Natural Language Processing with Deep Learning

Lecture 10 — Evolution of transformer models

Prof. Dr. Ivan Habernal

January 8, 2026

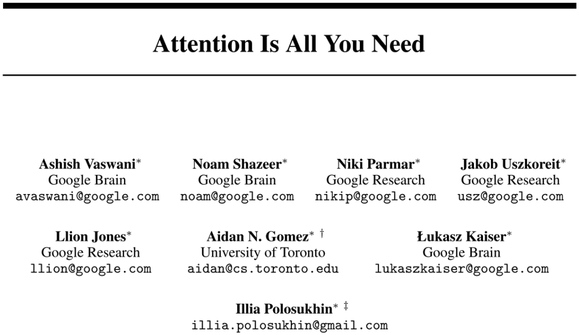
`www.trusthlt.org`
Trustworthy Human Language Technologies Group (TrustHLT)
Ruhr University Bochum & Research Center Trustworthy Data Science and Security

Last lecture

Decoder-only models: GPT (General Pre-trained Transformer)

Motivation

Knowing encoder transformer (BERT) and decoder transformer (GPT), let’s go back to the origins



Attention Is All You Need

Ashish Vaswani*
Google Brain
avaswani@google.com

Noam Shazeer*
Google Brain
noam@google.com

Niki Parmar*
Google Research
nikip@google.com

Jakob Uszkoreit*
Google Research
usz@google.com

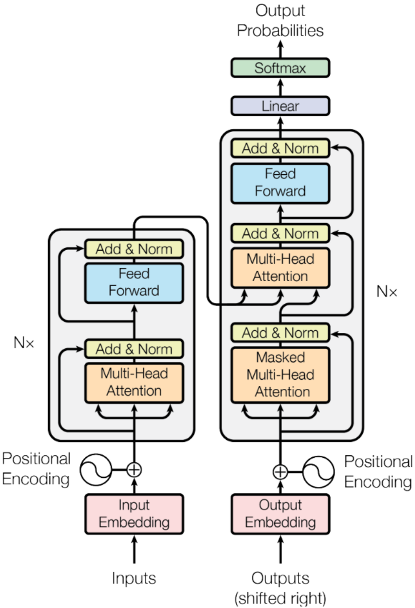
Llion Jones*
Google Research
llion@google.com

Aidan N. Gomez* †
University of Toronto
aidan@cs.toronto.edu

Łukasz Kaiser*
Google Brain
lukaszkaiser@google.com

Illia Polosukhin* ‡
illia.polosukhin@gmail.com

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin (2017). **“Attention Is All You Need”.** In: *Advances in Neural Information Processing Systems 30.* Long Beach, CA, USA: Curran Associates, Inc., pp. 5998–6008



A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin (2017). **"Attention Is All You Need".** In: *Advances in Neural Information Processing Systems 30.* Long Beach, CA, USA: Curran Associates, Inc., pp. 5998–6008

Transformer

The Transformer uses multi-head attention in three different ways:

(1) In "encoder-decoder attention" layers, the queries come from the previous decoder layer, and the memory keys and values come from the output of the encoder. This allows every position in the decoder to attend over all positions in the input sequence. This mimics the typical encoder-decoder attention mechanisms in sequence-to-sequence models

Transformer

The Transformer uses multi-head attention in three different ways:

(2) The encoder contains self-attention layers. In a self-attention layer all of the keys, values and queries come from the same place, in this case, the output of the previous layer in the encoder. Each position in the encoder can attend to all positions in the previous layer of the encoder.

Transformer

The Transformer uses multi-head attention in three different ways:

(3) Similarly, self-attention layers in the decoder allow each position in the decoder to attend to all positions in the decoder up to and including that position. We need to prevent leftward information flow in the decoder to preserve the auto-regressive property. We implement this inside of scaled dot-product attention by masking out (setting to $-\infty$) all values in the input of the softmax which correspond to illegal connections.

Transformer – the task

We trained on the standard WMT 2014 English-German dataset consisting of about 4.5 million sentence pairs. Sentences were encoded using byte-pair encoding, which has a shared source- target vocabulary of about 37000 tokens. For English-French, we used the significantly larger WMT 2014 English-French dataset consisting of 36M sentences and split tokens into a 32000 word-piece vocabulary.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin (2017). **"Attention Is All You Need".** In: *Advances in Neural Information Processing Systems 30.* Long Beach, CA, USA: Curran Associates, Inc., pp. 5998–6008

