



Redes Neuronales para Lenguaje Natural

2023

Grupo de Procesamiento de Lenguaje Natural
Instituto de Computación



Generative Pre-trained Transformers

Improving language understanding by generative pre-training.
Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018).

Language models are few-shot learners.
Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020).
Advances in neural information processing systems, 33, 1877-1901.

Training language models to follow instructions with human feedback
Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., ... & Lowe, R. (2022). URL
<https://arxiv.org/abs/2203.02155>, 13.

Llama 2: Open foundation and fine-tuned chat models.
Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., ... & Scialom, T. (2023). arXiv
preprint arXiv:2307.09288.

State of GPT. Andrej Karpathy. (2023)

<https://build.microsoft.com/en-US/sessions/db3f4859-cd30-4445-a0cd-553c3304f8e2>

Motivación

Hay muchas tareas que se resuelven con **aprendizaje supervisado**
(Ej. Sentiment, Hatespeech, Humor, Translation, Summarization, QA, SRL, POS, ...)

Los **datos anotados** en general son escasos y costosos.

Sería bueno usar datos no anotados para aprovechar mejor los datos anotados.

Receta para el éxito (por ahora): **pre-training + fine-tuning**

Transformer (BERT y GPT)

El modelo **transformer** mejora el tratamiento de secuencias en comparación con RNNs y CNNs

El **encoder pre-entrenado** (ej. MLM) puede ser utilizado en diversos tipos de tareas mediante un proceso de **fine-tuning** (ej. **BERT**)

El **decoder pre-entrenado** también puede ser utilizado con éxito mediante un proceso de **fine-tuning** en diversas tareas.

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).

Improving Language Understanding by Generative Pre-Training

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

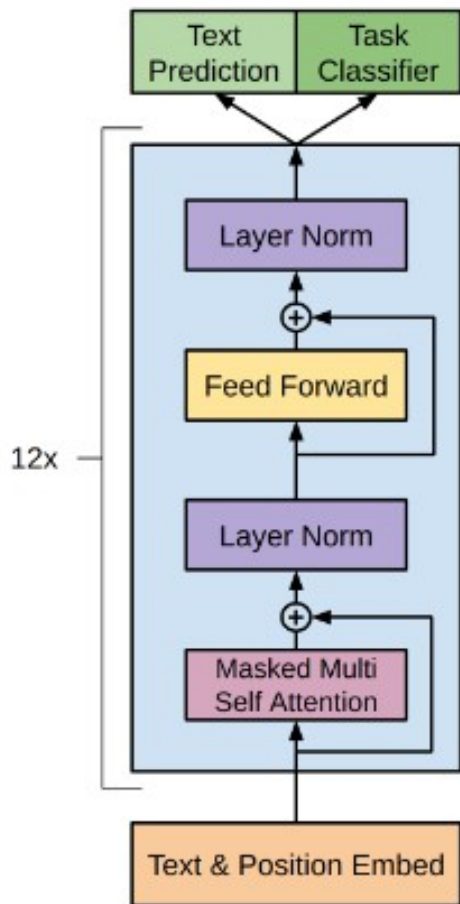
Transformer (el **decoder** en nomenclatura de “Attention is all you need”)

- **Causal Transformer** en la nomenclatura de Jurafsky y Martin 3
- **Generative Pre-trained Transformer** es otro nombre posible
(bloques de Transformer con **masked** self-attention)

Pre-training + fine-tuning para un conjunto de tareas:

- Natural Language Inference
- Question Answering
- Sentence Similarity
- Sentence Classification

Arquitectura



12 bloques de Transformer (768 dimensión de salida)

12 masked attention heads

Position-wise Feed Forward (3072 hidden dimension)

GELU (Gaussian Error Linear Unit) activation

Bytepair (BPE) encoding

Learned position embeddings

$$h_0 = UW_e + W_p$$

$$h_l = \text{transformer_block}(h_{l-1}) \forall i \in [1, n]$$

$$P(u) = \text{softmax}(h_n W_e^T)$$

Pre-training (unsupervised)

Función objetivo standard de modelo de lenguaje (predecir próximo token), maximizar:

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

Se ajustan los pesos con Adam, learning rate entre 0 y 2.5e-4 (oscilando)

100 epochs (minibatches de 64 randomly 512 tokens consecutivos)

Fine-tuning (supervised)

Función objetivo de clasificación de secuencias:

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

Se implementa usando la salida del último token de la secuencia

$$P(y|x^1, \dots, x^m) = \text{softmax}(h_l^m W_y)$$

Fine-tuning (supervised)

