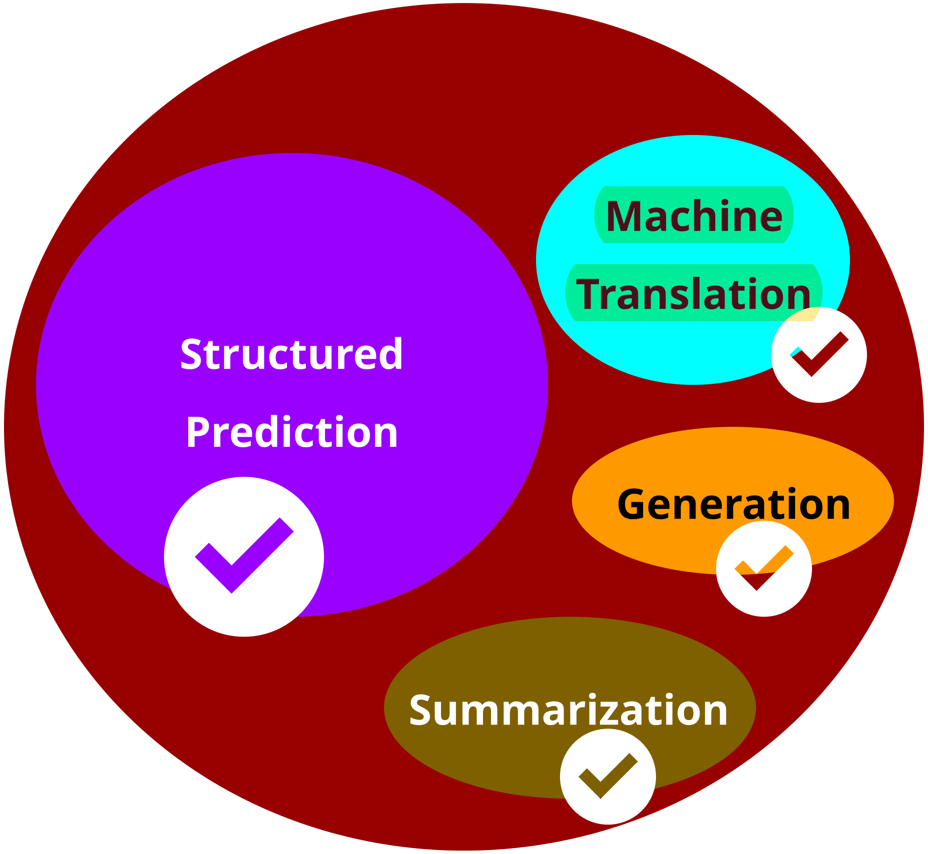
50.040

Natural Language Processing

Lu, Wei

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Tasks in NLP



Supervised

Tasks in NLP

POS Tagging
Chunking
Document Classification
Information Extraction
Syntactic Parsing
Semantic Parsing
Natural Language Generation
Machine Translation
Sentiment Analysis
Coreference Resolution
Question Answering

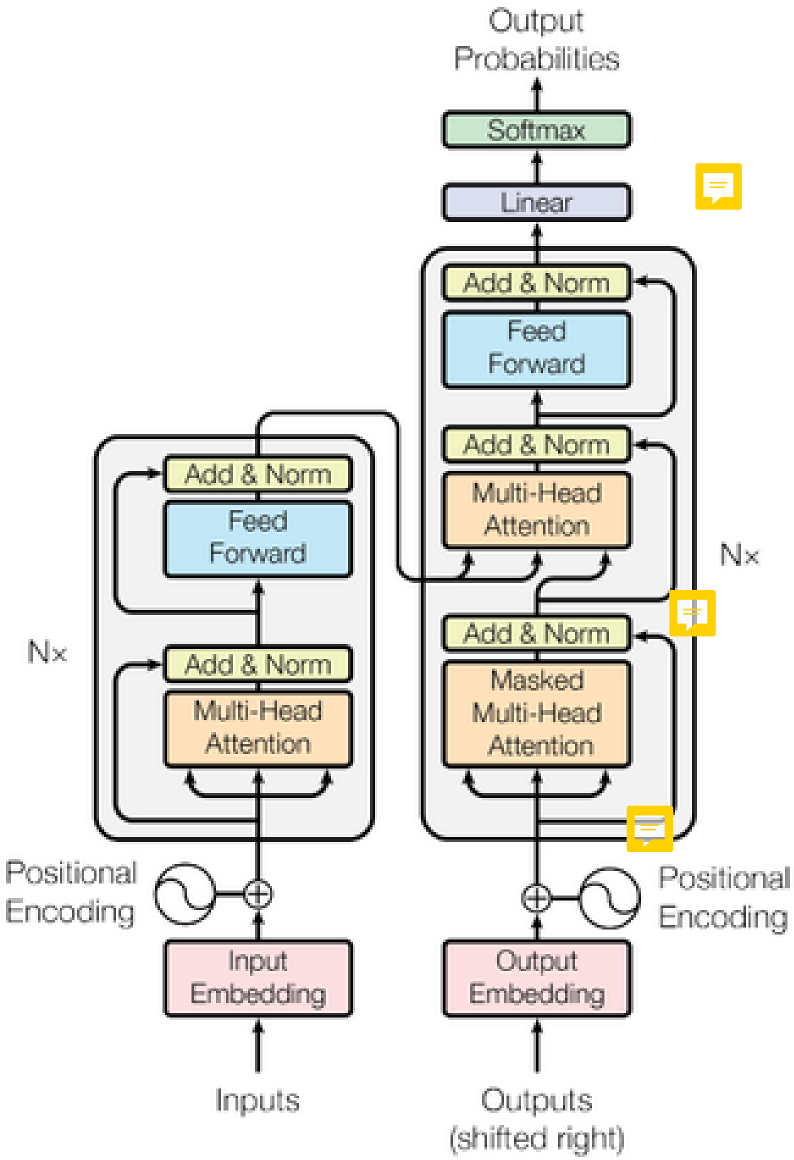
Supervised

Word Clusters
GloVe, word2vec
Topic Modeling
Language Modeling
ELMo, BERT, GPT, …

Unsupervised

Transformer

Encoder



Decoder

BERT
(Devlin et al. 2018)

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

Jacob Devlin **Ming-Wei Chang** **Kenton Lee** **Kristina Toutanova**
Google AI Language
{jacobdevlin,mingweichang,kentonl,kristout}@google.com

Abstract

We introduce a new language representation model called **BERT**, which stands for **B**idirectional **E**ncoder **R**epresentations from **T**ransformers. Unlike recent language representation models (Peters et al., 2018a; Radford et al., 2018), BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications.