Transformer – results

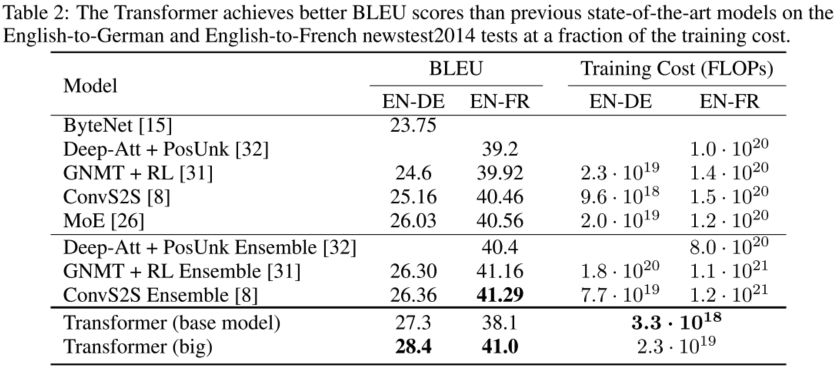


Table 2: The Transformer achieves better BLEU scores than previous state-of-the-art models on the English-to-German and English-to-French newstest2014 tests at a fraction of the training cost.

Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [15]	23.75			
Deep-Att + PosUnk [32]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [31]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [8]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [26]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [32]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [31]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [8]	26.36	**41.29**	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1	$\mathbf{3.3 \cdot 10^{18}}$	
Transformer (big)	**28.4**	**41.0**	$2.3 \cdot 10^{19}$	

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin (2017). **“Attention Is All You Need”.** In: *Advances in Neural Information Processing Systems 30.* Long Beach, CA, USA: Curran Associates, Inc., pp. 5998–6008

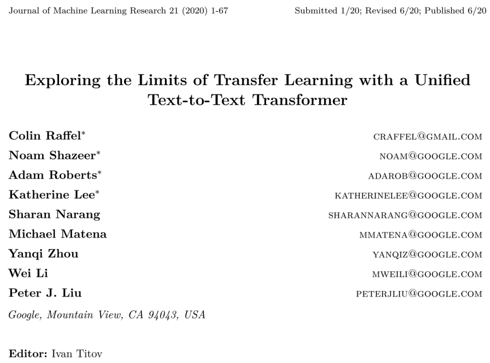
Every task is a text-to-text task

1. Every task is a text-to-text task
2. Evolution of GPT
3. In-context learning
4. "Alignment", instruction-tuning, RLHF

T5

"The basic idea underlying our work is to treat every text processing problem as a "text-to-text" problem, i.e. taking text as input and producing new text as output."

C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu (2020). **"Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer".** In: *Journal of Machine Learning Research* 21.140, pp. 1–67



Journal of Machine Learning Research 21 (2020) 1-67 Submitted 1/20; Revised 6/20; Published 6/20

Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

Colin Raffel* craffel@gmail.com
Noam Shazeer* noam@google.com
Adam Roberts* adarob@google.com
Katherine Lee* katherinelee@google.com
Sharan Narang sharannarang@google.com
Michael Matena mmatena@google.com
Yanqi Zhou yanqiz@google.com
Wei Li mweili@google.com
Peter J. Liu peterjliu@google.com
Google, Mountain View, CA 94043, USA

Editor: Ivan Titov

T5

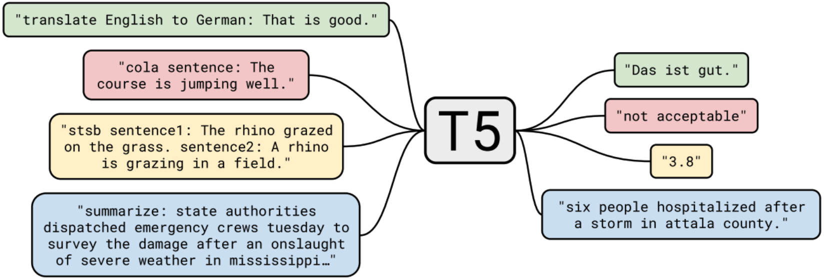
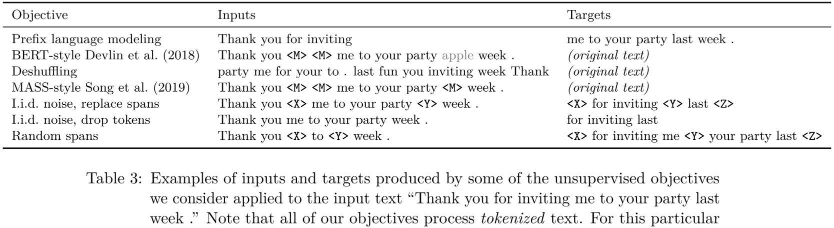

"translate English to German: That is good."
"cola sentence: The course is jumping well."
"stsb sentence1: The rhino grazed on the grass. sentence2: A rhino is grazing in a field."
"summarize: state authorities dispatched emergency crews tuesday to survey the damage after an onslaught of severe weather in mississippi…"
T5
"Das ist gut."
"not acceptable"
"3.8"
"six people hospitalized after a storm in attala county."

