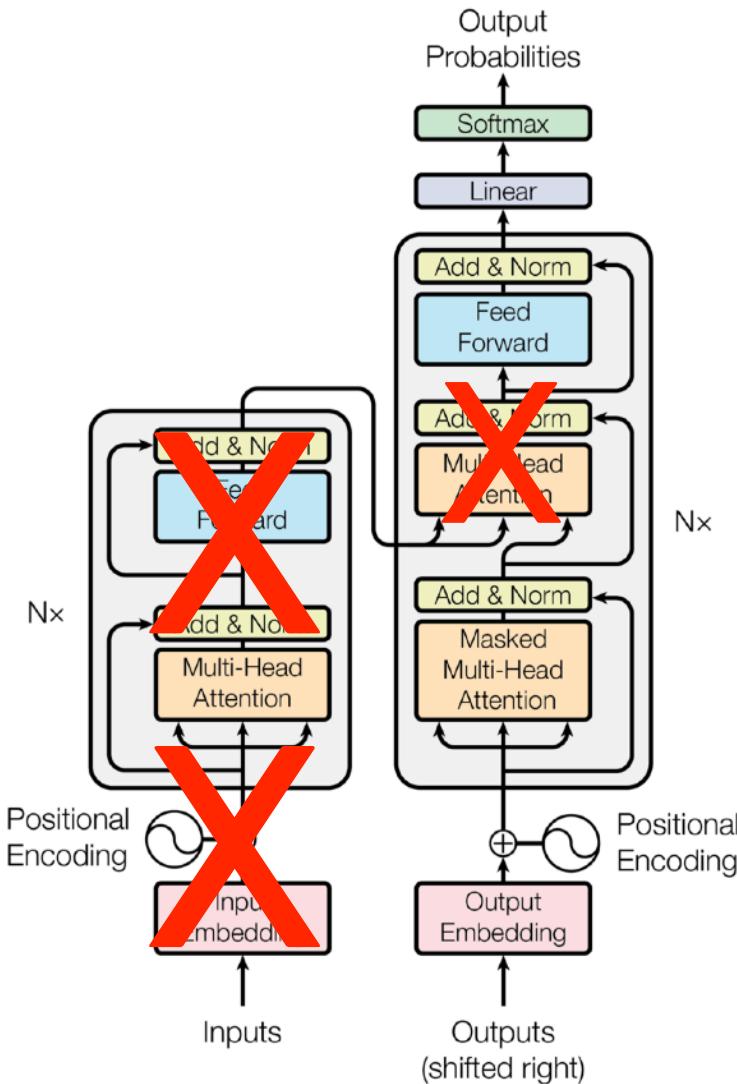
- GPT consists of two stages: the first stage is learning a high-capacity language model on a large corpus of text and followed by a fine-tuning stage.
- Unsupervised pre-training
  - Given an unsupervised corpus of tokens  $\mathcal{U} = \{u_1, \dots, u_n\}$ , a standard [language modeling objective](#) is used to maximize the following likelihood:

$$L_1(\mathcal{U}) = \sum_i \log P(u_i | u_{i-k}, \dots, u_{i=1}; \Theta)$$

where  $k$  is the size of the context window, and the conditional probability  $P$  is modeled using a neural network with parameters  $\Theta$

- A multi-layer [Transformer decoder](#) is used.

# Why Transformer Decoder?



# Traversal-Style Approach

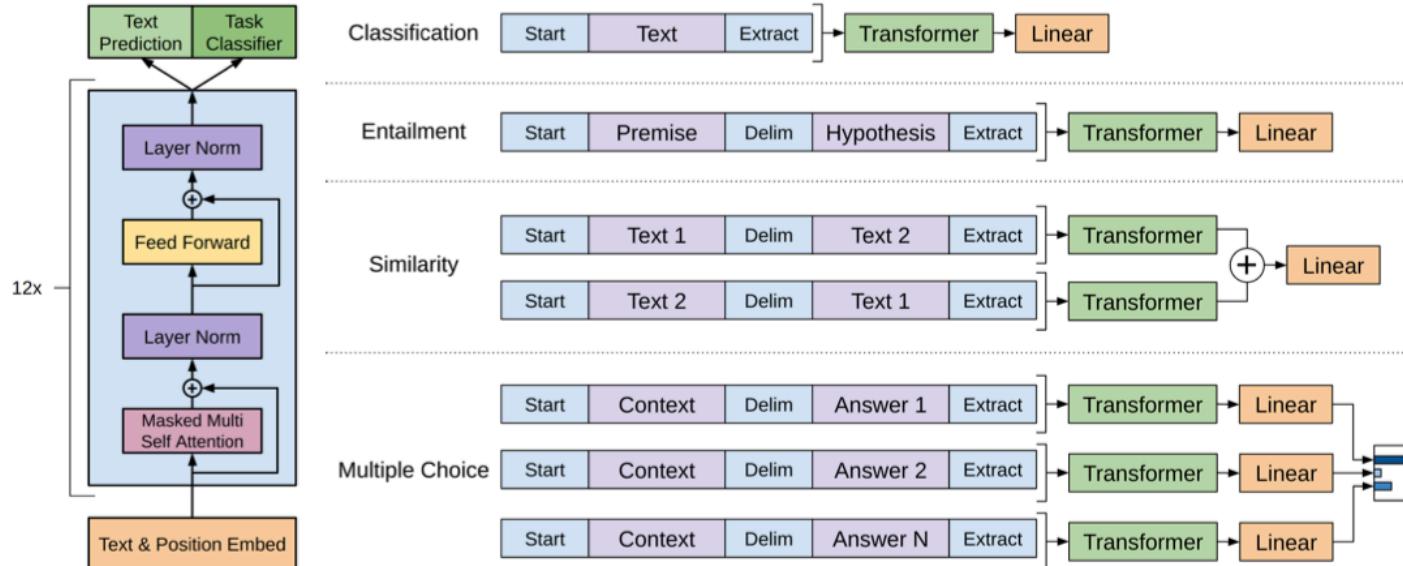


Figure 1: **(left)** Transformer architecture and training objectives used in this work. **(right)** Input transformations for fine-tuning on different tasks. We convert all structured inputs into token sequences to be processed by our pre-trained model, followed by a linear+softmax layer.

I dislike old cabin cruisers. (negative)

Peter is snoring. \\$ A man sleeps. (true)

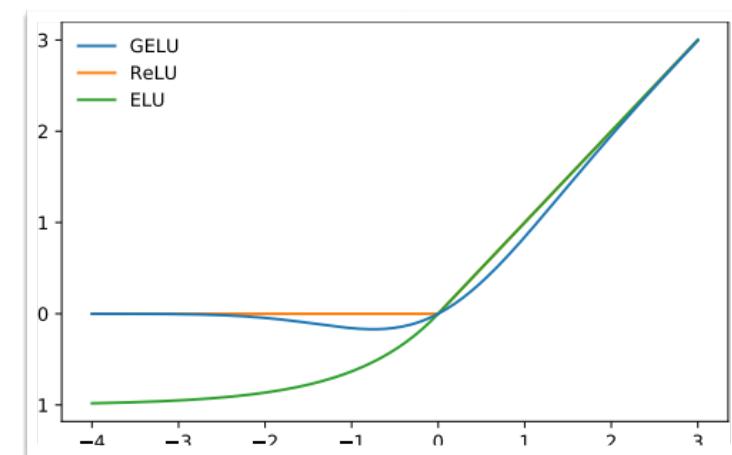
President greets the press in Chicago.  
Obama speaks in Illinois.  
(similar)

# Dataset

- **BooksCorpus** dataset is used for pre-training
  - It contains over 7,000 unique unpublished books from a variety of genres including Adventure, Fantasy, and Romance.
  - It contains long stretches of contiguous text, which allows the generative model to learn to condition on long-range information.
- **1D Word Benchmark** is an alternative option
  - It has approximately the same size as BooksCorpus but is shuffled at a sentence level, destroying long-range structure.
  - GPT achieves a very low performance on this dataset.

# Implementation Details

- 12-layer-decoder-only transformer with masked self-attention heads
  - 768 dimensional states and 12 attention heads
  - For the position-wise-feed-forward networks, 3072 dimensional inner states are used.
  - Adam optimizer with a maximum learning rate of  $2.5 * 10^{-4}$  with a cosine schedule
  - Gaussian Error Linear Unit (GELU) activations
  - Learned position embeddings
- Fine-tuning details
  - Dropout to the classifier with a rate of 0.1
  - Learning rate of  $6.25 * 10^{-5}$  with a linear decay and a batchsize of 32
  - 3 epochs of training was sufficient for most cases



# Experiments

Table 2: Experimental results on natural language inference tasks, comparing our model with current state-of-the-art methods. 5x indicates an ensemble of 5 models. All datasets use accuracy as the evaluation metric.

| Method                              | MNLI-m      | MNLI-mm     | SNLI        | SciTail     | QNLI        | RTE         |
|-------------------------------------|-------------|-------------|-------------|-------------|-------------|-------------|
| ESIM + ELMo [44] (5x)               | -           | -           | <u>89.3</u> | -           | -           | -           |
| CAFE [58] (5x)                      | 80.2        | 79.0        | <u>89.3</u> | -           | -           | -           |
| Stochastic Answer Network [35] (3x) | <u>80.6</u> | <u>80.1</u> | -           | -           | -           | -           |
| CAFE [58]                           | 78.7        | 77.9        | 88.5        | <u>83.3</u> |             |             |
| GenSen [64]                         | 71.4        | 71.3        | -           | -           | <u>82.3</u> | 59.2        |
| Multi-task BiLSTM + Attn [64]       | 72.2        | 72.1        | -           | -           | <u>82.1</u> | <b>61.7</b> |
| Finetuned Transformer LM (ours)     | <b>82.1</b> | <b>81.4</b> | <b>89.9</b> | <b>88.3</b> | <b>88.1</b> | 56.0        |

**Natural language inference (NLI):** reading a pair of sentences and judging the relationship between from one of entailment, contradiction, or neutral

# Experiments

Table 3: Results on question answering and commonsense reasoning, comparing our model with current state-of-the-art methods.. 9x means an ensemble of 9 models.

| Method                          | Story Cloze | RACE-m      | RACE-h      | RACE        |
|---------------------------------|-------------|-------------|-------------|-------------|
| val-LS-skip [55]                | 76.5        | -           | -           | -           |
| Hidden Coherence Model [7]      | <u>77.6</u> | -           | -           | -           |
| Dynamic Fusion Net [67] (9x)    | -           | 55.6        | 49.4        | 51.2        |
| BiAttention MRU [59] (9x)       | -           | <u>60.2</u> | <u>50.3</u> | <u>53.3</u> |
| Finetuned Transformer LM (ours) | <b>86.5</b> | <b>62.9</b> | <b>57.4</b> | <b>59.0</b> |

**Question answering and commonsense reasoning:** English passages with associated questions from middle and high school exams / selecting the correct ending to multi-sentence stories from two options.

# Experiments

Table 4: Semantic similarity and classification results, comparing our model with current state-of-the-art methods. All task evaluations in this table were done using the GLUE benchmark. (*mc*= Mathews correlation, *acc*=Accuracy, *pc*=Pearson correlation)

| Method                                | Classification |               | Semantic Similarity |              | GLUE        |             |
|---------------------------------------|----------------|---------------|---------------------|--------------|-------------|-------------|
|                                       | CoLA<br>(mc)   | SST2<br>(acc) | MRPC<br>(F1)        | STSB<br>(pc) | QQP<br>(F1) |             |
| Sparse byte mLSTM [16]                | -              | <b>93.2</b>   | -                   | -            | -           | -           |
| TF-KLD [23]                           | -              | -             | <b>86.0</b>         | -            | -           | -           |
| ECNU (mixed ensemble) [60]            | -              | -             | -                   | <b>81.0</b>  | -           | -           |
| Single-task BiLSTM + ELMo + Attn [64] | 35.0           | 90.2          | 80.2                | 55.5         | <u>66.1</u> | 64.8        |
| Multi-task BiLSTM + ELMo + Attn [64]  | 18.9           | 91.6          | 83.5                | 72.8         | <u>63.3</u> | <u>68.9</u> |
| Finetuned Transformer LM (ours)       | <b>45.4</b>    | 91.3          | 82.3                | <b>82.0</b>  | <b>70.3</b> | <b>72.8</b> |

**Semantic Similarity:** Predicting whether two sentences are semantically equivalent or not.

# GPT-2

"Language Models are Unsupervised Multitask Learners," 2018

# GPT-2



GPT-2 is a 1.5B parameter Transformer that achieves state of the art results on 7 out of 8 tested language modeling datasets in a **zero-shot setting**.