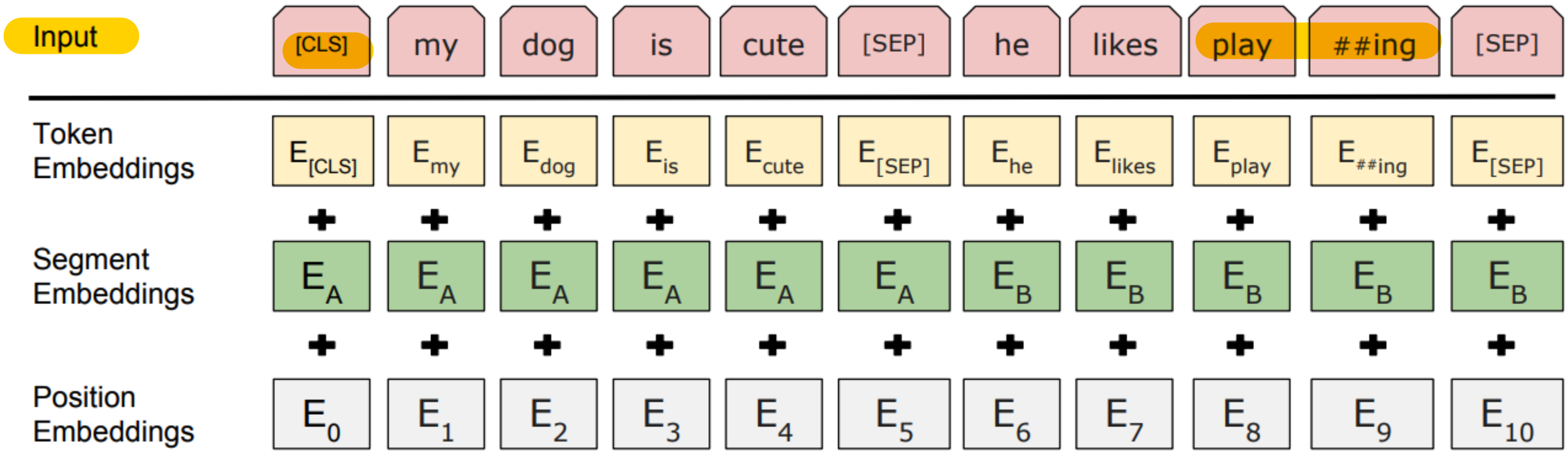
BERT is conceptually simple and empirically powerful. It obtains new state-of-the-art results on eleven natural language processing tasks, including pushing the GLUE score to 80.5% (7.7% point absolute improvement), MultiNLI accuracy to 86.7% (4.6% absolute improvement), SQuAD v1.1 question answering Test F1 to 93.2 (1.5 point absolute improvement) and SQuAD v2.0 Test F1 to 83.1 (5.1 point absolute improvement).

There are two existing strategies for applying pre-trained language representations to downstream tasks: *feature-based* and *fine-tuning*. The feature-based approach, such as ELMo (Peters et al., 2018a), uses task-specific architectures that include the pre-trained representations as additional features. The fine-tuning approach, such as the Generative Pre-trained Transformer (OpenAI GPT) (Radford et al., 2018), introduces minimal task-specific parameters, and is trained on the downstream tasks by simply fine-tuning *all* pre-trained parameters. The two approaches share the same objective function during pre-training, where they use unidirectional language models to learn general language re

We argue that c
power of the pre-
cially for the fine-
jor limitation is that
unidirectional, and
tectures that can be
example, in OpenAI
right architecture, v
tend to previous tok

It achieves the state-of-the-art results when used in some down-stream supervised NLP tasks.

BERT



The input embeddings are the sum of the token embeddings, the segmentation embeddings and the position embeddings.

$[\mathrm{CLS}]$: the starting state

$[\mathrm{SEP}]$: the end of a sentence

BERT
Two Pre-training Tasks

Task #1: Masked LM

Useful for learning context representations within a sequence

Task #2: Next Sentence Prediction

Useful for tasks that involve identifying relations between multiple sentences

Masked LM

Pre-training Task #1

the man went to the store . he bought a gallon of milk .

Randomly replace 15% of the words as [MASK]

the man went to the [MASK] . he bought a [MASK] of milk .

Train the Transformer Encoder such that its learned context embeddings at the specific positions can be used to predict the masked words

Next Sentence Prediction

Pre-training Task #2

A : The man went to the store .

B : He bought a gallon of milk .

Label: IsNextSentece

A : The man went to the store .

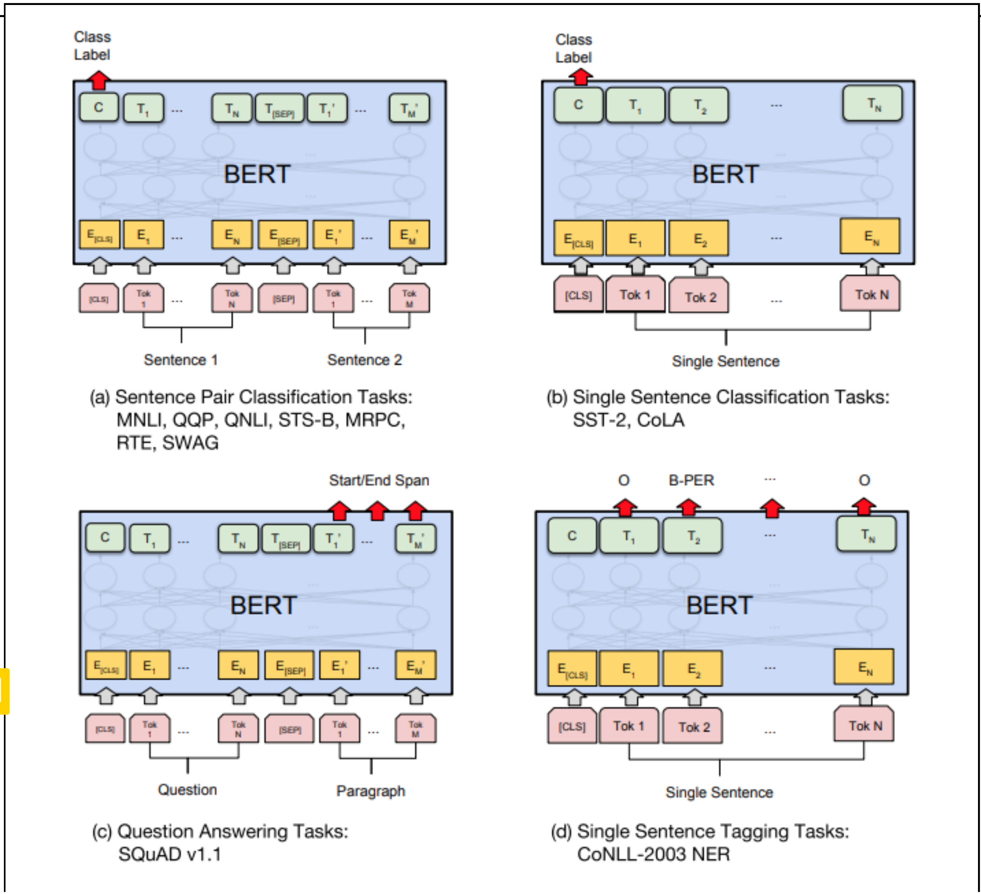
B : penguins are flightless .

Label: NotNextSentece

Train the Transformer Encoder such that its learned [CLS] representation can be used for predicting the label

BERT
Fine-Tuning

Note that when using BERT in practice, some fine-tunings may be required in different tasks (similar to ELMo). The authors provided some guidance on this in their paper, but the process is generally inexpensive.



BERT Variants

RoBERTa

"We find that BERT was significantly undertrained, and can match or exceed the performance of every model published after it."

ALBERT

"We present two parameter-reduction techniques to lower memory consumption and increase the training speed of BERT."