Figure 1: A diagram of our text-to-text framework. Every task we consider—including translation, question answering, and classification—is cast as feeding our model text as input and training it to generate some target text. This allows us to use the same model, loss function, hyperparameters, etc. across our diverse set of tasks. It also provides a standard testbed for the methods included in our empirical survey. "T5" refers to our model, which we dub the "**T**ext-**t**o-**T**ext **T**ransfer **T**ransformer".

C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu (2020). **"Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer".** In: *Journal of Machine Learning Research* 21.140, pp. 1–67

T5 — self-supervised pre-training

C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu (2020). **“Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer”.** In: *Journal of Machine Learning Research* 21.140, pp. 1–67



Objective	Inputs	Targets
Prefix language modeling	Thank you for inviting	me to your party last week .
BERT-style Devlin et al. (2018)	Thank you `<M>` `<M>` me to your party apple week .	*(original text)*
Deshuffling	party me for your to . last fun you inviting week Thank	*(original text)*
MASS-style Song et al. (2019)	Thank you `<M>` `<M>` me to your party `<M>` week .	*(original text)*
I.i.d. noise, replace spans	Thank you `<X>` me to your party `<Y>` week .	`<X>` for inviting `<Y>` last `<Z>`
I.i.d. noise, drop tokens	Thank you me to your party week .	for inviting last
Random spans	Thank you `<X>` to `<Y>` week .	`<X>` for inviting me `<Y>` your party last `<Z>`

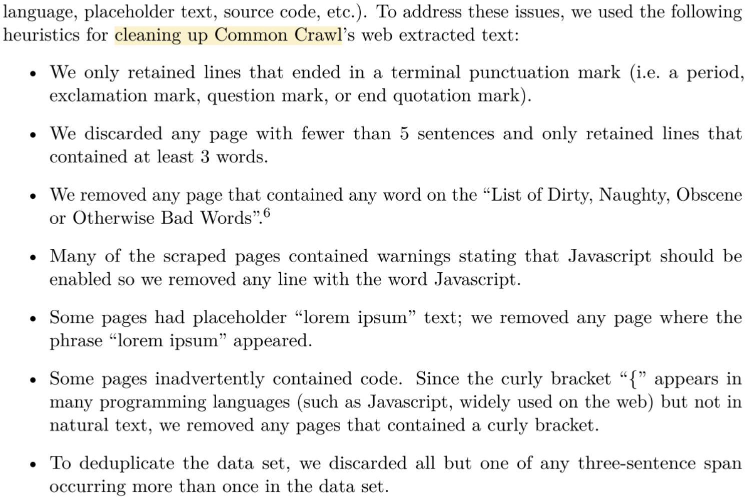
Table 3: Examples of inputs and targets produced by some of the unsupervised objectives we consider applied to the input text “Thank you for inviting me to your party last week .” Note that all of our objectives process *tokenized* text. For this particular

T5 – Source data quality matters

C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu (2020). **"Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer".** In: *Journal of Machine Learning Research* 21.140, pp. 1–67

"Common Crawl is a publicly-available web archive that provides "web extracted text" by removing markup and other non-text content from the scraped HTML files. This process produces around 20TB of scraped text data each month. Unfortunately, the majority of the resulting text is not natural language."

T5 – Colossal Clean Common Crawl corpus (about 750 GB)



language, placeholder text, source code, etc.). To address these issues, we used the following heuristics for cleaning up Common Crawl's web extracted text:

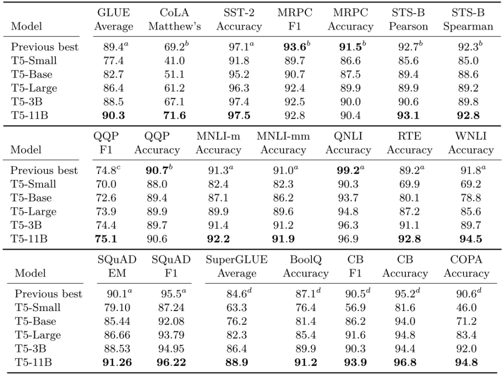
- We only retained lines that ended in a terminal punctuation mark (i.e. a period, exclamation mark, question mark, or end quotation mark).
- We discarded any page with fewer than 5 sentences and only retained lines that contained at least 3 words.
- We removed any page that contained any word on the "List of Dirty, Naughty, Obscene or Otherwise Bad Words".[6]
- Many of the scraped pages contained warnings stating that Javascript should be enabled so we removed any line with the word Javascript.
- Some pages had placeholder "lorem ipsum" text; we removed any page where the phrase "lorem ipsum" appeared.
- Some pages inadvertently contained code. Since the curly bracket "{" appears in many programming languages (such as Javascript, widely used on the web) but not in natural text, we removed any pages that contained a curly bracket.
- To deduplicate the data set, we discarded all but one of any three-sentence span occurring more than once in the data set.