# Language-based Task Specification

- Language Prompt
  - Language provides a flexible way to specify **tasks**, **inputs**, and **outputs** all as a sequence of symbols.
  - Translation tasks
    - (translate to french, english text, french text)
  - Reading comprehension tasks
    - (answer the question, document, question, answer)

# WebText

- OpenAI created a new web scrape which emphasizes document quality.
  - Web pages which have been curated/filtered by humans are only used.
  - First, all outbound links from Reddit which received at least 3 karma are scraped.
  - The resulting **WebText** contains the text subset of these 45 million links.
    - It contains slightly over 8 million documents for a total of 40 GB of text.
    - All Wikipedia documents are removed.

# WebText

---

"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**'.

If listened carefully at 29:55, a conversation can be heard between two guys in French: "**-Comment on fait pour aller de l'autre côté? -Quel autre côté?**", which means "**- How do you get to the other side? - What side?**".

If this sounds like a bit of a stretch, consider this question in French: **As-tu aller au cinéma?**, or **Did you go to the movies?**, which literally translates as Have-you to go to movies/theater?

**"Brevet Sans Garantie Du Gouvernement"**, translated to English: **"Patented without government warranty"**.

---

*Table 1.* Examples of naturally occurring demonstrations of English to French and French to English translation found throughout the WebText training set.

# Byte Pair Encoding

- Word-level encoding
  - Construct a fixed-size dictionary based on word frequencies
  - Out-of-vocabulary (OOV) problem occurs
- Character-level encoding
  - Construct a dictionary with characters (e.g., lower-case Alphabet: 26, Korean: 11,172)
- Byte Pair Encoding (BPE)
  - BPE is a practical middle ground between character and word level language modeling
  - It is based on subword segmentation

```
#dictionary (word: frequency)
low : 5
lower : 2
newest : 6
widest : 3
# vocabulary
l, o, w, e, r, n, w, s, t, i, d
```

→

```
#dictionary (word: frequency)
l o w : 5
l o w e r : 2
n e w es t : 6
w i d es t : 3
# vocabulary
l, o, w, e, r, n, w, s, t, i, d, es
```

→

```
#dictionary (word: frequency)
l o w : 5
l o w e r : 2
n e w est : 6
w i d est : 3
# vocabulary
l, o, w, e, r, n, w, s, t, i, d, es, est
```

# Implementation Details

- Transformer-based architecture following the OpenAI GPT model
  - Layer normalization is moved to the input of each sub-block and an additional layer normalization was added after the final self-attention block
  - The vocabulary is expanded to 50,257. The context size is increased from 512 to 1,024 tokens and a batchsize is 512.

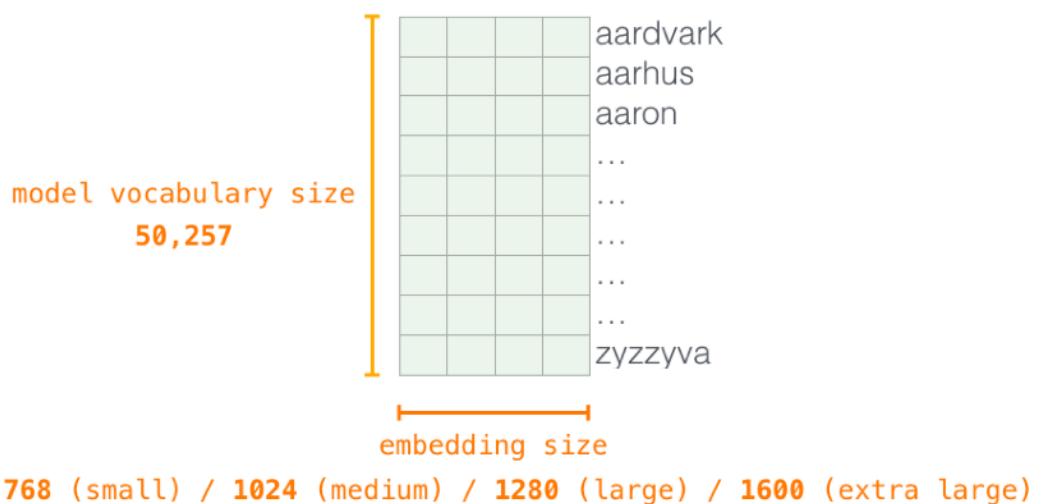
| Parameters | Layers | $d_{model}$ |   |
|------------|--------|-------------|---|
| 117M       | 12     | 768         | ← similar to the <a href="#">original GPT</a> |
| 345M       | 24     | 1024        | ← similar to the <a href="#">BERT</a>         |
| 762M       | 36     | 1280        |   |
| 1542M      | 48     | 1600        | ← this one is called <a href="#">GPT-2</a>    |

Table 2. Architecture hyperparameters for the 4 model sizes.

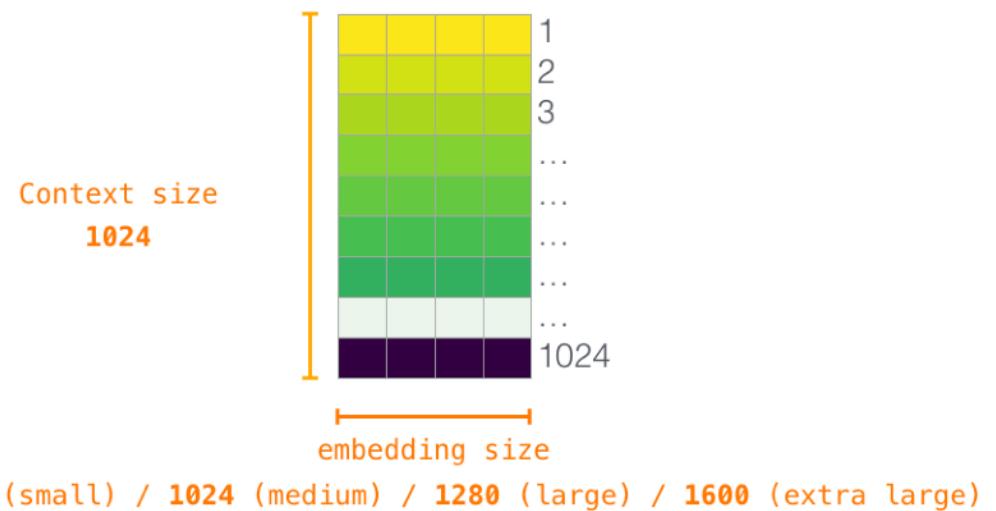
# Implementation Details



## Token Embeddings (wte)



## Positional Encodings (wpe)



# Experiments

**Language Models are Unsupervised Multitask Learners**

|       | LAMBADA<br>(PPL) | LAMBADA<br>(ACC) | CBT-CN<br>(ACC) | CBT-NE<br>(ACC) | WikiText2<br>(PPL) | PTB<br>(PPL) | enwik8<br>(BPB) | text8<br>(BPC) | WikiText103<br>(PPL) | 1BW<br>(PPL) |
|-------|------------------|------------------|-----------------|-----------------|--------------------|--------------|-----------------|----------------|----------------------|--------------|
| SOTA  | 99.8             | 59.23            | 85.7            | 82.3            | 39.14              | 46.54        | 0.99            | 1.08           | 18.3                 | <b>21.8</b>  |
| 117M  | <b>35.13</b>     | 45.99            | <b>87.65</b>    | <b>83.4</b>     | <b>29.41</b>       | 65.85        | 1.16            | 1.17           | 37.50                | 75.20        |
| 345M  | <b>15.60</b>     | 55.48            | <b>92.35</b>    | <b>87.1</b>     | <b>22.76</b>       | 47.33        | 1.01            | <b>1.06</b>    | 26.37                | 55.72        |
| 762M  | <b>10.87</b>     | <b>60.12</b>     | <b>93.45</b>    | <b>88.0</b>     | <b>19.93</b>       | <b>40.31</b> | <b>0.97</b>     | <b>1.02</b>    | 22.05                | 44.575       |
| 1542M | <b>8.63</b>      | <b>63.24</b>     | <b>93.30</b>    | <b>89.05</b>    | <b>18.34</b>       | <b>35.76</b> | <b>0.93</b>     | <b>0.98</b>    | <b>17.48</b>         | 42.16        |

*Table 3.* Zero-shot results on many datasets. No training or fine-tuning was performed for any of these results. PTB and WikiText-2 results are from (Gong et al., 2018). CBT results are from (Bajgar et al., 2016). LAMBADA accuracy result is from (Hoang et al., 2018) and LAMBADA perplexity result is from (Grave et al., 2016). Other results are from (Dai et al., 2019).

# Experiments

| Question  | Generated Answer      | Correct | Probability |
|---|-----------------------|---------|-------------|
| Who wrote the book the origin of species?   | Charles Darwin        | ✓       | 83.4%       |
| Who is the founder of the ubuntu project?   | Mark Shuttleworth     | ✓       | 82.0%       |
| Who is the quarterback for the green bay packers?                                 | Aaron Rodgers         | ✓       | 81.1%       |
| Panda is a national animal of which country?                                      | China                 | ✓       | 76.8%       |
| Who came up with the theory of relativity?  | Albert Einstein       | ✓       | 76.4%       |
| When was the first star wars film released?                                       | 1977                  | ✓       | 71.4%       |
| What is the most common blood type in sweden?                                     | A                     | ✗       | 70.6%       |
| Who is regarded as the founder of psychoanalysis?                                 | Sigmund Freud         | ✓       | 69.3%       |
| Who took the first steps on the moon in 1969?                                     | Neil Armstrong        | ✓       | 66.8%       |
| Who is the largest supermarket chain in the uk?                                   | Tesco                 | ✓       | 65.3%       |
| What is the meaning of shalom in english?   | peace                 | ✓       | 64.0%       |
| Who was the author of the art of war?   | Sun Tzu               | ✓       | 59.6%       |
| Largest state in the us by land mass?   | California            | ✗       | 59.2%       |
| Green algae is an example of which type of reproduction?                          | parthenogenesis       | ✗       | 56.5%       |
| Vikram samvat calender is official in which country?                              | India                 | ✓       | 55.6%       |
| Who is mostly responsible for writing the declaration of independence?            | Thomas Jefferson      | ✓       | 53.3%       |
| What us state forms the western boundary of montana?                              | Montana               | ✗       | 52.3%       |
| Who plays ser davos in game of thrones?   | Peter Dinklage        | ✗       | 52.1%       |
| Who appoints the chair of the federal reserve system?                             | Janet Yellen          | ✗       | 51.5%       |
| State the process that divides one nucleus into two genetically identical nuclei? | mitosis               | ✓       | 50.7%       |
| Who won the most mvp awards in the nba?   | Michael Jordan        | ✗       | 50.2%       |
| What river is associated with the city of rome?                                   | the Tiber             | ✓       | 48.6%       |
| Who is the first president to be impeached?                                       | Andrew Johnson        | ✓       | 48.3%       |
| Who is the head of the department of homeland security 2017?                      | John Kelly            | ✓       | 47.0%       |
| What is the name given to the common currency to the european union?              | Euro                  | ✓       | 46.8%       |
| What was the emperor name in star wars?   | Palpatine             | ✓       | 46.5%       |
| Do you have to have a gun permit to shoot at a range?                             | No                    | ✓       | 46.4%       |
| Who proposed evolution in 1859 as the basis of biological development?            | Charles Darwin        | ✓       | 45.7%       |
| Nuclear power plant that blew up in russia?                                       | Chernobyl             | ✓       | 45.7%       |
| Who played john connor in the original terminator?                                | Arnold Schwarzenegger | ✗       | 45.2%       |

Table 5. The 30 most confident answers generated by GPT-2 on the development set of Natural Questions sorted by their probability according to GPT-2. None of these questions appear in WebText according to the procedure described in Section 4.

GPT-2 answers 4.1% of questions correctly, which is much worse than 30-50% SOTA results.

# BERT

"BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,"  
2019

# BERT



Bidirectional Encoder Representation from Transformers (**BERT**)

# BERT



- **OpenAI GPT** uses left-to-right architecture, where every token can only attend to **previous** tokens in previous tokens in the self-attention layers of the Transformer.
- **BERT** alleviates the previously mentioned unidirectionality constraint by using a "**masked language model**" (MLN) pre-training objective.
  - The **masked language model** randomly masks some of the tokens from the input, and the objective is to predict the original vocabulary id of the masked word based only on its context.
  - In addition to the masked language model, a "**next sentence prediction**" task is also used that jointly pre-trains text-pair representations.