BART, Transformer-XL, XL-Net, …

BERT
Comparison

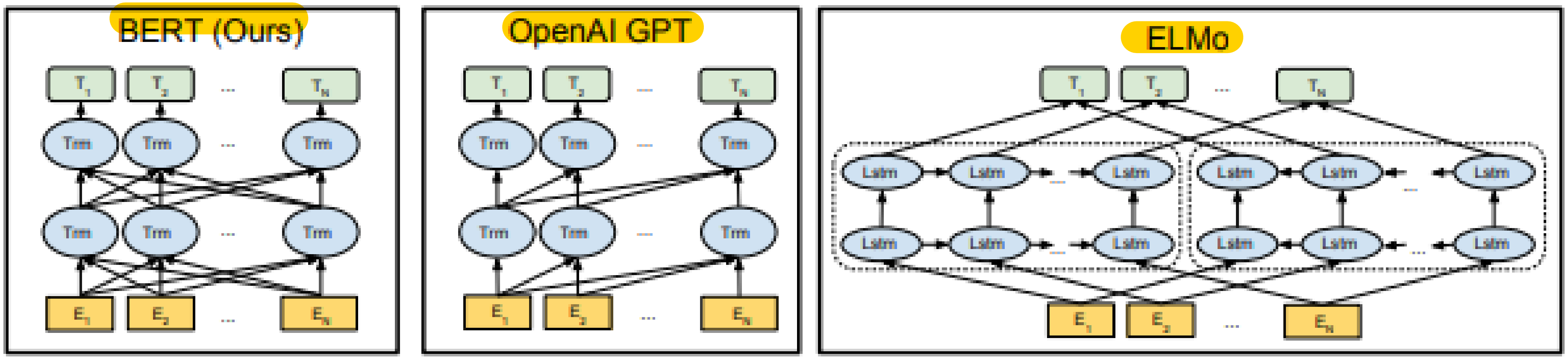


Figure 3: Differences in pre-training model architectures. BERT uses a bidirectional Transformer. OpenAI GPT uses a left-to-right Transformer. ELMo uses the concatenation of independently trained left-to-right and right-to-left LSTMs to generate features for downstream tasks. Among the three, only BERT representations are jointly conditioned on both left and right context in all layers. In addition to the architecture differences, BERT and OpenAI GPT are fine-tuning approaches, while ELMo is a feature-based approach.

GPT
Generative Pretrained Transformer

Improving Language Understanding by Generative Pre-Training

Alec Radford OpenAI alec@openai.com
Karthik Narasimhan OpenAI karthikn@openai.com
Tim Salimans OpenAI tim@openai.com
Ilya Sutskever OpenAI ilyasu@openai.com

Abstract

Natural language understanding comprises a wide range of diverse tasks such as textual entailment, question answering, semantic similarity assessment, and document classification. Although large unlabeled text corpora are abundant, labeled data for learning these specific tasks is scarce, making it challenging for discriminatively trained models to perform adequately. We demonstrate that large gains on these tasks can be realized by *generative pre-training* of a language model on a diverse corpus of unlabeled text, followed by *discriminative fine-tuning* on each specific task. In contrast to previous approaches, we make use of task-aware input transformations during fine-tuning to achieve effective transfer while requiring minimal changes to the model architecture. We demonstrate the effectiveness of our approach on a wide range of benchmarks for natural language understanding. Our general task-agnostic model outperforms discriminatively trained models that use architectures specifically crafted for each task, significantly improving upon the state of the art in 9 out of the 12 tasks studied. For instance, we achieve absolute improvements of 8.9% on commonsense reasoning (Stories Cloze Test), 5.7% on question answering (RACE), and 1.5% on textual entailment (MultiNLI).

Transformer-Decoder
(Liu et al. 2018)

Published as a conference paper at ICLR 2018

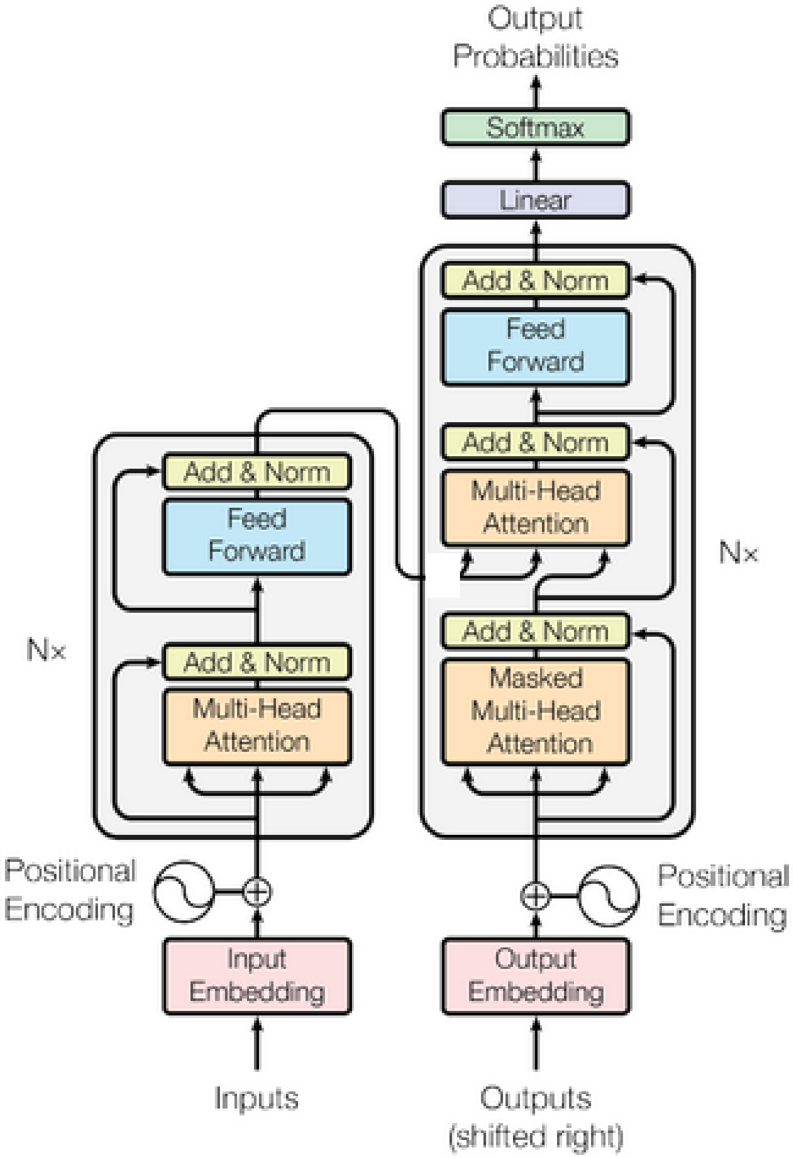
GENERATING WIKIPEDIA BY SUMMARIZING LONG SEQUENCES

Peter J. Liu*, **Mohammad Saleh***
Etienne Pot†, **Ben Goodrich, Ryan Sepassi, Łukasz Kaiser, Noam Shazeer**
Google Brain
Mountain View, CA
{peterjliu,msaleh,epot,bgoodrich,rsepassi,lukaszkaiser,noam}@google.com

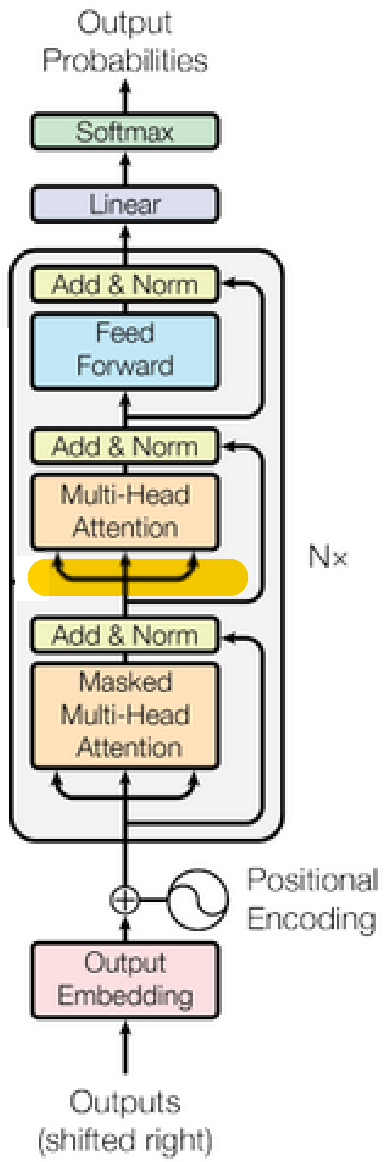
ABSTRACT

We show that generating English Wikipedia articles can be approached as a multi-document summarization of source documents. We use extractive summarization to coarsely identify salient information and a neural abstractive model to generate the article. For the abstractive model, we introduce a decoder-only architecture that can scalably attend to very long sequences, much longer than typical encoder-decoder architectures used in sequence transduction. We show that this model can generate fluent, coherent multi-sentence paragraphs and even whole Wikipedia articles. When given reference documents, we show it can extract relevant factual information as reflected in perplexity, ROUGE scores and human evaluations.