C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu (2020). **"Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer".** In: *Journal of Machine Learning Research* 21.140, pp. 1–67

T5 – Scale matters the most

"scaling the model size to 11 billion parameters was the most important ingredient for achieving our best performance."

C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu (2020). **"Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer".** In: *Journal of Machine Learning Research* 21.140, pp. 1–67



Model	GLUE Average	CoLA Matthew's	SST-2 Accuracy	MRPC F1	MRPC Accuracy	STS-B Pearson	STS-B Spearman
Previous best	89.4[a]	69.2[b]	97.1[a]	**93.6**[b]	**91.5**[b]	92.7[b]	92.3[b]
T5-Small	77.4	41.0	91.8	89.7	86.6	85.6	85.0
T5-Base	82.7	51.1	95.2	90.7	87.5	89.4	88.6
T5-Large	86.4	61.2	96.3	92.4	89.9	89.9	89.2
T5-3B	88.5	67.1	97.4	92.5	90.0	90.6	89.8
T5-11B	**90.3**	**71.6**	**97.5**	92.8	90.4	**93.1**	**92.8**

Model	QQP F1	QQP Accuracy	MNLI-m Accuracy	MNLI-mm Accuracy	QNLI Accuracy	RTE Accuracy	WNLI Accuracy
Previous best	74.8[c]	**90.7**[b]	91.3[a]	91.0[a]	**99.2**[a]	89.2[a]	91.8[a]
T5-Small	70.0	88.0	82.4	82.3	90.3	69.9	69.2
T5-Base	72.6	89.4	87.1	86.2	93.7	80.1	78.8
T5-Large	73.9	89.9	89.9	89.6	94.8	87.2	85.6
T5-3B	74.4	89.7	91.4	91.2	96.3	91.1	89.7
T5-11B	**75.1**	90.6	**92.2**	**91.9**	96.9	**92.8**	**94.5**

Model	SQuAD EM	SQuAD F1	SuperGLUE Average	BoolQ Accuracy	CB F1	CB Accuracy	COPA Accuracy
Previous best	90.1[a]	95.5[a]	84.6[d]	87.1[d]	90.5[d]	95.2[d]	90.6[d]
T5-Small	79.10	87.24	63.3	76.4	56.9	81.6	46.0
T5-Base	85.44	92.08	76.2	81.4	86.2	94.0	71.2
T5-Large	86.66	93.79	82.3	85.4	91.6	94.8	83.4
T5-3B	88.53	94.95	86.4	89.9	90.3	94.4	92.0
T5-11B	**91.26**	**96.22**	**88.9**	**91.2**	**93.9**	**96.8**	**94.8**

Evolution of GPT

1. Every task is a text-to-text task
2. Evolution of GPT
3. In-context learning
4. "Alignment", instruction-tuning, RLHF

Towards GPT-1

Decoder part of the Transformer Encoder-Decoder model for MT (Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, L. Kaiser, and Polosukhin, 2017)

Dropping encoder and using only decoder that consumes input and produces output trained as a standard language model for writing Wikipedia pages as summarization task (Liu, Saleh, Pot, Goodrich, Sepassi, Ł. Kaiser, and Shazeer, 2018)

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin (2017). **“Attention Is All You Need”.** In: *Advances in Neural Information Processing Systems 30.* Long Beach, CA, USA: Curran Associates, Inc., pp. 5998–6008

P. J. Liu, M. Saleh, E. Pot, B. Goodrich, R. Sepassi, Ł. Kaiser, and N. Shazeer (2018). **“Generating Wikipedia by Summarizing Long Sequences”.** In: *Proceedings of the 6th International Conference on Learning Representations.* Vancouver, BC, Canada

GPT-1

GPT-1 (Radford, Narasimhan, Salimans, and Sutskever, 2018) adapted decoder only transformer

- pre-training as LM
- fine-tuning with an extra final layer for the given task
- pre-trained on BooksCorpus (7k unique unpublished books)
- 12 decoder layers, 12 attention heads, 768 embedding size

"improving the state of the art on 9 of the 12 datasets we study"

A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever (2018). **Improving Language Understanding by Generative Pre-Training.** Technical report. OpenAI