# BERT

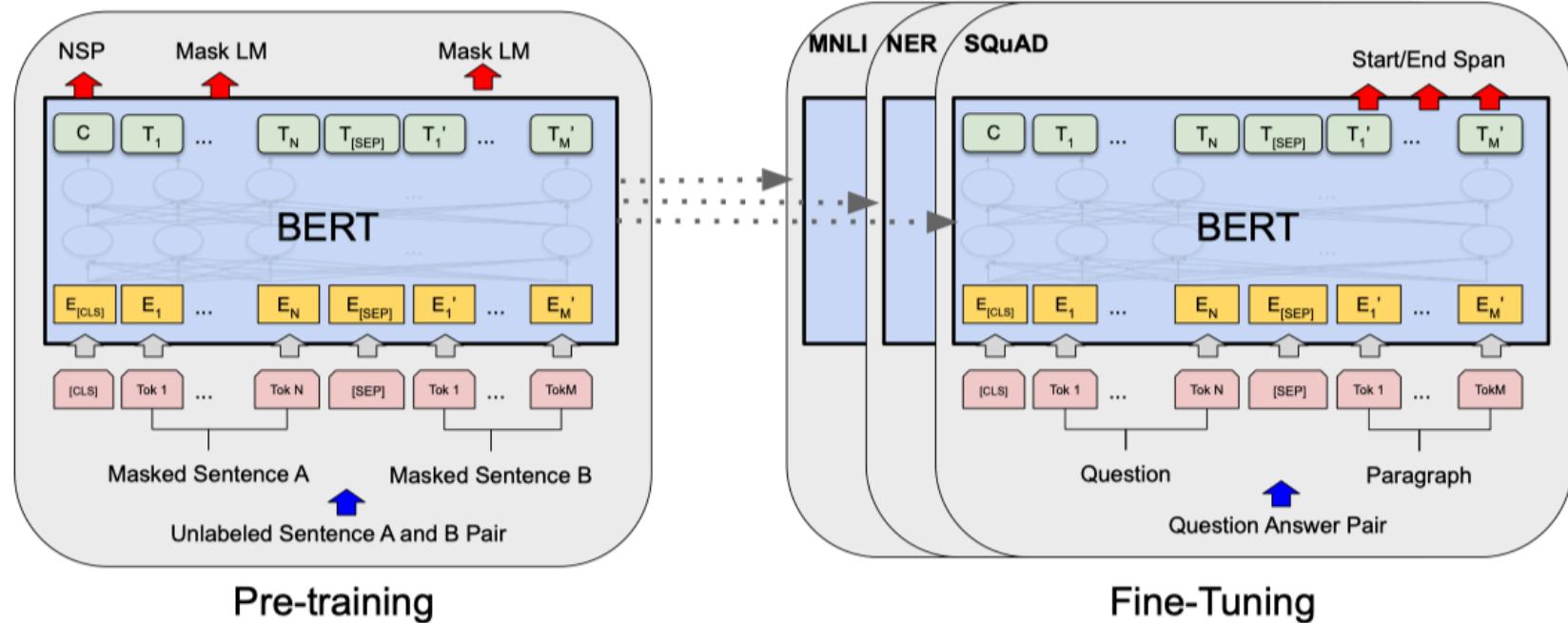


Figure 1: Overall pre-training and fine-tuning procedures for BERT. Apart from output layers, the same architectures are used in both pre-training and fine-tuning. The same pre-trained model parameters are used to initialize models for different down-stream tasks. During fine-tuning, all parameters are fine-tuned. [CLS] is a special symbol added in front of every input example, and [SEP] is a special separator token (e.g. separating questions/answers).

# Implementation Details

- BERT's model architecture is a multi-layer bidirectional **Transformer encoder**.
- Suppose that  $L$  is the number of layers,  $H$  is the hidden size, and  $A$  is the number of self-attention heads:
  - $\text{BERT}_{\text{BASE}}$ : ( $L = 12, H = 768, A = 12$ ) #param: 110M
  - $\text{BERT}_{\text{LARGE}}$ : ( $L = 24, H = 1024, A = 16$ ) #param: 340M
  - GPT-1: 125M, GPT-2: 1.5B, GPT-3: 175B
- Input/Output Representation
  - Both a single sentence and a pair of sentences (e.g., question and answer pairs) can be used as an input to BERT.
    - Sentence pairs are packed and they are separated with a special token, **[SEP]**. A learned embedding to every token is added to indicate whether it belongs to sentence A or sentence B.
  - WordPiece embeddings with a 30,000 token vocabulary are used.
  - The first token is always a special classification token, **[CLS]**. The final hidden state corresponding to this token is used as the aggregate sequence representation for classification tasks.

# Input Representation

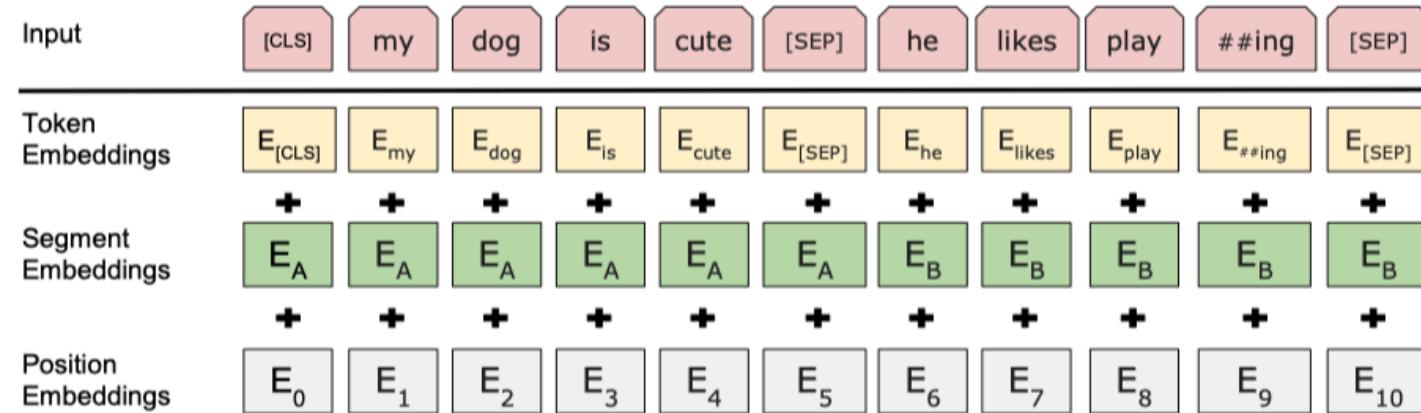


Figure 2: BERT input representation. The input embeddings are the sum of the token embeddings, the segmentation embeddings and the position embeddings.

# Pre-training BERT

- BERT is pre-trained with two unsupervised tasks: **masked language model** and **next sentence prediction**.
- Task 1: Masked LM
  - The masked words (15%) are only predicted rather than the entire input.
  - One downside is that it creates a mismatch between pre-training and fine-tuning phases, since the **[MASK]** token does not appear during fine-tuning.
  - To mitigate this issue, if the  $i$ -th token is chosen, we replace the  $i$ -th token with (1) the **[MASK]** token 80% of the time (2) a random token 10% of the time (3) the unchanged  $i$ -th token 10% of the time.
- Task 2: Next Sentence Prediction (NSP)
  - A binarized next sentence prediction tasks is used.
  - When choosing two sentences A and B, 50% of the time B is the actual next sentence that follows A and 50% of the time it is a random sentence from the corpus.

# Experiments

| System                | MNLI-(m/mm)<br>392k | QQP<br>363k | QNLI<br>108k | SST-2<br>67k | CoLA<br>8.5k | STS-B<br>5.7k | MRPC<br>3.5k | RTE<br>2.5k | Average<br>- |
|-----------------------|---------------------|-------------|--------------|--------------|--------------|---------------|--------------|-------------|--------------|
| Pre-OpenAI SOTA       | 80.6/80.1           | 66.1        | 82.3         | 93.2         | 35.0         | 81.0          | 86.0         | 61.7        | 74.0         |
| BiLSTM+ELMo+Attn      | 76.4/76.1           | 64.8        | 79.8         | 90.4         | 36.0         | 73.3          | 84.9         | 56.8        | 71.0         |
| OpenAI GPT            | 82.1/81.4           | 70.3        | 87.4         | 91.3         | 45.4         | 80.0          | 82.3         | 56.0        | 75.1         |
| BERT <sub>BASE</sub>  | 84.6/83.4           | 71.2        | 90.5         | 93.5         | 52.1         | 85.8          | 88.9         | 66.4        | 79.6         |
| BERT <sub>LARGE</sub> | <b>86.7/85.9</b>    | <b>72.1</b> | <b>92.7</b>  | <b>94.9</b>  | <b>60.5</b>  | <b>86.5</b>   | <b>89.3</b>  | <b>70.1</b> | <b>82.1</b>  |

Table 1: GLUE Test results, scored by the evaluation server (<https://gluebenchmark.com/leaderboard>). The number below each task denotes the number of training examples. The “Average” column is slightly different than the official GLUE score, since we exclude the problematic WNLI set.<sup>8</sup> BERT and OpenAI GPT are single-model, single task. F1 scores are reported for QQP and MRPC, Spearman correlations are reported for STS-B, and accuracy scores are reported for the other tasks. We exclude entries that use BERT as one of their components.

# GLUE



- General Language Understanding Evaluation (**GLUE**)
  - **CoLA**: The Corpus of Linguistic Acceptability: Binary classification: single sentences that are either grammatical or ungrammatical
  - **SST-2**: Stanford Sentiment Treebank: phrases culled from movie reviews scored on their positivity/negativity. Phrases can be positive, negative, or completely neutral
  - **STS-B**: The Semantic Textual Similarity Benchmark: task of determining the similarity on a continuous scale from 1 to 5 of a pair of sentences drawn from various sources
  - **QQP**: The Quora Question Pairs dataset: collection of question pairs from the community question-answering website Quora: Given two questions, the task is to determine whether they are semantically equivalent
- [https://docs.google.com/spreadsheets/d/1BrOdjJgky7FfeiwC\\_VDURZuRPUFUAz\\_jfczPPT35P00/edit#gid=0](https://docs.google.com/spreadsheets/d/1BrOdjJgky7FfeiwC_VDURZuRPUFUAz_jfczPPT35P00/edit#gid=0)

# RoBERTa

"RoBERTa: A Robustly Optimized BERT Pretraining Approach," 2019

# RoBERTa



Robustly optimized BERT approach (RoBERTa)

# RoBERTa

- This paper presents an improved recipe for training BERT models, named RoBERTa.
  - training the model longer, with bigger batches, over more data
  - removing the next sentence prediction objective
  - training on longer sequences
    - dynamically changing the masking pattern applied to the training data
- A large new dataset (CC-News) is collected.

# Dataset

- BERT-style pre-training crucially relies on large quantities of text. Five English-language corpora of varying sizes and domains, totaling over 160GB of uncompressed text, are used.
  - **BookCorpus plus English Wikipedia:** the original data used to train BERT (16GB)
  - **CC-News:** English portion of the CommonCrawl News dataset. The data contains 63 million English news articles crawled between Sep. 2016 and Feb. 2019 (76GB after filtering)
  - **OpenWebText:** The text is web content extracted from URLs shared on Reddit with at least three upvotes (38GB), Recreation of WebText corpora for training GPT-2
  - **Stories:** A subset of CommonCrawl filtered to match the story-like style of Winograd schemas (31GB)

# Implementation Details

- $\text{BERT}_{\text{BASE}}$  ( $L = 12, H = 768, A = 12, 110M$  params) configuration
  - Static vs. Dynamic Masking
    - The original BERT implementation performed masking once during data preprocessing, resulting in a single static mask.
    - Dynamic masking generates the masking pattern every time a sequence is fed to the model.

| <b>Masking</b>               | <b>SQuAD 2.0</b> | <b>MNLI-m</b> | <b>SST-2</b> |
|------------------------------|------------------|---------------|--------------|
| reference                    | 76.3             | 84.3          | 92.8         |
| <i>Our reimplementation:</i> |                  |               |              |
| static                       | 78.3             | 84.3          | 92.5         |
| dynamic                      | 78.7             | 84.0          | 92.9         |

Table 1: Comparison between static and dynamic masking for  $\text{BERT}_{\text{BASE}}$ . We report F1 for SQuAD and accuracy for MNLI-m and SST-2. Reported results are medians over 5 random initializations (seeds). Reference results are from [Yang et al. \(2019\)](#).

# Implementation Details

- $\text{BERT}_{\text{BASE}}$  ( $L = 12, H = 768, A = 12, 110M$  params) configuration
  - Nest Sentence Prediction (NSP)
    - The original BERT model emphasized the NSP objectives.
    - However, some recent work questioned the necessity of the NSP loss.

| Model   | SQuAD 1.1/2.0 | MNLI-m | SST-2 | RACE |
|---|---------------|--------|-------|------|
| <i>Our reimplementation (with NSP loss):</i>    |               |        |       |      |
| SEGMENT-PAIR                                    | 90.4/78.7     | 84.0   | 92.9  | 64.2 |
| SENTENCE-PAIR                                   | 88.7/76.2     | 82.9   | 92.1  | 63.0 |
| <i>Our reimplementation (without NSP loss):</i> |               |        |       |      |
| FULL-SENTENCES                                  | 90.4/79.1     | 84.7   | 92.5  | 64.8 |
| DOC-SENTENCES                                   | 90.6/79.7     | 84.7   | 92.7  | 65.6 |
| $\text{BERT}_{\text{BASE}}$                     | 88.5/76.3     | 84.3   | 92.8  | 64.3 |
| $\text{XLNet}_{\text{BASE}} (K=7)$              | -/81.3        | 85.8   | 92.7  | 66.1 |
| $\text{XLNet}_{\text{BASE}} (K=6)$              | -/81.0        | 85.6   | 93.4  | 66.7 |

Table 2: Development set results for base models pretrained over BOOKCORPUS and WIKIPEDIA. All models are trained for 1M steps with a batch size of 256 sequences. We report F1 for SQuAD and accuracy for MNLI-m, SST-2 and RACE. Reported results are medians over five random initializations (seeds). Results for  $\text{BERT}_{\text{BASE}}$  and  $\text{XLNet}_{\text{BASE}}$  are from [Yang et al. \(2019\)](#).