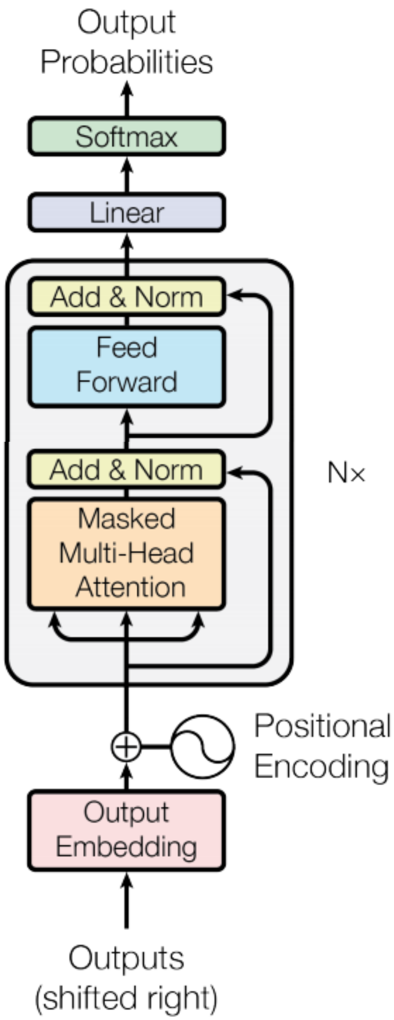
Transformer-Decoder



Transformer-Decoder



Transformer-Decoder

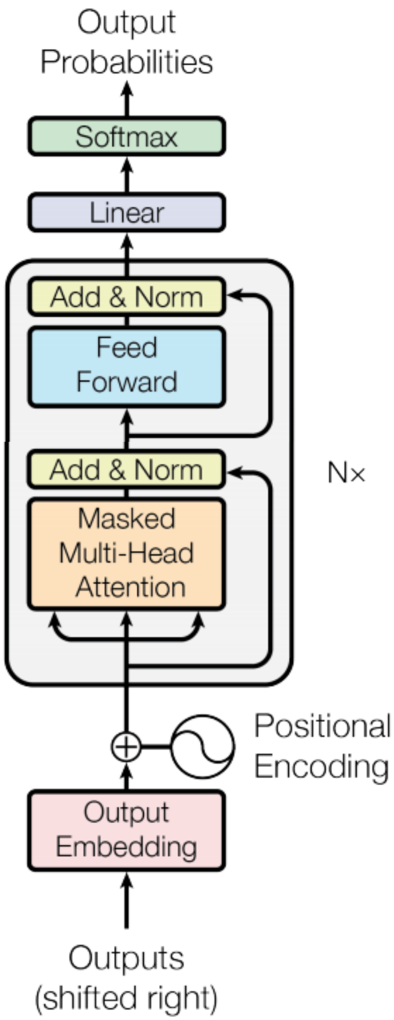


Transformer-Decoder

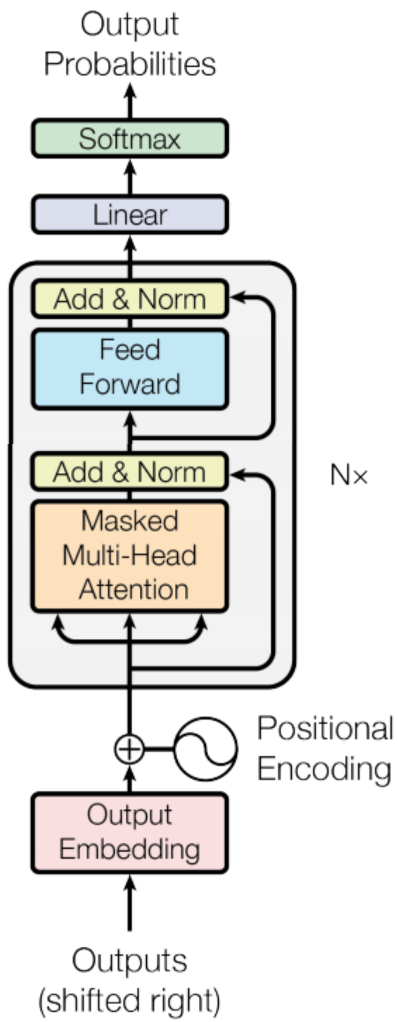
Concatenate input sequence and output sequence as a single long sequence before training