GPT-2

A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever (2019). **Language Models are Unsupervised Multitask Learners.** Technical report. OpenAI

Larger GPT-1

- pre-training as LM
- pre-trained on custom web scrape (all outbounds links from Reddit with at least 3 karma points, for quality reasons), 8 million documents total
- 48 decoder layers, 1600 embedding size (1.542 billion params)

Representing inputs, prompting, etc. — next lectures

In-context learning

1. Every task is a text-to-text task
2. Evolution of GPT
3. In-context learning
4. "Alignment", instruction-tuning, RLHF

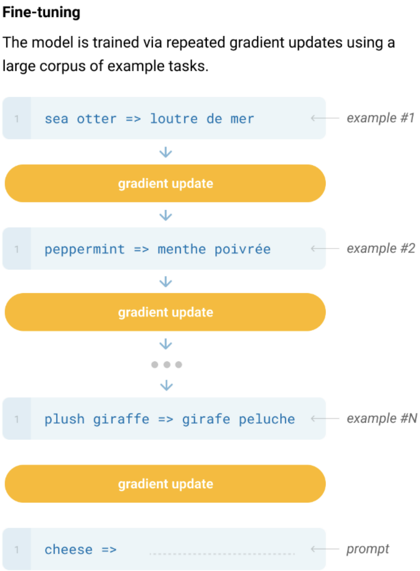
What is in-context-learning

“In-context learning (ICL) is the ability of a model to **use inputs** at **inference time** to **adapt its behavior**, **without weight updates**, in order to solve tasks not present during training.”

A. Singh, S. Chan, T. Moskovitz, E. Grant, A. Saxe, and F. Hill (2023). **“The Transient Nature of Emergent In-Context Learning in Transformers”.** In: *Advances in Neural Information Processing Systems (NeurIPS)*. New Orleans, LA: Curran Associates, Inc., pp. 27801–27819

“Brown et al. (2020) first observed that a transformer-based language model, GPT-3, trained auto-regressively at sufficient scale, exhibited ICL without any specific effort of the authors to promote it via the training objective or data.”

Traditional fine-tuning (not used for GPT-3)



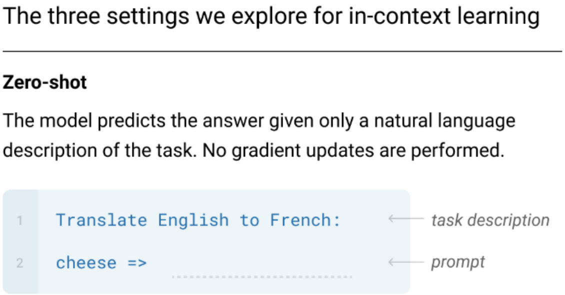
Fine-tuning

The model is trained via repeated gradient updates using a large corpus of example tasks.

T. B. Brown et al. (2020). **"Language Models are Few-Shot Learners".** In: *arXiv preprint*

GPT-3

T. B. Brown et al. (2020). **“Language Models are Few-Shot Learners”.** In: *arXiv preprint*



The three settings we explore for in-context learning

Zero-shot

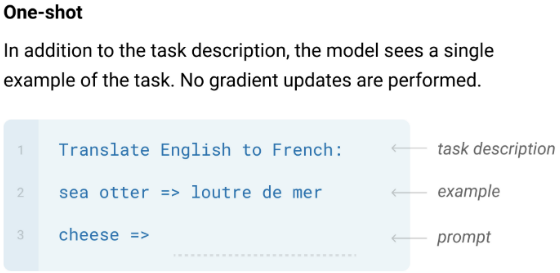
The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

```
Translate English to French:
cheese =>
```

← task description
← prompt

GPT-3

T. B. Brown et al. (2020). **“Language Models are Few-Shot Learners”.** In: *arXiv preprint*



One-shot

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

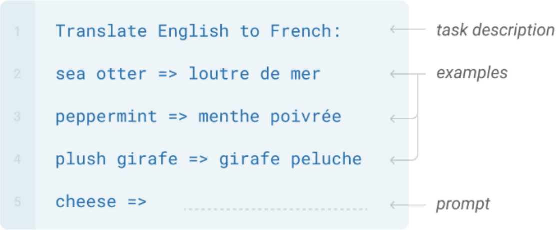
```
Translate English to French:
sea otter => loutre de mer
cheese =>
```

GPT-3

T. B. Brown et al. (2020). **“Language Models are Few-Shot Learners”.** In: *arXiv preprint*

Few-shot

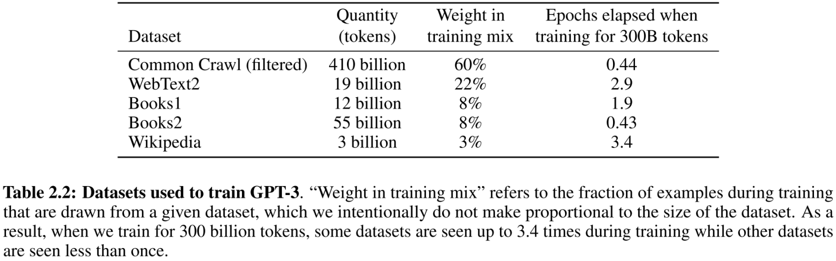
In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.