# Implementation Details

- $\text{BERT}_{\text{BASE}}$  ( $L = 12, H = 768, A = 12, 110M$  params) configuration
- Training with large batches
  - The original BERT trained  $\text{BERT}_{\text{BASE}}$  with 1M steps with a batch size of 256 sequences.
  - With the same computational cost, we can increase the batch size while reducing the steps.

| <b>bsz</b> | <b>steps</b> | <b>lr</b> | <b>ppl</b>  | <b>MNLI-m</b> | <b>SST-2</b> |
|------------|--------------|-----------|-------------|---------------|--------------|
| 256        | 1M           | 1e-4      | 3.99        | 84.7          | 92.7         |
| 2K         | 125K         | 7e-4      | <b>3.68</b> | <b>85.2</b>   | <b>92.9</b>  |
| 8K         | 31K          | 1e-3      | 3.77        | 84.6          | 92.8         |

Table 3: Perplexity on held-out training data (*ppl*) and development set accuracy for base models trained over BOOKCORPUS and WIKIPEDIA with varying batch sizes (*bsz*). We tune the learning rate (*lr*) for each setting. Models make the same number of passes over the data (epochs) and have the same computational cost.

# Experiments

| Model                        | data  | bsz | steps | SQuAD<br>(v1.1/2.0) | MNLI-m      | SST-2       |
|------------------------------|-------|-----|-------|---------------------|-------------|-------------|
| <b>RoBERTa</b>               |       |     |       |                     |             |             |
| with BOOKS + WIKI            | 16GB  | 8K  | 100K  | 93.6/87.3           | 89.0        | 95.3        |
| + additional data (§3.2)     | 160GB | 8K  | 100K  | 94.0/87.7           | 89.3        | 95.6        |
| + pretrain longer            | 160GB | 8K  | 300K  | 94.4/88.7           | 90.0        | 96.1        |
| + pretrain even longer       | 160GB | 8K  | 500K  | <b>94.6/89.4</b>    | <b>90.2</b> | <b>96.4</b> |
| <b>BERT<sub>LARGE</sub></b>  |       |     |       |                     |             |             |
| with BOOKS + WIKI            | 13GB  | 256 | 1M    | 90.9/81.8           | 86.6        | 93.7        |
| <b>XLNet<sub>LARGE</sub></b> |       |     |       |                     |             |             |
| with BOOKS + WIKI            | 13GB  | 256 | 1M    | 94.0/87.8           | 88.4        | 94.4        |
| + additional data            | 126GB | 2K  | 500K  | 94.5/88.8           | 89.8        | 95.6        |

Table 4: Development set results for RoBERTa as we pretrain over more data (16GB → 160GB of text) and pretrain for longer (100K → 300K → 500K steps). Each row accumulates improvements from the rows above. RoBERTa matches the architecture and training objective of BERT<sub>LARGE</sub>. Results for BERT<sub>LARGE</sub> and XLNet<sub>LARGE</sub> are from Devlin et al. (2019) and Yang et al. (2019), respectively. Complete results on all GLUE tasks can be found in the Appendix.

# Experiments

|   | <b>MNLI</b>      | <b>QNLI</b> | <b>QQP</b>  | <b>RTE</b>  | <b>SST</b>  | <b>MRPC</b> | <b>CoLA</b> | <b>STS</b>  | <b>WNLI</b> | <b>Avg</b>  |
|---|------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| <i>Single-task single models on dev</i>                         |                  |             |             |             |             |             |             |             |             |             |
| BERT <sub>LARGE</sub>   | 86.6/-           | 92.3        | 91.3        | 70.4        | 93.2        | 88.0        | 60.6        | 90.0        | -           | -           |
| XLNet <sub>LARGE</sub>  | 89.8/-           | 93.9        | 91.8        | 83.8        | 95.6        | 89.2        | 63.6        | 91.8        | -           | -           |
| RoBERTa   | <b>90.2/90.2</b> | <b>94.7</b> | <b>92.2</b> | <b>86.6</b> | <b>96.4</b> | <b>90.9</b> | <b>68.0</b> | <b>92.4</b> | <b>91.3</b> | -           |
| <i>Ensembles on test (from leaderboard as of July 25, 2019)</i> |                  |             |             |             |             |             |             |             |             |             |
| ALICE   | 88.2/87.9        | 95.7        | <b>90.7</b> | 83.5        | 95.2        | 92.6        | <b>68.6</b> | 91.1        | 80.8        | 86.3        |
| MT-DNN  | 87.9/87.4        | 96.0        | 89.9        | 86.3        | 96.5        | 92.7        | 68.4        | 91.1        | 89.0        | 87.6        |
| XLNet   | 90.2/89.8        | 98.6        | 90.3        | 86.3        | <b>96.8</b> | <b>93.0</b> | 67.8        | 91.6        | <b>90.4</b> | 88.4        |
| RoBERTa   | <b>90.8/90.2</b> | <b>98.9</b> | 90.2        | <b>88.2</b> | 96.7        | 92.3        | 67.8        | <b>92.2</b> | 89.0        | <b>88.5</b> |

Table 5: Results on GLUE. All results are based on a 24-layer architecture. BERT<sub>LARGE</sub> and XLNet<sub>LARGE</sub> results are from [Devlin et al. \(2019\)](#) and [Yang et al. \(2019\)](#), respectively. RoBERTa results on the development set are a median over five runs. RoBERTa results on the test set are ensembles of *single-task* models. For RTE, STS and MRPC we finetune starting from the MNLI model instead of the baseline pretrained model. Averages are obtained from the GLUE leaderboard.

# ALBERT

"ALBERT: A LITE BERT FOR SELF-SUPERVISED LEARNING OF LANGUAGE REPRESENTATIONS," 2020

# ALBERT



A Lite BERT ([ALBERT](#))

# ALBERT



- ALBERT presents two parameter-reduction techniques to lower memory consumption and increase the training speed of BERT.
  - The first one is a **factorized embedding parametrization** by decomposing the large vocabulary embedding matrix into two small matrices.
  - The second technique is **cross-layer parameter sharing** which prevents the parameter from growing with the depth of the network.
- A self-supervised loss focussing on modeling inter-sentence coherence is used.
  - A **sentence-order prediction** (SOP) focuses on inter-sentence coherence and is designed to address the ineffectiveness of the next sentence prediction (NSP) loss proposed in the original BERT.

# ALBERT



- Factorized embedding parametrization
  - The size of the embedding matrix has the size of  $V \times H$  where  $V$  is the vocabulary size and  $H$  is the size of hidden space.
  - By projecting them into a lower dimensional embedding space of size  $E$ , the embedding parameters are reduced from  $O(V \times H)$  to  $O(V \times E + E \times H)$ . This parameter reduction is significant when  $H \gg E$ .
- Cross-layer parameter sharing
  - There are multiple ways to share parameters (e.g., only sharing feed-forward network parameters across layers, or only sharing attention parameters, or all parameters which is the default decision for ALBERT).
- Inter-sentence coherence loss
  - The sentence-order-prediction (SOP) task uses as positive examples the same as BERT and as negative examples the same two consecutive segments but with their order swapped.

# Model

- ALBERT-large has about 18x fewer parameters compare to BERT-large, 18M versus 334M.
- An ALBERT-xlarge configuration with  $H = 2048$  has 233M parameters (i.e., around 70% of BERT)

|        | Model   | Parameters | Layers | Hidden | Embedding | Parameter-sharing |
|--------|---------|------------|--------|--------|-----------|-------------------|
| BERT   | base    | 108M       | 12     | 768    | 768       | False             |
|        | large   | 334M       | 24     | 1024   | 1024      | False             |
| ALBERT | base    | 12M        | 12     | 768    | 128       | True              |
|        | large   | 18M        | 24     | 1024   | 128       | True              |
|        | xlarge  | 60M        | 24     | 2048   | 128       | True              |
|        | xxlarge | 235M       | 12     | 4096   | 128       | True              |

Table 1: The configurations of the main BERT and ALBERT models analyzed in this paper.

# Experiments

|        | Model   | Parameters | SQuAD1.1         | SQuAD2.0         | MNLI        | SST-2       | RACE        | Avg         | Speedup |
|--------|---------|------------|------------------|------------------|-------------|-------------|-------------|-------------|---------|
| BERT   | base    | 108M       | 90.4/83.2        | 80.4/77.6        | 84.5        | 92.8        | 68.2        | 82.3        | 4.7x    |
|        | large   | 334M       | 92.2/85.5        | 85.0/82.2        | 86.6        | 93.0        | 73.9        | 85.2        | 1.0     |
| ALBERT | base    | 12M        | 89.3/82.3        | 80.0/77.1        | 81.6        | 90.3        | 64.0        | 80.1        | 5.6x    |
|        | large   | 18M        | 90.6/83.9        | 82.3/79.4        | 83.5        | 91.7        | 68.5        | 82.4        | 1.7x    |
|        | xlarge  | 60M        | 92.5/86.1        | 86.1/83.1        | 86.4        | 92.4        | 74.8        | 85.5        | 0.6x    |
|        | xxlarge | 235M       | <b>94.1/88.3</b> | <b>88.1/85.1</b> | <b>88.0</b> | <b>95.2</b> | <b>82.3</b> | <b>88.7</b> | 0.3x    |

Table 2: Dev set results for models pretrained over BOOKCORPUS and Wikipedia for 125k steps. Here and everywhere else, the Avg column is computed by averaging the scores of the downstream tasks to its left (the two numbers of F1 and EM for each SQuAD are first averaged).

# Experiments

|                                | Model            | Parameters | SQuAD1.1  | SQuAD2.0  | MNLI | SST-2 | RACE | Avg  |
|--------------------------------|------------------|------------|-----------|-----------|------|-------|------|------|
| ALBERT<br>base<br><i>E=768</i> | all-shared       | 31M        | 88.6/81.5 | 79.2/76.6 | 82.0 | 90.6  | 63.3 | 79.8 |
|                                | shared-attention | 83M        | 89.9/82.7 | 80.0/77.2 | 84.0 | 91.4  | 67.7 | 81.6 |
|                                | shared-FFN       | 57M        | 89.2/82.1 | 78.2/75.4 | 81.5 | 90.8  | 62.6 | 79.5 |
|                                | not-shared       | 108M       | 90.4/83.2 | 80.4/77.6 | 84.5 | 92.8  | 68.2 | 82.3 |
| ALBERT<br>base<br><i>E=128</i> | all-shared       | 12M        | 89.3/82.3 | 80.0/77.1 | 82.0 | 90.3  | 64.0 | 80.1 |
|                                | shared-attention | 64M        | 89.9/82.8 | 80.7/77.9 | 83.4 | 91.9  | 67.6 | 81.7 |
|                                | shared-FFN       | 38M        | 88.9/81.6 | 78.6/75.6 | 82.3 | 91.7  | 64.4 | 80.2 |
|                                | not-shared       | 89M        | 89.9/82.8 | 80.3/77.3 | 83.2 | 91.5  | 67.9 | 81.6 |

Table 4: The effect of cross-layer parameter-sharing strategies, ALBERT-base configuration.