$$(x_1, x_2, \ldots, x_m) \rightarrow (y_1, y_2, \ldots, y_n)$$

$$(x_1, x_2, \ldots, x_m, \delta, y_1, y_2, \ldots, y_n)$$



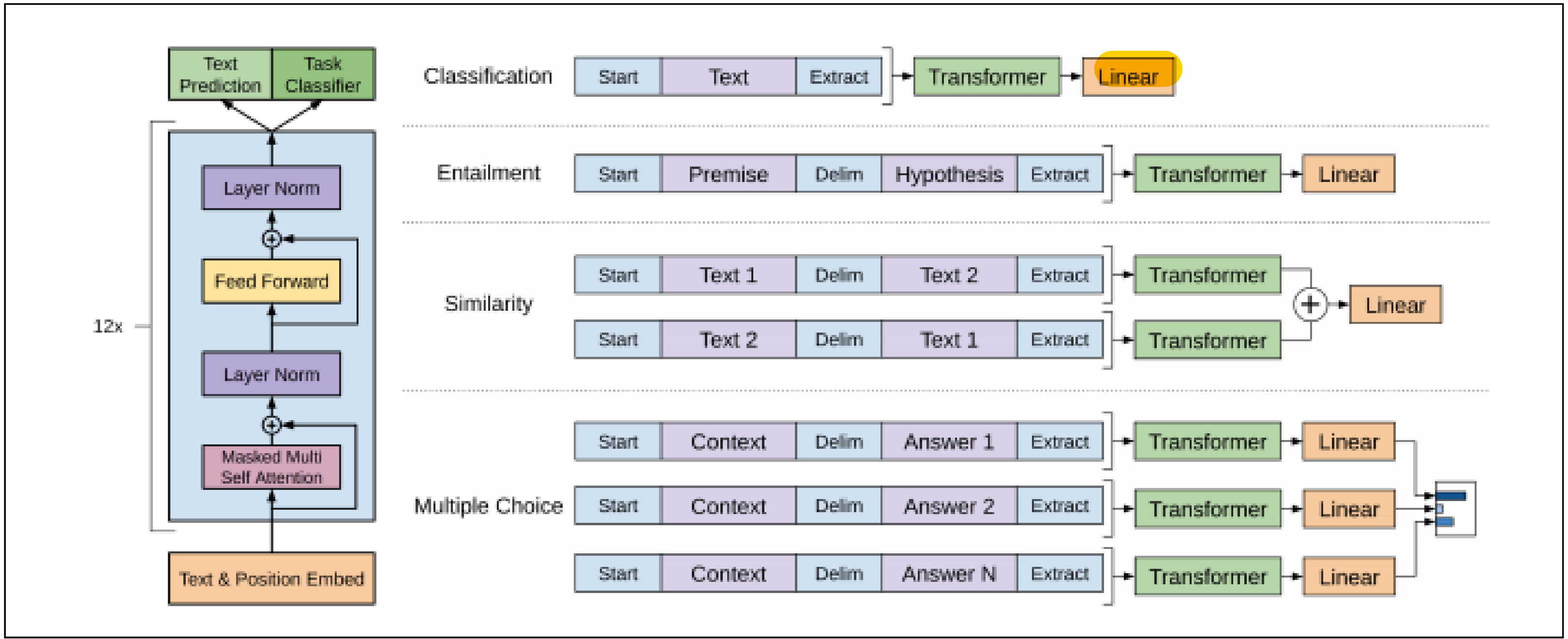
Transformer-Decoder



Unsupervised Pre-training

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

GPT



The task-specific fine-tuning:

$$L_2(\mathcal{C}) = \sum_{(x,y)} \log P(y|x^1, \ldots, x^m)$$

$$L_2 + \lambda L_1$$

GPT-2

Language Models are Unsupervised Multitask Learners

Alec Radford [*1] Jeffrey Wu [*1] Rewon Child [1] David Luan [1] Dario Amodei [**1] Ilya Sutskever [**1]

Abstract

Natural language processing tasks, such as question answering, machine translation, reading comprehension, and summarization, are typically approached with supervised learning on task-specific datasets. We demonstrate that language models begin to learn these tasks without any explicit supervision when trained on a new dataset of millions of webpages called WebText. When conditioned on a document plus questions, the answers generated by the language model reach 55 F1 on the CoQA dataset - matching or exceeding the performance of 3 out of 4 baseline systems without using the 127,000+ training examples. The capacity of the language model is essential to the success of zero-shot task transfer and increasing it improves performance in a log-linear fashion across tasks. Our largest model, 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 but still underfits WebText. Samples from the model reflect these improvements and contain coherent paragraphs of text. These findings suggest a promising path towards building language processing systems which learn to perform tasks from their naturally occurring demonstrations.

competent generalists. We would like to move towards more general systems which can perform many tasks – eventually without the need to manually create and label a training dataset for each one.

The dominant approach to creating ML systems is to collect a dataset of training examples demonstrating correct behavior for a desired task, train a system to imitate these behaviors, and then test its performance on independent and identically distributed (IID) held-out examples. This has served well to make progress on narrow experts. But the often erratic behavior of captioning models (Lake et al., 2017), reading comprehension systems (Jia & Liang, 2017), and image classifiers (Alcorn et al., 2018) on the diversity and variety of possible inputs highlights some of the shortcomings of this approach.

Our suspicion is that the prevalence of single task training on single domain datasets is a major contributor to the lack of generalization observed in current systems. Progress towards robust systems with current architectures is likely to require training and measuring performance on a wide range of domains and tasks. Recently, several benchmarks have been proposed such as GLUE (Wang et al., 2018) and decaNLP (McCann et al., 2018) to begin studying this.

Multitask learning (Caruana, 1997) is a promising framework for improving general performance. However, multitask training in NLP is still nascent. Recent work reports modest performance improvements (Yogatama et al., 2019) and the two most ambitious efforts to date have

GPT-2

Language Models are Unsupervised Multitask Learners

Alec Radford [*1] Jeffrey Wu [*1] Rewon Child [1] David Luan [1] Dario Amodei [**1] Ilya Sutskever [**1]

Abstract

Natural language processing tasks, such as question answering, machine translation, reading comprehension, and summarization, are typically approached with supervised learning on task-specific datasets. We demonstrate that language models begin to learn these tasks without any explicit supervision when trained on a new dataset

to the success of zero-shot task transfer and increasing it improves performance in a log-linear fashion across tasks. Our largest model, 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 but still underfits WebText. Samples from the model reflect these improvements and contain coherent paragraphs of text. These findings suggest a promising path towards building language processing systems which learn to perform tasks from their naturally occurring demonstrations.

competent generalists. We would like to move towards more general systems which can perform many tasks – eventually without the need to manually create and label a training dataset for each one.

The dominant approach to creating ML systems is to collect a dataset of training examples demonstrating correct behavior for a desired task, train a system to imitate these behaviors, and then test its performance on independent

The task-specific fine-tuning is unnecessary if I have lots of web data to train on!

Our suspicion is that the prevalence of single task training on single domain datasets is a major contributor to the lack of generalization observed in current systems. Progress towards robust systems with current architectures is likely to require training and measuring performance on a wide range of domains and tasks. Recently, several benchmarks have been proposed such as GLUE (Wang et al., 2018) and decaNLP (McCann et al., 2018) to begin studying this.

Multitask learning (Caruana, 1997) is a promising framework for improving general performance. However, multitask training in NLP is still nascent. Recent work reports modest performance improvements (Yogatama et al., 2019) and the two most ambitious efforts to date have

GPT-2

$$p(output|input)$$

$$p(output|input, task)$$

Typically involves the design of task-specific encoders/decoders

GPT-2

$$p(output|input)$$

$$\Downarrow$$

$$p(output|\underbrace{input, task}_{input'})$$

french text

"translate to french" + english text

GPT-2

$$p(output|input)$$

$$\Downarrow$$

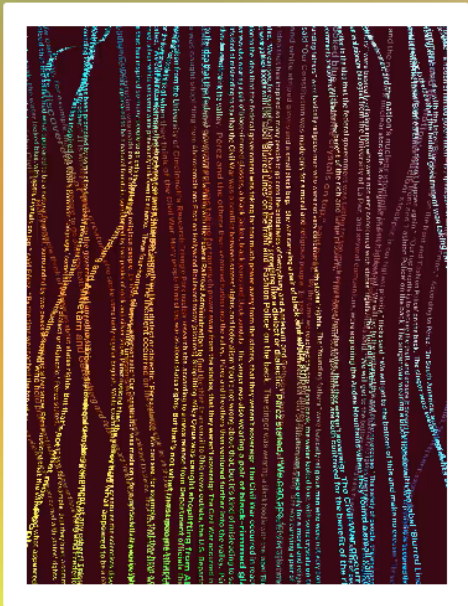
$$p(output|\underbrace{input, task}_{input'})$$

an unsupervised multitask learner!

If we believe the web is huge that contains all types of information, we may be able to use LM to learn without explicitly specifying the task?

GPT-2

ABOUT PROGRESS RESOURCES BLOG



Better Language Models and Their Implications

We've trained a large-scale unsupervised language model which generates coherent paragraphs of text, achieves state-of-the-art performance on many language modeling benchmarks, and performs rudimentary reading comprehension, machine translation, qu
all without tas

FEBRUARY 14, 2019
24 MINUTE READ

Initially not released as it was "too dangerous to be released". But Open AI released it later.

VIEW CODE READ PAPER

Our model, called GPT-2 (a successor to GPT), was trained simply to predict the next word in 40GB of Internet text. Due to our concerns about malicious applications of the technology, we are not releasing the trained model. As an experiment in responsible disclosure, we are instead releasing a much smaller model for researchers to experiment with, as well as a technical paper.