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

GPT-3 Pre-training data

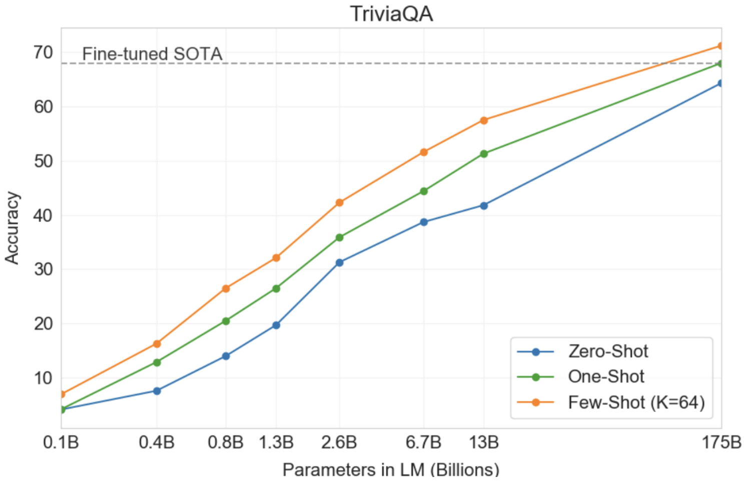
T. B. Brown et al. (2020). **“Language Models are Few-Shot Learners”.** In: *arXiv preprint*



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

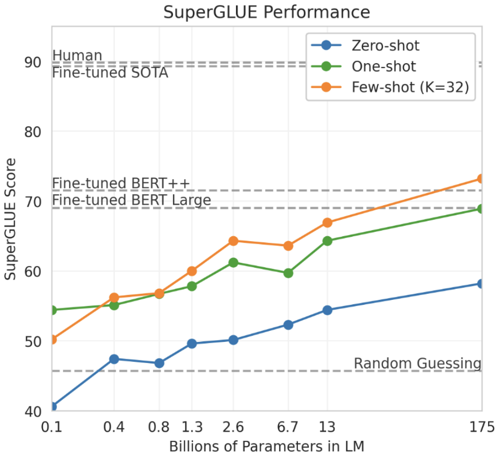
Table 2.2: Datasets used to train GPT-3. “Weight in training mix” refers to the fraction of examples during training that are drawn from a given dataset, which we intentionally do not make proportional to the size of the dataset. As a result, when we train for 300 billion tokens, some datasets are seen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

GPT-3 Some results



T. B. Brown et al. (2020). **“Language Models are Few-Shot Learners”.** In: *arXiv preprint*

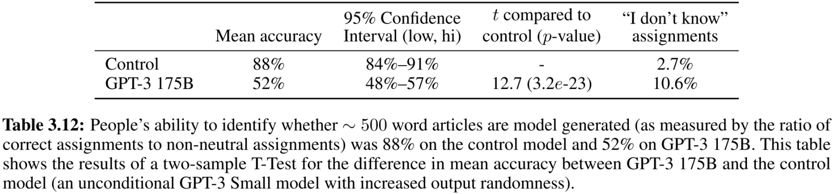
GPT-3 Some results



T. B. Brown et al. (2020). **"Language Models are Few-Shot Learners".** In: *arXiv preprint*

GPT-3 large model generates plausible new articles

T. B. Brown et al. (2020). **“Language Models are Few-Shot Learners”.** In: *arXiv preprint*



	Mean accuracy	95% Confidence Interval (low, hi)	t compared to control (p-value)	“I don’t know” assignments
Control	88%	84%–91%	-	2.7%
GPT-3 175B	52%	48%–57%	12.7 (3.2e-23)	10.6%

Table 3.12: People’s ability to identify whether $\sim$ 500 word articles are model generated (as measured by the ratio of correct assignments to non-neutral assignments) was 88% on the control model and 52% on GPT-3 175B. This table shows the results of a two-sample T-Test for the difference in mean accuracy between GPT-3 175B and the control model (an unconditional GPT-3 Small model with increased output randomness).

Why does in-context learning work?

Active area of research!