# Experiments

| SP tasks | Intrinsic Tasks |      |      | Downstream Tasks |                  |             |             |             | Avg         |
|----------|-----------------|------|------|------------------|------------------|-------------|-------------|-------------|-------------|
|          | MLM             | NSP  | SOP  | SQuAD1.1         | SQuAD2.0         | MNLI        | SST-2       | RACE        |             |
| None     | 54.9            | 52.4 | 53.3 | 88.6/81.5        | 78.1/75.3        | 81.5        | 89.9        | 61.7        | 79.0        |
| NSP      | 54.5            | 90.5 | 52.0 | 88.4/81.5        | 77.2/74.6        | 81.6        | <b>91.1</b> | 62.3        | 79.2        |
| SOP      | 54.0            | 78.9 | 86.5 | <b>89.3/82.3</b> | <b>80.0/77.1</b> | <b>82.0</b> | 90.3        | <b>64.0</b> | <b>80.1</b> |

Table 5: The effect of sentence-prediction loss, NSP vs. SOP, on intrinsic and downstream tasks.

| Models         | Steps | Time | SQuAD1.1         | SQuAD2.0         | MNLI | SST-2       | RACE        | Avg         |
|----------------|-------|------|------------------|------------------|------|-------------|-------------|-------------|
| BERT-large     | 400k  | 34h  | 93.5/87.4        | 86.9/84.3        | 87.8 | 94.6        | 77.3        | 87.2        |
| ALBERT-xxlarge | 125k  | 32h  | <b>94.0/88.1</b> | <b>88.3/85.3</b> | 87.8 | <b>95.4</b> | <b>82.5</b> | <b>88.7</b> |

Table 6: The effect of controlling for training time, BERT-large vs ALBERT-xxlarge configurations

# Experiments

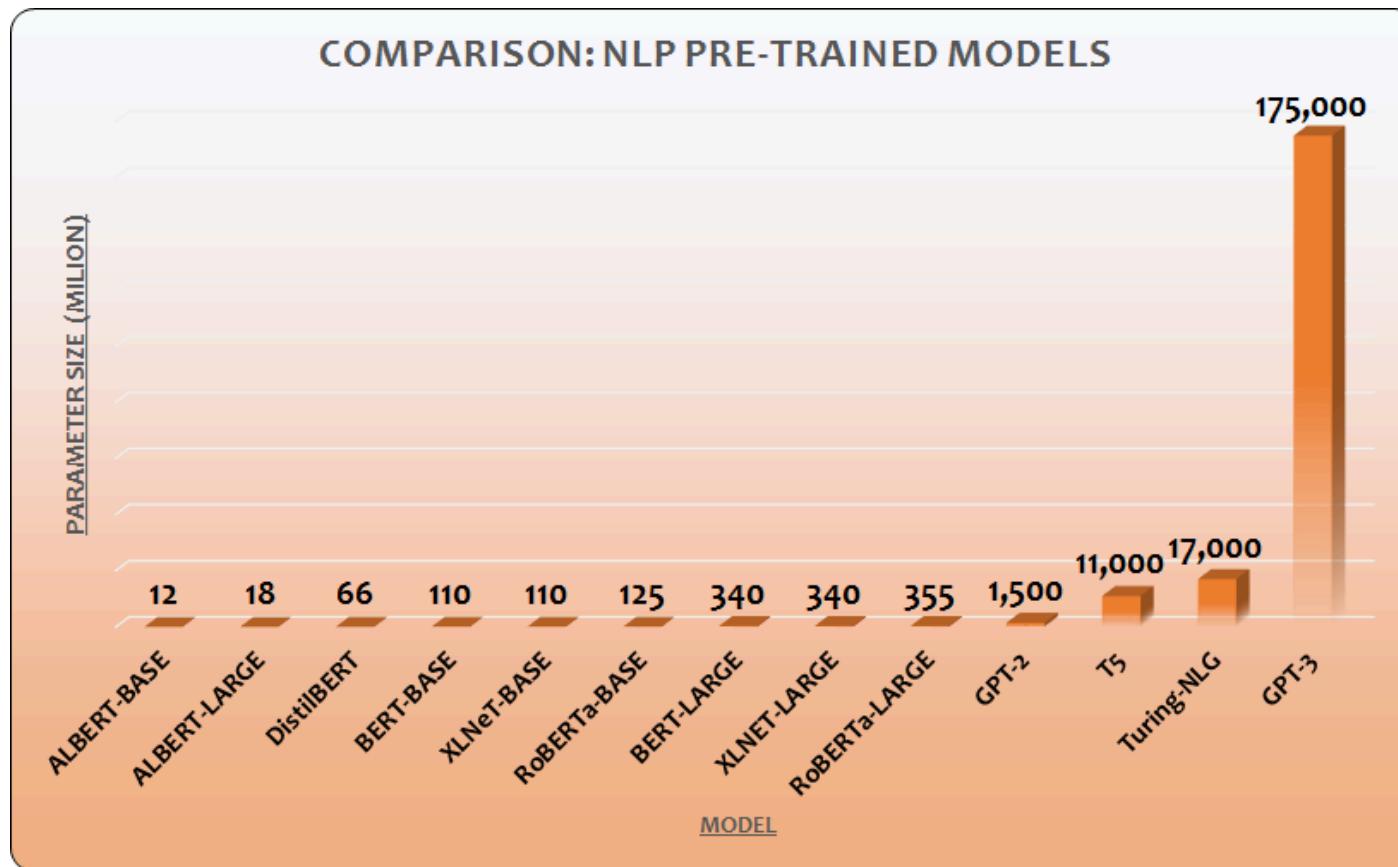
| Models   | MNLI        | QNLI        | QQP         | RTE         | SST         | MRPC        | CoLA        | STS         | WNLI        | Avg         |
|--|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| <i>Single-task single models on dev</i>                          |             |             |             |             |             |             |             |             |             |             |
| BERT-large   | 86.6        | 92.3        | 91.3        | 70.4        | 93.2        | 88.0        | 60.6        | 90.0        | -           | -           |
| XLNet-large  | 89.8        | 93.9        | 91.8        | 83.8        | 95.6        | 89.2        | 63.6        | 91.8        | -           | -           |
| RoBERTa-large  | 90.2        | 94.7        | <b>92.2</b> | 86.6        | 96.4        | <b>90.9</b> | 68.0        | 92.4        | -           | -           |
| ALBERT (1M)  | 90.4        | 95.2        | 92.0        | 88.1        | 96.8        | 90.2        | 68.7        | 92.7        | -           | -           |
| ALBERT (1.5M)  | <b>90.8</b> | <b>95.3</b> | <b>92.2</b> | <b>89.2</b> | <b>96.9</b> | <b>90.9</b> | <b>71.4</b> | <b>93.0</b> | -           | -           |
| <i>Ensembles on test (from leaderboard as of Sept. 16, 2019)</i> |             |             |             |             |             |             |             |             |             |             |
| ALICE  | 88.2        | 95.7        | <b>90.7</b> | 83.5        | 95.2        | 92.6        | <b>69.2</b> | 91.1        | 80.8        | 87.0        |
| MT-DNN   | 87.9        | 96.0        | 89.9        | 86.3        | 96.5        | 92.7        | 68.4        | 91.1        | 89.0        | 87.6        |
| XLNet  | 90.2        | 98.6        | 90.3        | 86.3        | 96.8        | 93.0        | 67.8        | 91.6        | 90.4        | 88.4        |
| RoBERTa  | 90.8        | 98.9        | 90.2        | 88.2        | 96.7        | 92.3        | 67.8        | 92.2        | 89.0        | 88.5        |
| Adv-RoBERTa  | 91.1        | 98.8        | 90.3        | 88.7        | 96.8        | 93.1        | 68.0        | 92.4        | 89.0        | 88.8        |
| ALBERT   | <b>91.3</b> | <b>99.2</b> | 90.5        | <b>89.2</b> | <b>97.1</b> | <b>93.4</b> | 69.1        | <b>92.5</b> | <b>91.8</b> | <b>89.4</b> |

Table 9: State-of-the-art results on the GLUE benchmark. For single-task single-model results, we report ALBERT at 1M steps (comparable to RoBERTa) and at 1.5M steps. The ALBERT ensemble uses models trained with 1M, 1.5M, and other numbers of steps.

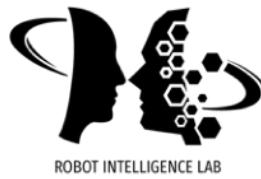
# GPT-3

"Language Models are Few-Shot Learners," 2020

# GPT-3



# GPT-3



**What is new** besides the model size has increased over 10 times?

# Datasets

| 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  |

**Table 2.2: Datasets used to train GPT-3.** “Weight in training mix” refers to the fraction of examples during training that are drawn from a given dataset, which we intentionally do not make proportional to the size of the dataset. As a result, when we train for 300 billion tokens, some datasets are seen up to 3.4 times during training while other datasets are seen less than once.

# Dataset

- Three different techniques are used (1) **filtering CommonCrawl**, (2) fuzzy deduplication, (3) added high-quality reference corpora to the training mix to augment CommonCrawl
  - An **automatic filtering method** is developed to remove low quality documents. A classifier is trained by using the original WebText as a proxy for high-quality documents to distinguish low quality ones from raw CommonCrawl.
  - The classifier (logistic regression) is used to re-sample CommonCrawl by prioritizing documents which were predicted by the classifier to be higher quality. The positive examples include curated datasets such as **WebText**, **Wikipedia**, and **web books corpus**, and the negative examples include **unfiltered Common Crawl**.

# Dataset

- Three different techniques are used (1) filtering CommonCrawl, (2) **fuzzy deduplication**, (3) added high-quality reference corpora to the training mix to augment CommonCrawl
  - To improve the model quality and prevent overfitting, the authors fuzzily deduplicated documents (i.e., removed documents with high overlap with other documents). WebText was fuzzily removed from Common Crawl.
  - Overall, this decreases dataset size by an average of 10%.

# In-context Learning

## Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.



## One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.



## Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.



# Implementation Details

| Model Name            | $n_{\text{params}}$ | $n_{\text{layers}}$ | $d_{\text{model}}$ | $n_{\text{heads}}$ | $d_{\text{head}}$ | Batch Size | Learning Rate        |
|-----------------------|---------------------|---------------------|--------------------|--------------------|-------------------|------------|----------------------|
| GPT-3 Small           | 125M                | 12                  | 768                | 12                 | 64                | 0.5M       | $6.0 \times 10^{-4}$ |
| GPT-3 Medium          | 350M                | 24                  | 1024               | 16                 | 64                | 0.5M       | $3.0 \times 10^{-4}$ |
| GPT-3 Large           | 760M                | 24                  | 1536               | 16                 | 96                | 0.5M       | $2.5 \times 10^{-4}$ |
| GPT-3 XL              | 1.3B                | 24                  | 2048               | 24                 | 128               | 1M         | $2.0 \times 10^{-4}$ |
| GPT-3 2.7B            | 2.7B                | 32                  | 2560               | 32                 | 80                | 1M         | $1.6 \times 10^{-4}$ |
| GPT-3 6.7B            | 6.7B                | 32                  | 4096               | 32                 | 128               | 2M         | $1.2 \times 10^{-4}$ |
| GPT-3 13B             | 13.0B               | 40                  | 5140               | 40                 | 128               | 2M         | $1.0 \times 10^{-4}$ |
| GPT-3 175B or “GPT-3” | 175.0B              | 96                  | 12288              | 96                 | 128               | 3.2M       | $0.6 \times 10^{-4}$ |

**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.

# Experiments

Title: United Methodists Agree to Historic Split

Subtitle: Those who oppose gay marriage will form their own denomination

Article: After two days of intense debate, the United Methodist Church has agreed to a historic split - one that is expected to end in the creation of a new denomination, one that will be "theologically and socially conservative," according to The Washington Post. The majority of delegates attending the church's annual General Conference in May voted to strengthen a ban on the ordination of LGBTQ clergy and to write new rules that will "discipline" clergy who officiate at same-sex weddings. But those who opposed these measures have a new plan: They say they will form a separate denomination by 2020, calling their church the Christian Methodist denomination.

The Post notes that the denomination, which claims 12.5 million members, was in the early 20th century the "largest Protestant denomination in the U.S.," but that it has been shrinking in recent decades. The new split will be the second in the church's history. The first occurred in 1968, when roughly 10 percent of the denomination left to form the Evangelical United Brethren Church. The Post notes that the proposed split "comes at a critical time for the church, which has been losing members for years," which has been "pushed toward the brink of a schism over the role of LGBTQ people in the church." Gay marriage is not the only issue that has divided the church. In 2016, the denomination was split over ordination of transgender clergy, with the North Pacific regional conference voting to ban them from serving as clergy, and the South Pacific regional conference voting to allow them.

**Figure 3.14:** The GPT-3 generated news article that humans had the greatest difficulty distinguishing from a human written article (accuracy: 12%).

# Experiments

A "whatpu" is a small, furry animal native to Tanzania. An example of a sentence that uses the word whatpu is:

We were traveling in Africa and we saw these very cute whatpus.

---

To do a "farduddle" means to jump up and down really fast. An example of a sentence that uses the word farduddle is:

**One day when I was playing tag with my little sister, she got really excited and she started doing these crazy farduddles.**

A "yalubalu" is a type of vegetable that looks like a big pumpkin. An example of a sentence that uses the word yalubalu is:

**I was on a trip to Africa and I tried this yalubalu vegetable that was grown in a garden there. It was delicious.**

---

A "Burringo" is a car with very fast acceleration. An example of a sentence that uses the word Burringo is:

**In our garage we have a Burringo that my father drives to work every day.**

---

A "Gigamuru" is a type of Japanese musical instrument. An example of a sentence that uses the word Gigamuru is:

**I have a Gigamuru that my uncle gave me as a gift. I love to play it at home.**

---

To "screeg" something is to swing a sword at it. An example of a sentence that uses the word screeg is:

**We screeghed at each other for several minutes and then we went outside and ate ice cream.**

**Figure 3.16:** Representative GPT-3 completions for the few-shot task of using a new word in a sentence. Boldface is GPT-3's completions, plain text is human prompts. In the first example both the prompt and the completion are provided by a human; this then serves as conditioning for subsequent examples where GPT-3 receives successive additional prompts and provides the completions. Nothing task-specific is provided to GPT-3 other than the conditioning shown here.