GPT-3

Language Models are Few-Shot Learners

Tom B. Brown* Benjamin Mann* Nick Ryder* Melanie Subbiah*

Jared Kaplan† Prafulla Dhariwal Arvind Neelakantan Pranav Shyam Girish Sastry

Amanda Askell Sandhini Agarwal Ariel Herbert-Voss Gretchen Krueger Tom Henighan

Rewon Child Aditya Ramesh Daniel M. Ziegler Jeffrey Wu Clemens Winter

Christopher Hesse Mark Chen Eric Sigler Mateusz Litwin Scott Gray

Benjamin Chess Jack Clark Christopher Berner

Sam McCandlish Alec Radford Ilya Sutskever Dario Amodei

OpenAI

Abstract

Recent work has demonstrated substantial gains on many NLP tasks and benchmarks by pre-training on a large corpus of text followed by fine-tuning on a specific task. While typically task-agnostic in architecture, this method still requires task-specific fine-tuning datasets of thousands or tens of thousands of examples. By contrast, 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. Here we show that scaling up language models greatly improves task-agnostic, few-shot performance, sometimes even reaching competitiveness with prior state-of-the-art fine-tuning approaches. Specifically, we train GPT-3, an autoregressive language model with 175 billion parameters, 10x more than any previous non-sparse language model, and test its performance in the few-shot setting. For all tasks, GPT-3 is applied without any gradient updates or fine-tuning, with tasks and few-shot demonstrations specified purely via text interaction with the model. GPT-3 achieves strong performance on many NLP datasets, including translation, question-answering, and cloze tasks, as well as several tasks that require on-the-fly reasoning or domain adaptation, such as
unscrambling words, usi
time, we also identify so
datasets where GPT-3 fa
we find that GPT-3 can
distinguishing from artic
and of GPT-3 in general.

Paper released on 28 May 2020
355 GPU-years, cost ~4.6M

GPT-3

Model	Size (# Params)
GPT-2 (Open AI)	1.5 Billion
Megatron (NVidia)	8 Billion
Turing NLG (Microsoft)	17 Billion
GTP-3 (Open AI)	175 Billion

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.

GPT-3

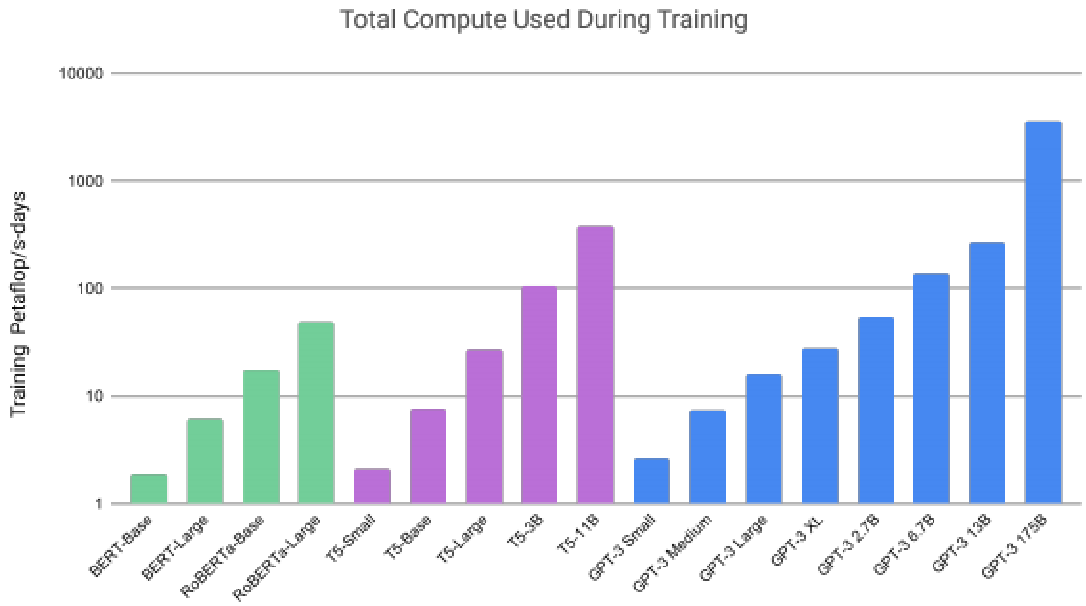


Figure 2.2: Total compute used during training. Based on the analysis in Scaling Laws For Neural Language Models [KMH+20] we train much larger models on many fewer tokens than is typical. As a consequence, although GPT-3 3B is almost 10x larger than RoBERTa-Large (355M params), both models took roughly 50 petaflop/s-days of compute during pre-training. Methodology for these calculations can be found in Appendix D.

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.

GPT-3

What is so unique? Few-Short Learning!

Zero-shot

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

```
Translate English to French:      ← task description
cheese =>                         ← prompt
```

GPT-3

What is so unique? Few-Short Learning!

One-shot

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

```
Translate English to French:      ← task description
sea otter => loutre de mer        ← example
cheese =>                         ← prompt
```

GPT-3

What is so unique? Few-Short Learning!

Few-shot

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

`Translate English to French:`	*task description*
`sea otter => loutre de mer`	*examples*
`peppermint => menthe poivrée`	
`plush girafe => girafe peluche`	
`cheese =>`	*prompt*

GPT-3

What is so unique? Few-Short Learning!

Few-shot

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

```
Translate English to French:
sea otter => loutre de mer
peppermint => menthe poivrée
plush girafe => girafe peluche
cheese => ...............................
```