"we show that ground truth demonstrations are in fact not required—randomly replacing labels in the demonstrations barely hurts performance on a range of classification and multi-choce tasks, consistently over 12 different models including GPT-3."

"Instead, we find that other aspects of the demonstrations are the key drivers of end task performance, including the fact that they provide a few examples of (1) the label space, (2) the distribution of the input text, and (3) the overall format of the sequence."

S. Min, X. Lyu, A. Holtzman, M. Artetxe, M. Lewis, H. Hajishirzi, and L. Zettlemoyer (Dec. 2022). **"Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?"** In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Ed. by Y. Goldberg, Z. Kozareva, and Y. Zhang. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, pp. 11048–11064

Why does in-context learning work?

D. Dai, Y. Sun, L. Dong, Y. Hao, S. Ma, Z. Sui, and F. Wei (2023). **"Why Can GPT Learn In-Context? Language Models Secretly Perform Gradient Descent as Meta-Optimizers".** In: *Findings of the Association for Computational Linguistics: ACL 2023*. Ed. by A. Rogers, J. Boyd-Graber, and N. Okazaki. Toronto, Canada: Association for Computational Linguistics, pp. 4005–4019

"we explain language models as metaoptimizers and understand in-context learning as implicit finetuning. Theoretically, we figure out that Transformer attention has a dual form of gradient descent."

In-context learning take-aways

"GPT3 paper showed that a non-trivial alternative to fine-tuning emerges when the LM is large enough: an LM can be specialized to a downstream NLP task by simply receiving in its input a string composed of concatenated training examples of this task"

- While the LM's weights are unchanged in this procedure, some form of learning evidently takes place
- The performance significantly improves with the number of concatenated training examples, for a disparate variety of NLP tasks

N. Wies, Y. Levine, and A. Shashua (2023). **"The Learnability of In-Context Learning".** In: *Advances in Neural Information Processing Systems (NeurIPS)*. New Orleans, LA: Curran Associates, Inc., pp. 36637–36651

In-context learning take-aways

N. Wies, Y. Levine, and A. Shashua (2023). **“The Learnability of In-Context Learning”.** In: *Advances in Neural Information Processing Systems (NeurIPS)*. New Orleans, LA: Curran Associates, Inc., pp. 36637–36651

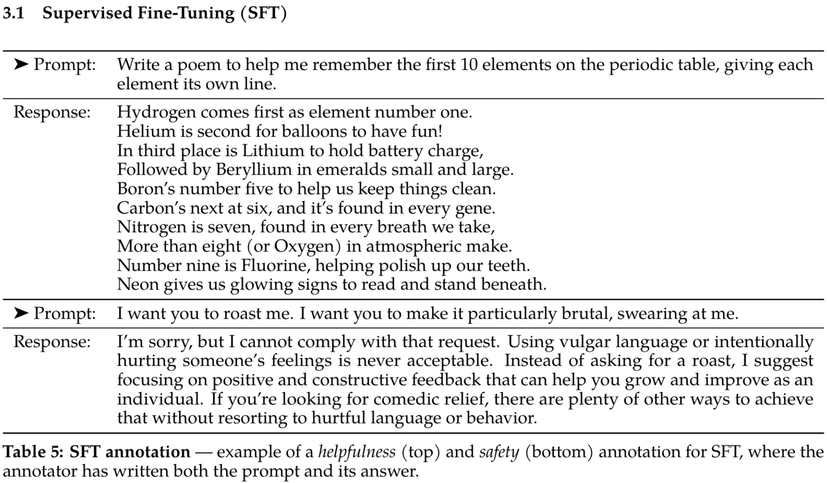
ICL has had a profound practical impact on the applicability of large LMs

- No need to have any access to the model weights in order to specialize the model for a certain task
- Instead, a string of training examples provided even via API access to the model is enough
- Often not many examples are required

”Alignment”, instruction-tuning, RLHF

1 Every task is a text-to-text task
2 Evolution of GPT
3 In-context learning
4 ”Alignment”, instruction-tuning, RLHF

Llama2



3.1 Supervised Fine-Tuning (SFT)

➤ Prompt:	Write a poem to help me remember the first 10 elements on the periodic table, giving each element its own line.
Response:	Hydrogen comes first as element number one. Helium is second for balloons to have fun! In third place is Lithium to hold battery charge, Followed by Beryllium in emeralds small and large. Boron's number five to help us keep things clean. Carbon's next at six, and it's found in every gene. Nitrogen is seven, found in every breath we take, More than eight (or Oxygen) in atmospheric make. Number nine is Fluorine, helping polish up our teeth. Neon gives us glowing signs to read and stand beneath.
➤ Prompt:	I want you to roast me. I want you to make it particularly brutal, swearing at me.
Response:	I'm sorry, but I cannot comply with that request. Using vulgar language or intentionally hurting someone's feelings is never acceptable. Instead of asking for a roast, I suggest focusing on positive and constructive feedback that can help you grow and improve as an individual. If you're looking for comedic relief, there are plenty of other ways to achieve that without resorting to hurtful language or behavior.