# Experiments

|                      |  |
|----------------------|--|
| Poor English input:  | I eated the purple berries.  |
| Good English output: | <b>I ate the purple berries.</b>   |
| Poor English input:  | Thank you for picking me as your designer. I'd appreciate it.  |
| Good English output: | <b>Thank you for choosing me as your designer. I appreciate it.</b>  |
| Poor English input:  | The mentioned changes have done. or I did the alteration that you requested. or I changed things you wanted and did the modifications.                                       |
| Good English output: | <b>The requested changes have been made. or I made the alteration that you requested. or I changed things you wanted and made the modifications.</b>                         |
| Poor English input:  | I'd be more than happy to work with you in another project.  |
| Good English output: | <b>I'd be more than happy to work with you on another project.</b>   |
| <hr/>                |  |
| Poor English input:  | Please provide me with a short brief of the design you're looking for and that'd be nice if you could share some examples or project you did before.                         |
| Good English output: | <b>Please provide me with a brief description of the design you're looking for and that would be nice if you could share some examples or projects you have done before.</b> |
| <hr/>                |  |
| Poor English input:  | The patient was died.  |
| Good English output: | <b>The patient died.</b>   |
| <hr/>                |  |
| Poor English input:  | We think that Leslie likes ourselves.  |
| Good English output: | <b>We think that Leslie likes us.</b>  |
| <hr/>                |  |
| Poor English input:  | Janet broke Bill on the finger.  |
| Good English output: | <b>Janet broke Bill's finger.</b>  |
| <hr/>                |  |
| Poor English input:  | Mary arranged for, in St. Louis, John to rent a house cheap.   |
| Good English output: | <b>Mary arranged for John to rent a house in St. Louis.</b>  |
| <hr/>                |  |
| Poor English input:  | Today I have went to the store to to buys some many bottle of water.   |
| Good English output: | <b>Today I went to the store to buy some bottles of water.</b>   |
| <hr/>                |  |
| Poor English input:  | I have tried to hit ball with bat, but my swing is has miss.   |
| Good English output: | <b>I tried to hit the ball with the bat, but my swing missed.</b>  |

**Figure 3.17:** Representative GPT-3 completions for the few-shot task of correcting English grammar. Boldface is GPT-3's completions, plain text is human prompts. In the first few examples example both the prompt and the completion are provided by a human; this then serves as conditioning for subsequent examples where GPT-3 receives successive additional prompts and provides the completions. Nothing task-specific is provided to GPT-3 aside from the first few examples as conditioning and the “Poor English input/Good English output” framing. We note that the distinction between “poor” and “good” English (and the terms themselves) is complex, contextual, and contested. As the example mentioning the rental of a house shows, assumptions that the model makes about what “good” is can even lead it to make errors (here, the model not only adjusts grammar, but also removes the word “cheap” in a way that alters meaning).

# InstructGPT

"Training language models to follow instructions with human feedback," 2022

# InstructGPT

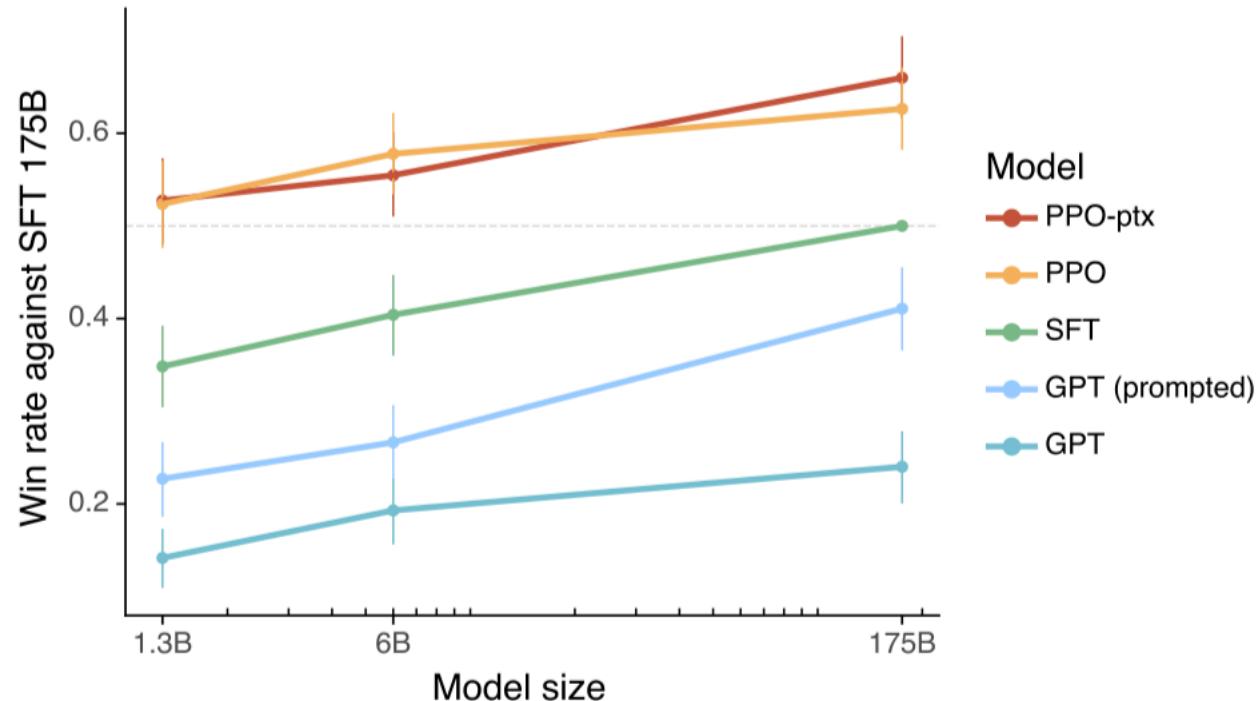


Figure 1: Human evaluations of various models on our API prompt distribution, evaluated by how often outputs from each model were preferred to those from the 175B SFT model. Our InstructGPT models (PPO-ptx) as well as its variant trained without pretraining mix (PPO) significantly outperform the GPT-3 baselines (GPT, GPT prompted); outputs from our 1.3B PPO-ptx model are preferred to those from the 175B GPT-3. Error bars throughout the paper are 95% confidence intervals.

# Alignment



"We make progress on **aligning language models** by training them  
to act in accordance with the user's intention"

# High-level Methodology

Step 1

**Collect demonstration data, and train a supervised policy.**

A prompt is sampled from our prompt dataset.

Explain the moon landing to a 6 year old

A labeler demonstrates the desired output behavior.

Some people went to the moon...

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

SFT

Step 2

**Collect comparison data, and train a reward model.**

A prompt and several model outputs are sampled.

Explain the moon landing to a 6 year old

A B  
Explain gravity... Explain war...  
C D  
Moon is natural satellite of... People went to the moon...

A labeler ranks the outputs from best to worst.

D > C > A = B

This data is used to train our reward model.

RM  
D > C > A = B

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.

Frog  
Write a story about frogs

PPO

Once upon a time...

RM

$r_k$

# InstructGPT



- InstructGPT starts with the GPT-3 pre-trained language model.
  - Supervised fine-tuning (**SFT**)
    - The prompt dataset consists of text prompts submitted to the OpenAI API.
    - Human labeler demonstrations are used to fine-tune GPT-3 to get **SFT models**.
  - Reward modeling (**RM**)
    - The comparison dataset is collected by comparing two different outputs of models.
    - The **reward function** is trained to predict human preferences.
  - Reinforcement Learning from Human Feedback (RLHF)
    - **SFT models** are fine-tuned using the **reward function**.
      - In particular, the model (**InstructGPT**) completes the given prompt, and the **reward model** outputs the estimated reward.
      - The model is fine-tuned to maximize the estimated reward.

# InstructGPT vs. SFT model

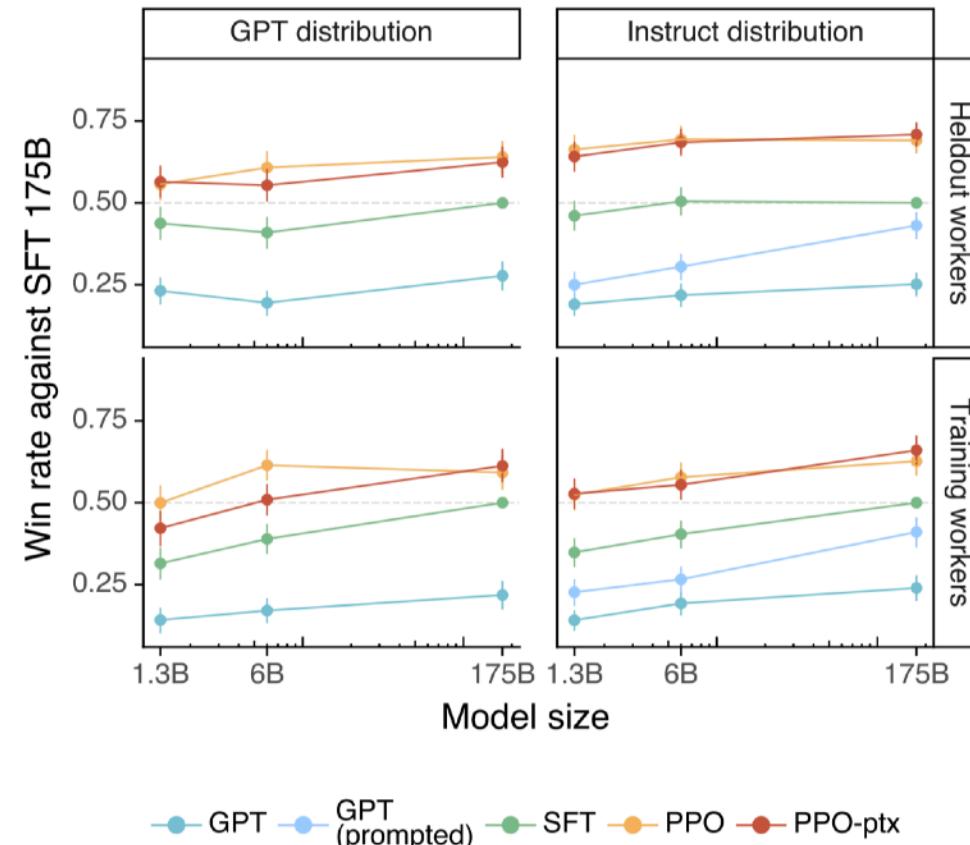
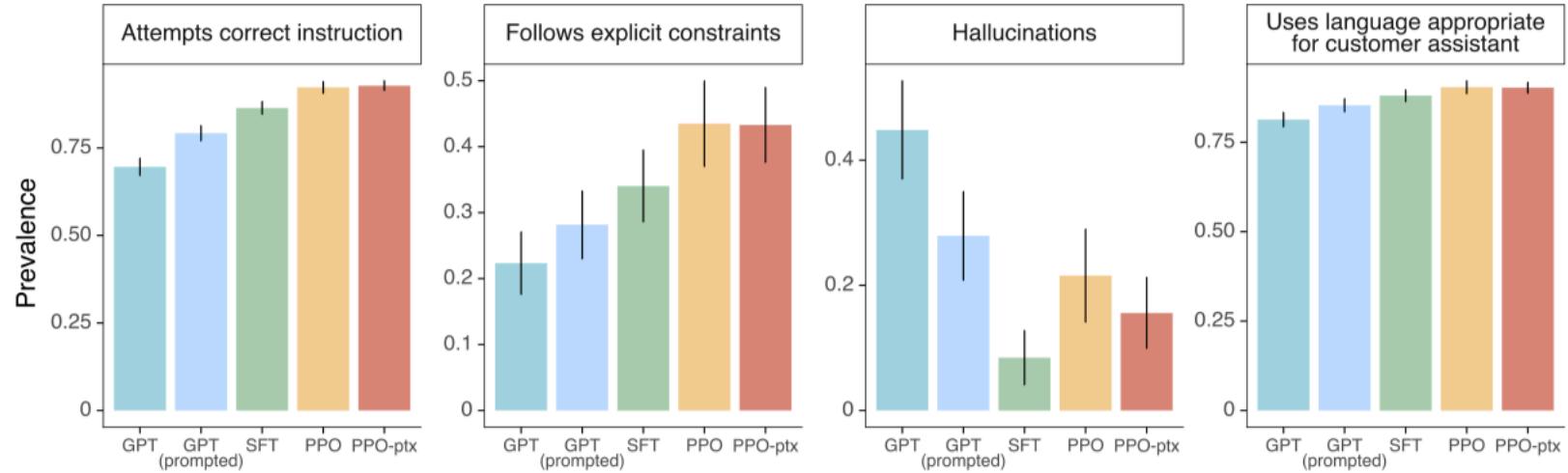


Figure 3: Preference results of our models, measured by winrate against the 175B SFT model. Left: results on prompts submitted to GPT models on the API; Right: results on prompts submitted to InstructGPT models on the API; Top: results from held-out labelers; Bottom: results from training labelers. We omit GPT (prompted) from the evals on prompts submitted to GPT-3 models (left) as these prompts are already designed to perform well for GPT-3, as opposed to prompts submitted to InstructGPT models (right).