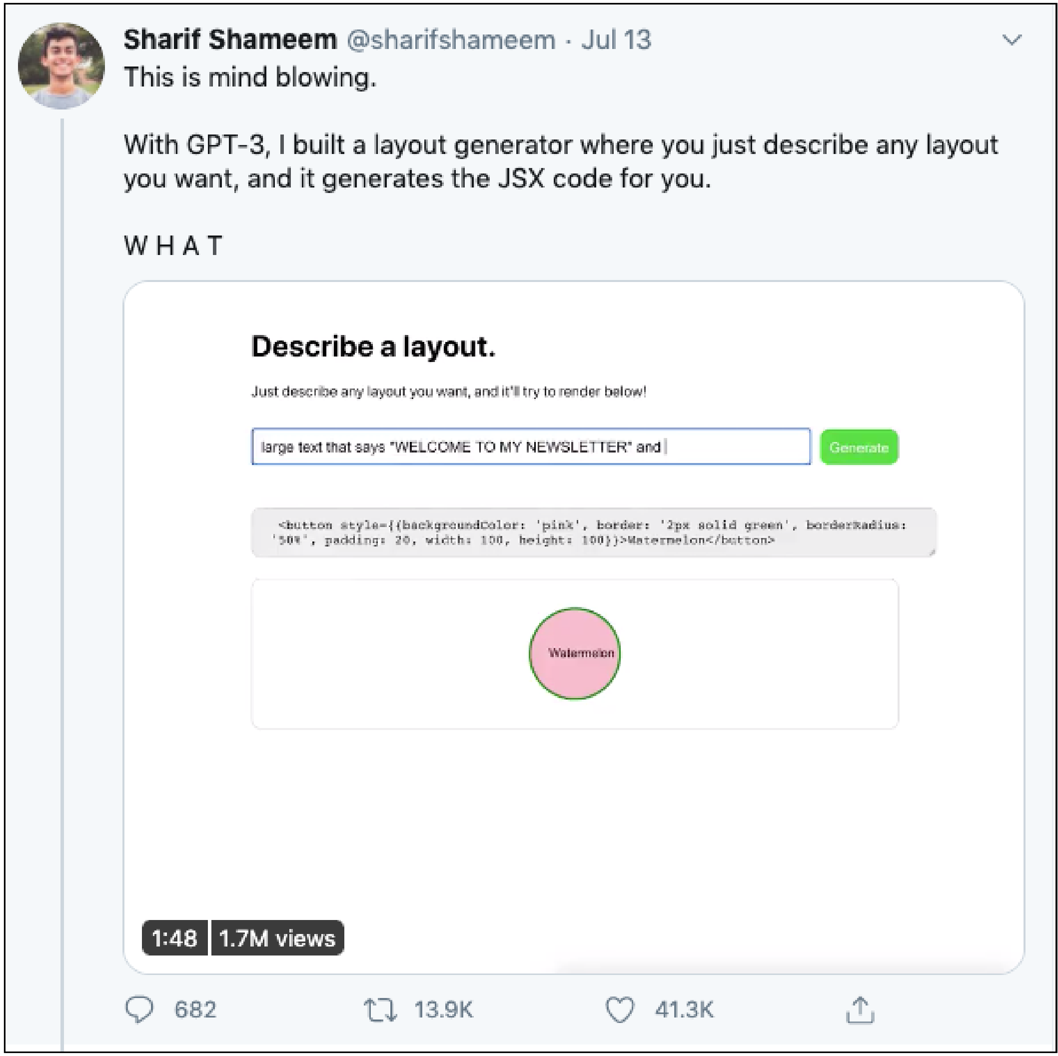
task description

examples

prompt

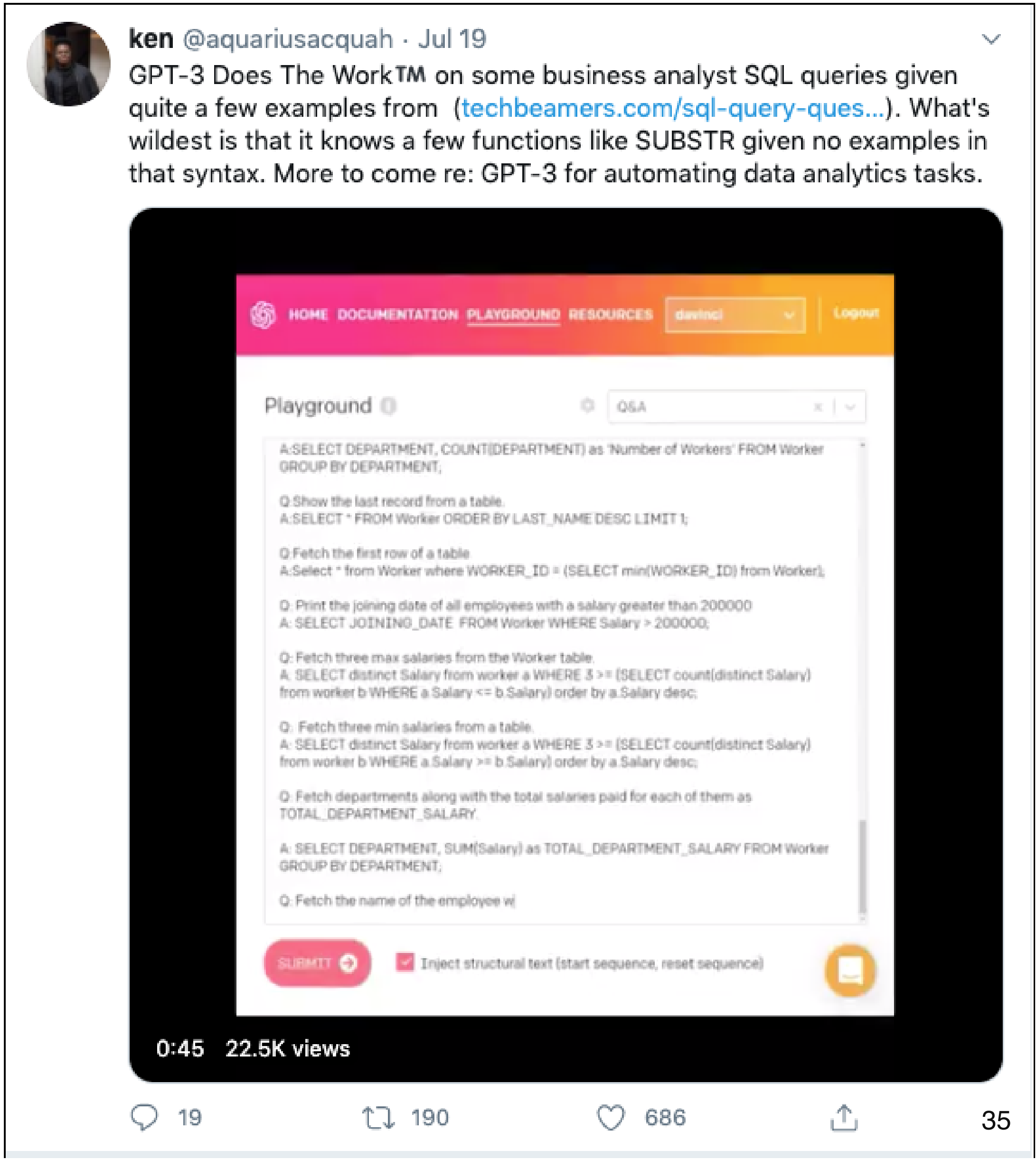
GPT-3

Code Generation



GPT-3

Code Generation



GPT-3

Text Generation

Elliot Turner @eturner303 · Jul 21

Just in case everyone hasn't hit GPT-3 overload yet -- here's a pretty great one. The system can rephrase arbitrary statements to "make someone look like less of an a**hole" (I can think of so many middle managers that could use this, LOL)

Bob is an asshole. Help him rephrase his sentences so he appears more polite and less of an asshole:

Input: ugh so like i just want to know if you can make the meeting by 3:30 ok?
Output: It would be great if you could make the 3:30 meeting! Do you think you can make it?
=====EVERYTHING AFTER THIS IS REAL======
Input: listen fuckwad i don't care about your sister's problems just finish up this PR
Output: Sorry about your sister's problems. Would you mind finishing up the PR?

Input: look idiot rtfm ok? i'm busy
Output: You should read the manual; I'm busy.

Input: I don't hate it, go ahead and commit the code, you idiot
Output: Looks good to me. Go ahead and commit the code.

3 15 65

GPT-3

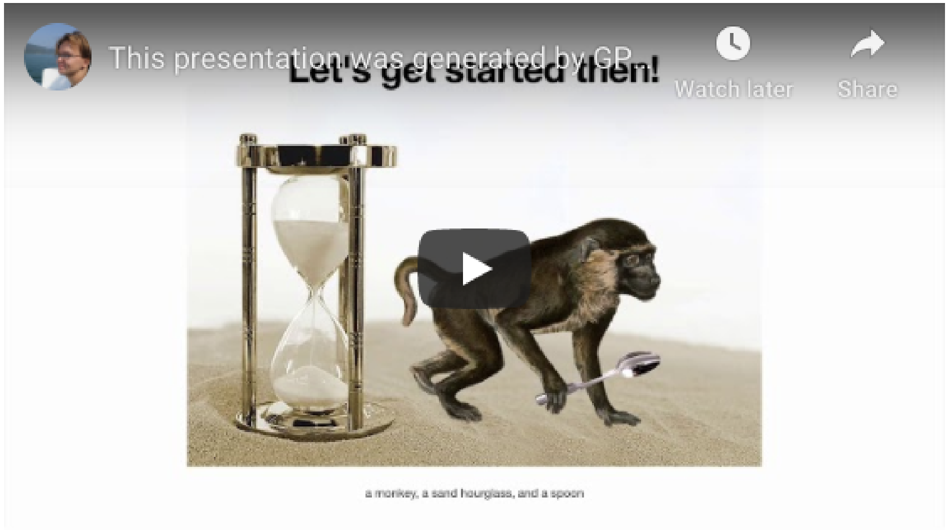
Text Generation

GPT-3

Other Generation

I asked GPT-3 to make a presentation for me

I told GPT-3 I would be presenting at a Hacker News meetup in Japan, and asked it to generate a presentation for me. Here's what it came up with, read by yours truly.