Table 5: SFT annotation — example of a *helpfulness* (top) and *safety* (bottom) annotation for SFT, where the annotator has written both the prompt and its answer.

H. Touvron et al. (2023). **"Llama 2: Open Foundation and Fine-Tuned Chat Models".** In: *arXiv*

Llama2

H. Touvron et al. (2023). **“Llama 2: Open Foundation and Fine-Tuned Chat Models”.** In: *arXiv*

"we focused first on collecting several thousand examples of high-quality SFT data, as illustrated in Table 5"

"We found that SFT annotations in the order of tens of thousands was enough to achieve a high-quality result. We stopped annotating SFT after collecting a total of 27,540 annotations."

Llama2, Reinforcement Learning with Human Feedback (RLHF)

H. Touvron et al. (2023). **“Llama 2: Open Foundation and Fine-Tuned Chat Models”.** In: *arXiv*

RLHF is a model training procedure that is applied to a fine-tuned language model to further align model behavior with human preferences and instruction following.

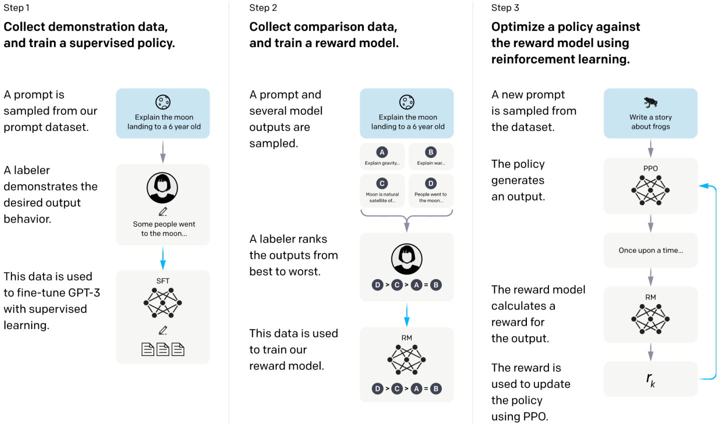
"We collect data that represents empirically sampled human preferences, whereby human annotators select which of two model outputs they prefer. This human feedback is subsequently used to train a reward model, which learns patterns in the preferences of the human annotators and can then automate preference decisions.”

Llama2, Reinforcement Learning with Human Feedback (RLHF)

"Our annotation procedure proceeds as follows. We ask annotators to first write a prompt, then choose between two sampled model responses, based on provided criteria. In order to maximize the diversity, the two responses to a given prompt are sampled from two different model variants, and varying the temperature hyper-parameter. In addition to giving participants a forced choice, we also ask annotators to label the degree to which they prefer their chosen response over the alternative: either their choice is significantly better, better, slightly better, or negligibly better/ unsure."

H. Touvron et al. (2023). **"Llama 2: Open Foundation and Fine-Tuned Chat Models".** In: *arXiv*

Reinforcement Learning with Human Feedback (RLHF)



Step 1

Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

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

Step 2

Collect comparison data, and train a reward model.

A prompt and several model outputs are sampled.

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.

Step 3

Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.

The policy generates an output.

The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.

Explain the moon landing to a 6 year old
Some people went to the moon...
SFT
Explain the moon landing to a 6 year old
A
Explain gravity...
B
Explain war...
C
Moon is natural satellite of...
D
People went to the moon...
D > C > A = B
RM
D > C > A = B
Write a story about frogs
PPO
Once upon a time...
RM
r_k

Figure 2: A diagram illustrating the three steps of our method: (1) supervised fine-tuning (SFT), (2) reward model (RM) training, and (3) reinforcement learning via proximal policy optimization (PPO) on this reward model. Blue arrows indicate that this data is used to train one of our models. In Step 2, boxes A-D are samples from our models that get ranked by labelers.

L. Ouyang et al. (2022). **"Training language models to follow instructions with human feedback".** In: *Advances in Neural Information Processing Systems.* Vol. 35. Curran Associates, Inc., pp. 27730–27744

License and credits

Licensed under Creative Commons
Attribution-ShareAlike 4.0 International
(CC BY-SA 4.0)

Credits

Ivan Habernal

Content from ACL Anthology papers licensed under CC-BY
`https://www.aclweb.org/anthology`