# Results

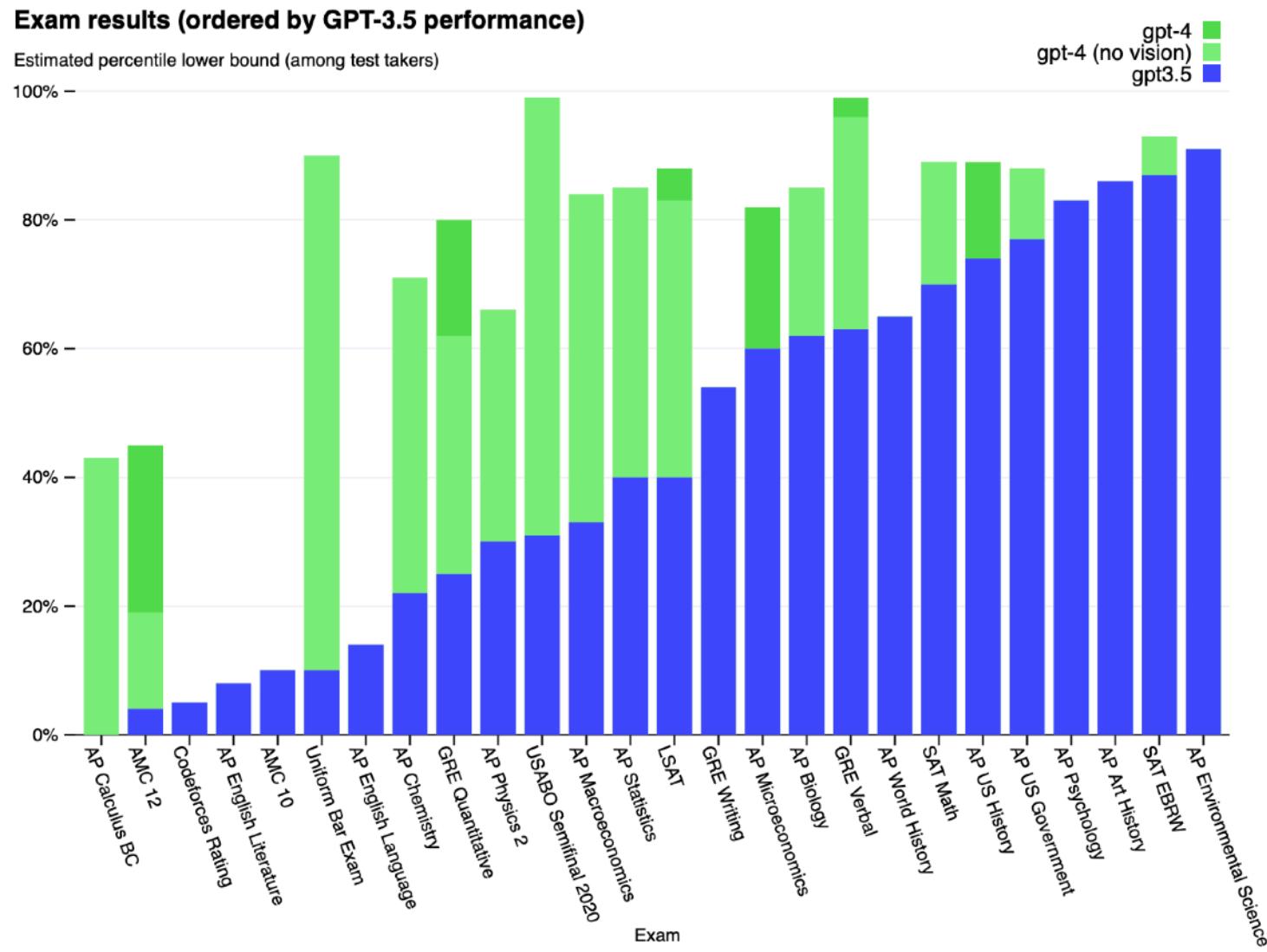


**Figure 4: Metadata results on the API distribution.** Note that, due to dataset sizes, these results are collapsed across model sizes. See Appendix E.2 for analysis that includes model size. Compared to GPT-3, the PPO models are more appropriate in the context of a customer assistant, are better at following explicit constraints in the instruction and attempting the correct instruction, and less likely to ‘hallucinate’ (meaning, making up information on closed domain tasks like summarization).

# GPT-4

<https://openai.com/research/gpt-4>

# Exam Results

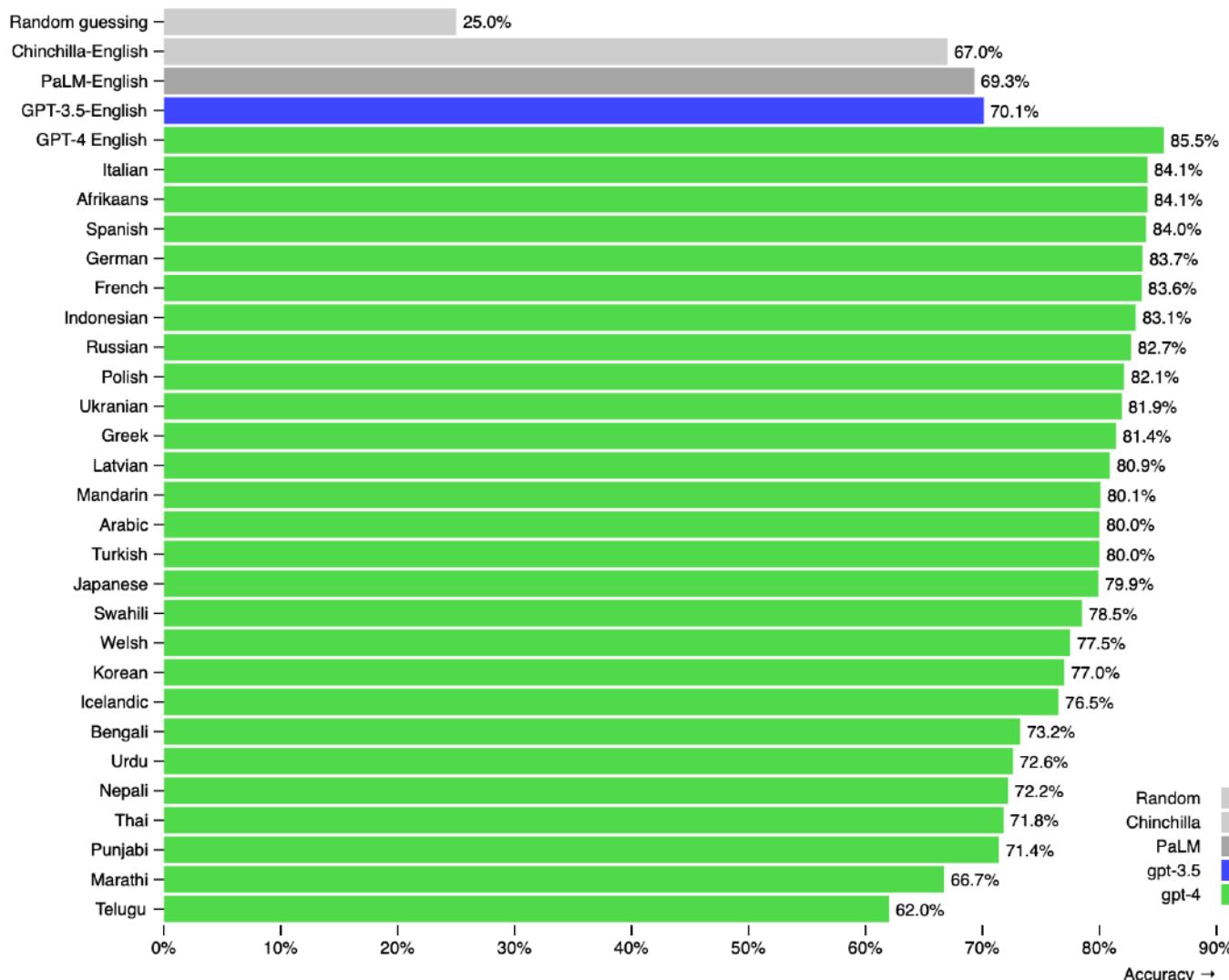


# Traditional ML Benchmarks

| Benchmark  | GPT-4<br>Evaluated few-shot | GPT-3.5<br>Evaluated few-shot | LM SOTA<br>Best external LM evaluated few-shot | SOTA<br>Best external model (includes benchmark-specific training) |
|--|-----------------------------|-------------------------------|--|--|
| <b>MMLU</b><br>Multiple-choice questions in 57 subjects (professional & academic)                      | <b>86.4%</b><br>5-shot      | <b>70.0%</b><br>5-shot        | <b>70.7%</b><br><u>5-shot U-PaLM</u>           | <b>75.2%</b><br><u>5-shot Flan-PaLM</u>                            |
| <b>HellaSwag</b><br>Commonsense reasoning around everyday events                                       | <b>95.3%</b><br>10-shot     | <b>85.5%</b><br>10-shot       | <b>84.2%</b><br><u>LLAMA (validation set)</u>  | <b>85.6%</b><br><u>ALUM</u>  |
| <b>AI2 Reasoning Challenge (ARC)</b><br>Grade-school multiple choice science questions. Challenge-set. | <b>96.3%</b><br>25-shot     | <b>85.2%</b><br>25-shot       | <b>84.2%</b><br><u>8-shot PaLM</u>             | <b>85.6%</b><br><u>ST-MOE</u>                                      |
| <b>WinoGrande</b><br>Commonsense reasoning around pronoun resolution                                   | <b>87.5%</b><br>5-shot      | <b>81.6%</b><br>5-shot        | <b>84.2%</b><br><u>5-shot PALM</u>             | <b>85.6%</b><br><u>5-shot PALM</u>                                 |
| <b>HumanEval</b><br>Python coding tasks  | <b>67.0%</b><br>0-shot      | <b>48.1%</b><br>0-shot        | <b>26.2%</b><br><u>0-shot PaLM</u>             | <b>65.8%</b><br><u>CodeT + GPT-3.5</u>                             |
| <b>DROP (f1 score)</b><br>Reading comprehension & arithmetic.  | <b>80.9</b><br>3-shot       | <b>64.1</b><br>3-shot         | <b>70.8</b><br><u>1-shot PaLM</u>              | <b>88.4</b><br><u>QDGAT</u>  |

# Across Languages

GPT-4 3-shot accuracy on MMLU across languages



# Visual Inputs

User What is funny about this image? Describe it panel by panel.



Source: [hmmm \(Reddit\)](#)

GPT-4 The image shows a package for a "Lightning Cable" adapter with three panels.

Panel 1: A smartphone with a VGA connector (a large, blue, 15-pin connector typically used for computer monitors) plugged into its charging port.

Panel 2: The package for the "Lightning Cable" adapter with a picture of a VGA connector on it.

Panel 3: A close-up of the VGA connector with a small Lightning connector (used for charging iPhones and other Apple devices) at the end.

The humor in this image comes from the absurdity of plugging a large, outdated VGA connector into a small, modern smartphone charging port.