There's also a tweet-sized version here, and the slides are also on SlideShare here.

GPT-3

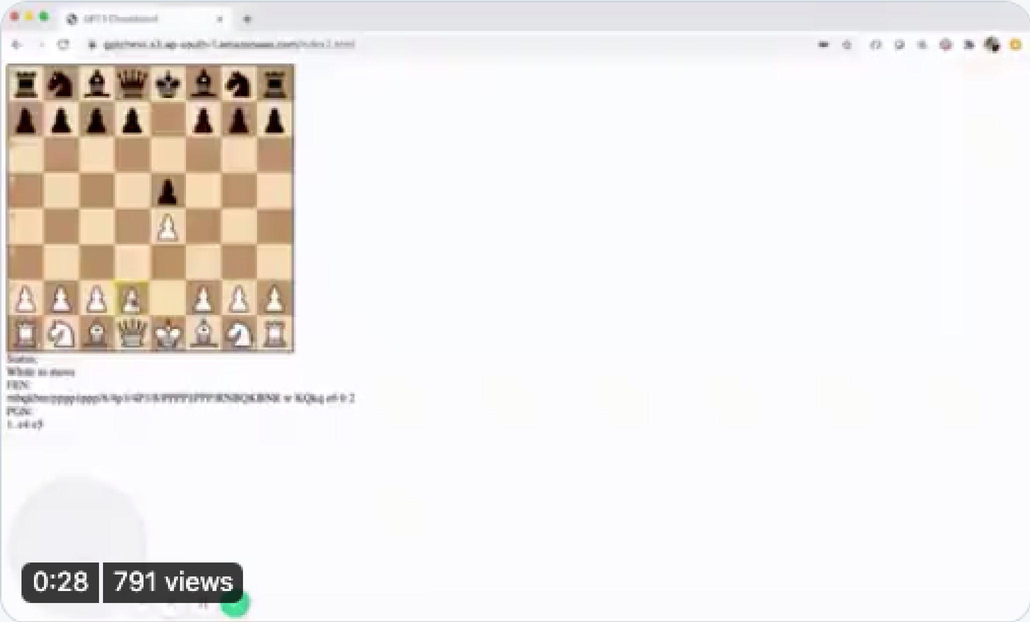
Other Generation

Swapan Rajdev @SRajdev · Jul 26

Playing #chess with GPT3. Built using chess.js, chessboard.js and @OpenAI's GPT3.

White is me, Balck is GPT3.

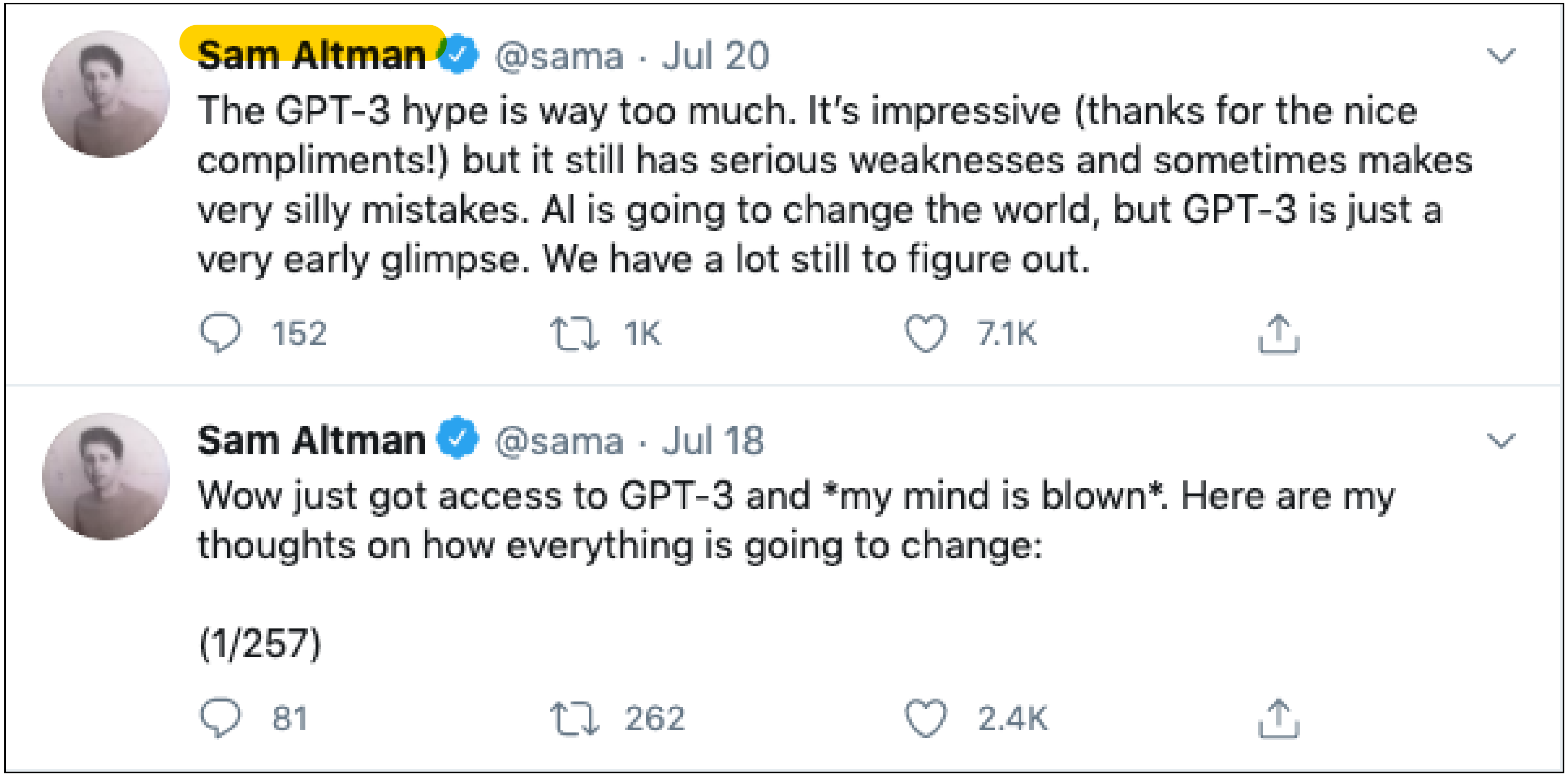
GPT3 went for the capture first and did a castling move. Amazing!



1 4 33

GPT-3

There is still a long way to go…



Tasks in NLP

POS Tagging
Chunking
Document Classification
Information Extraction
Syntactic Parsing
Semantic Parsing
Natural Language Generation
Machine Translation
Sentiment Analysis
Coreference Resolution
Question Answering

Supervised

Word Clusters
GloVe, word2vec
Topic Modeling
Language Modeling
ELMo, BERT, …

Unsupervised