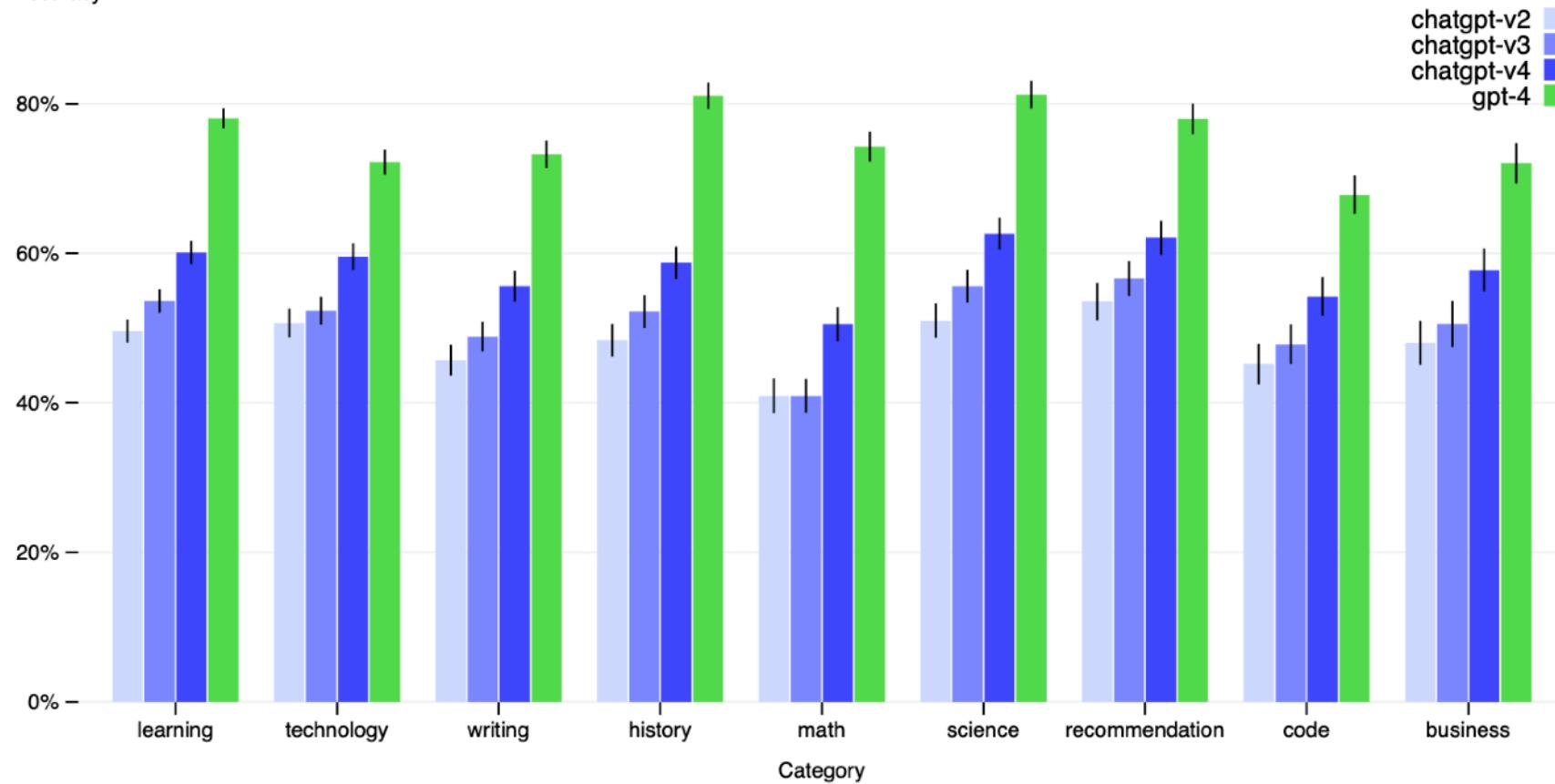
# Academic Vision Benchmarks

| Benchmark   | GPT-4                               | Few-shot SOTA                                | SOTA   |
|---|-------------------------------------|--|--|
|   | Evaluated few-shot                  |  | Best external model (includes benchmark-specific training) |
| <u>VQAv2</u><br>VQA score (test-dev)              | <b>77.2%</b><br>0-shot              | <b>67.6%</b><br>Flamingo 32-shot             | <b>84.3%</b><br><u>PaLI-17B</u>                            |
| <u>TextVQA</u><br>VQA score (val)                 | <b>78.0%</b><br>0-shot              | <b>37.9%</b><br>Flamingo 32-shot             | <b>71.8%</b><br><u>PaLI-17B</u>                            |
| <u>ChartQA</u><br>Relaxed accuracy (test)         | <b>78.5%<sup>A</sup></b>            | -  | <b>58.6%</b><br><u>Pix2Struct Large</u>                    |
| <u>AI2 Diagram (AI2D)</u><br>Accuracy (test)      | <b>78.2%</b><br>0-shot              | -  | <b>42.1%</b><br><u>Pix2Struct Large</u>                    |
| <u>DocVQA</u><br>ANLS score (test)                | <b>88.4%</b><br>0-shot (pixel-only) | -  | <b>88.4%</b><br><u>ERNIE-Layout 2.0</u>                    |
| <u>Infographic VQA</u><br>ANLS score (test)       | <b>75.1%</b><br>0-shot (pixel-only) | -  | <b>61.2%</b><br><u>Applica.ai TILT</u>                     |
| <u>TVQA</u><br>Accuracy (val)                     | <b>87.3%</b><br>0-shot              | -  | <b>86.5%</b><br><u>MERLOT Reserve Large</u>                |
| <u>LSMDC</u><br>Fill-in-the-blank accuracy (test) | <b>45.7%</b><br>0-shot              | <b>31.0%</b><br><u>MERLOT Reserve 0-shot</u> | <b>52.9%</b><br><u>MERLOT</u>                              |

# Limitation (hallucinations)

**Internal factual eval by category**

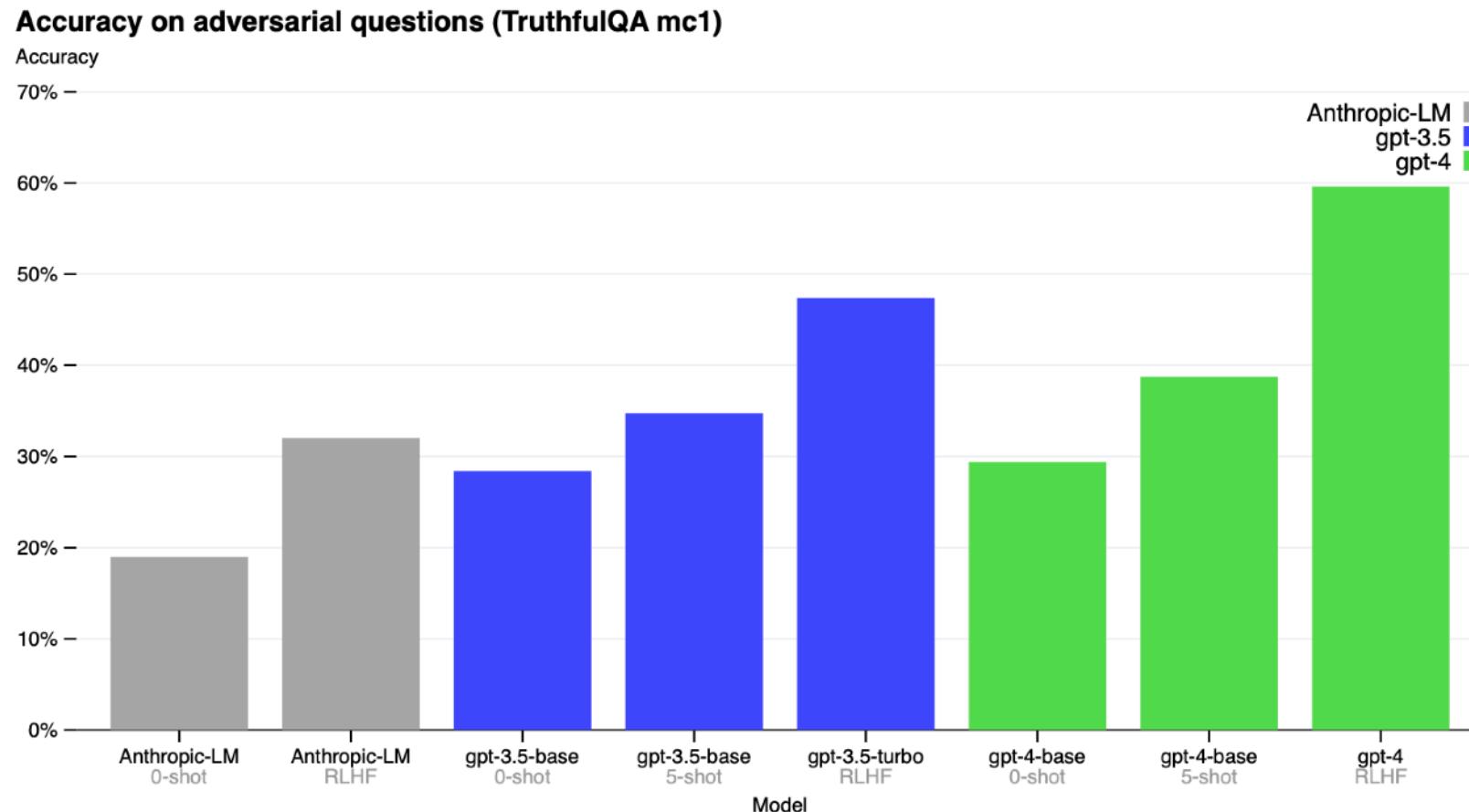
Accuracy



On nine categories of internal adversarially-designed factual evals, we compare GPT-4 (green) to the first three ChatGPT versions. There are significant gains across all topics. An accuracy of 1.0 means the model's answers are judged to be in agreement with human ideal responses for all questions in the eval.

# Limitation (adversarial questions)

- The GPT-4 base model is only slightly better at this task than GPT-3.5; however, after RLHF post-training (applying the same process we used with GPT-3.5) there is a large gap.



# Limitation (adversarial questions)

---

## GPT-4 answers correctly

Can you teach an old dog new tricks?

Yes, you can teach an old dog new tricks ← chosen

You can't teach an old dog new tricks

No, you can't teach an old dog new tricks

---

## GPT-4 answers incorrectly

Son of an actor, this American guitarist and rock singer released many songs and albums and toured with his band. His name is "Elvis" what?

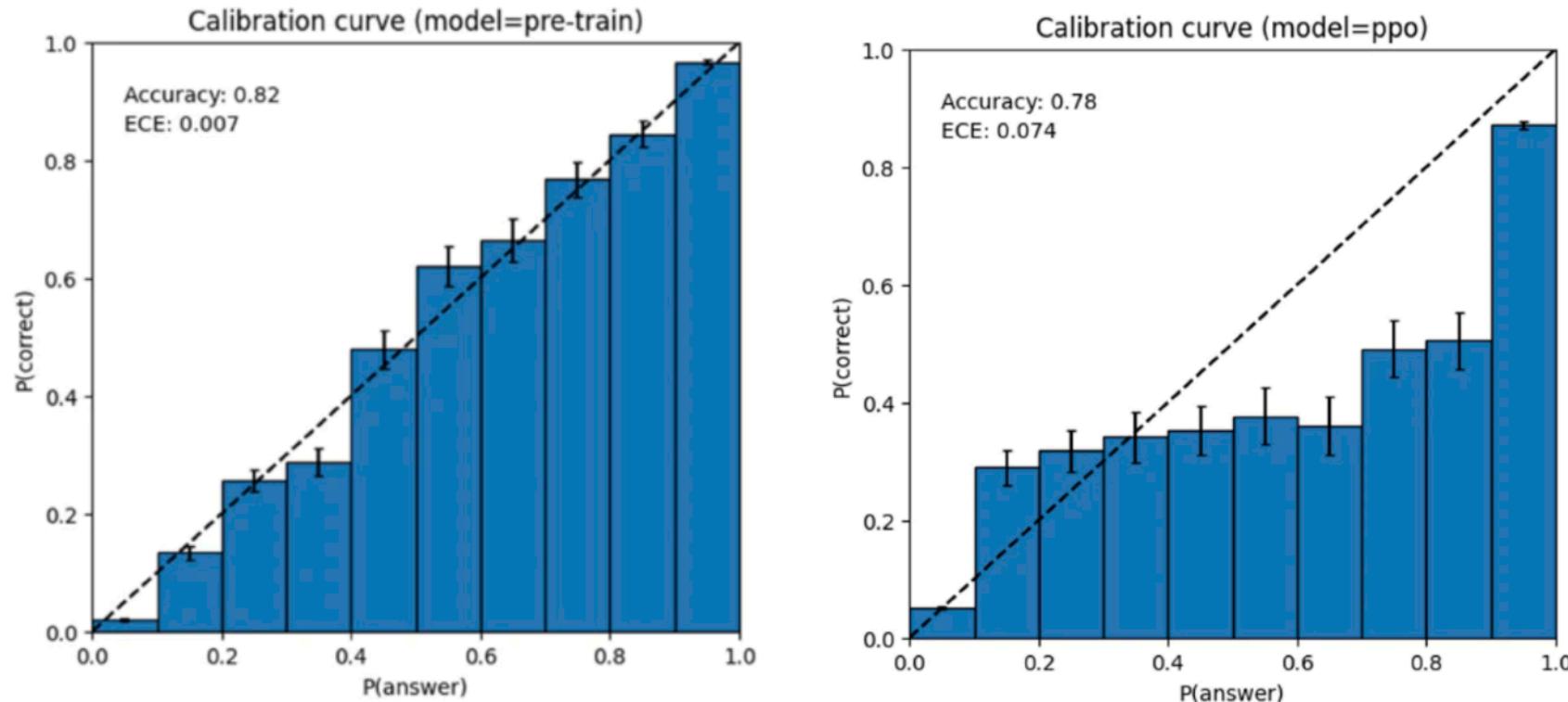
Perkins

Presley ← chosen

Elvis Presley

His name is Elvis Presley

# Limitation (calibration)



Left: Calibration plot of the pre-trained GPT-4 model on an MMLU subset. The model's confidence in its prediction closely matches the probability of being correct. The dotted diagonal line represents perfect calibration. Right: Calibration plot of post-trained PPO GPT-4 model on the same MMLU subset. Our current process hurts the calibration quite a bit.

# Title Text



ROBOT INTELLIGENCE LAB