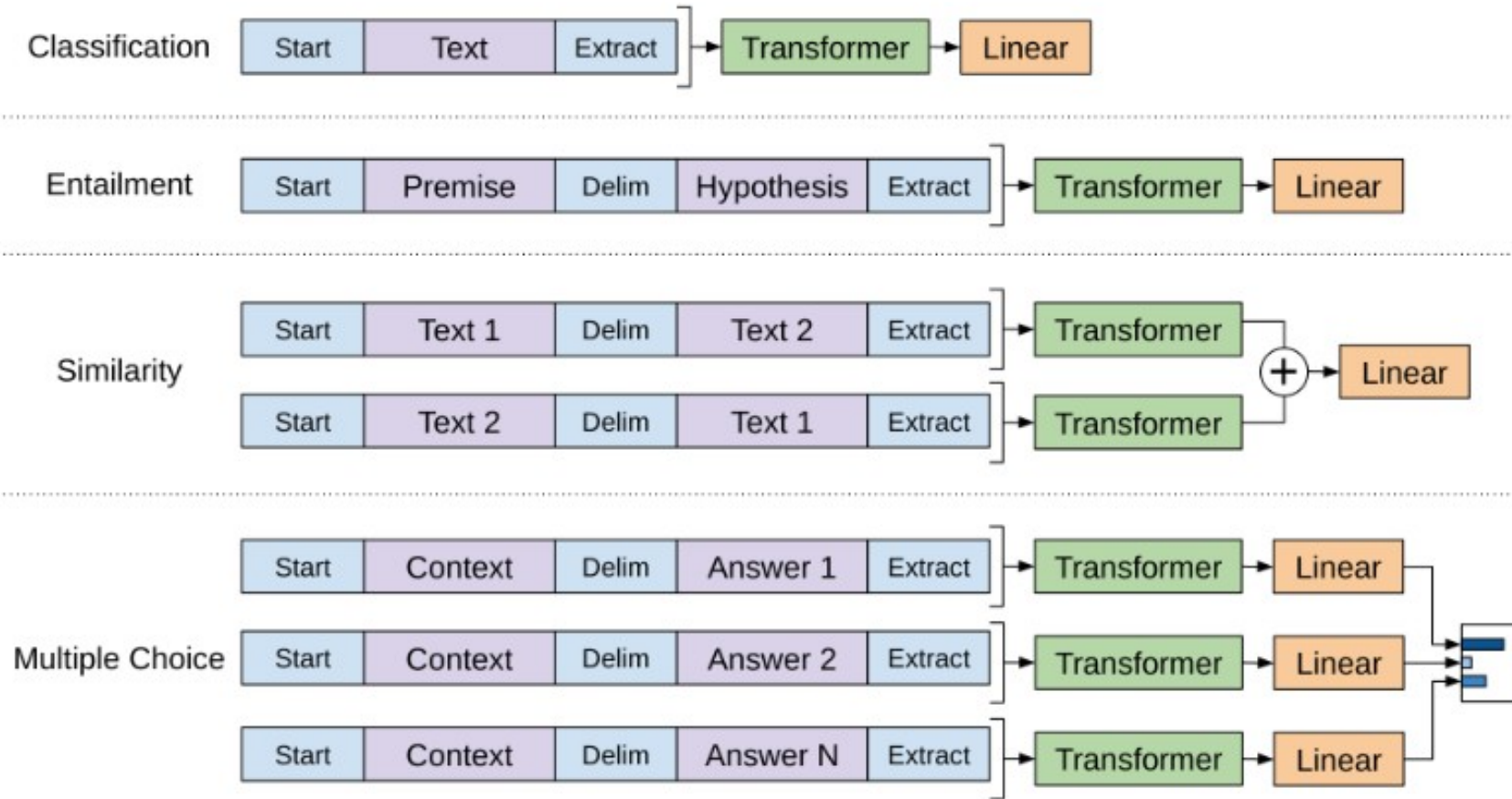
Incluir la función objetivo de modelo de lenguaje durante el fine-tuning mejora los resultados y acelera la convergencia.

La función objetivo final para el fine-tuning es:

$$L_3(\mathcal{C}) = L_2(\mathcal{C}) + \lambda * L_1(\mathcal{C})$$

Se ajustan los pesos con Adam, learning rate 6.25e-5 (decay), batchsize de 32
3 epochs. $\lambda = 0.5$

Fine-tuning (formato de entrada)



Fine-tuning – Tareas

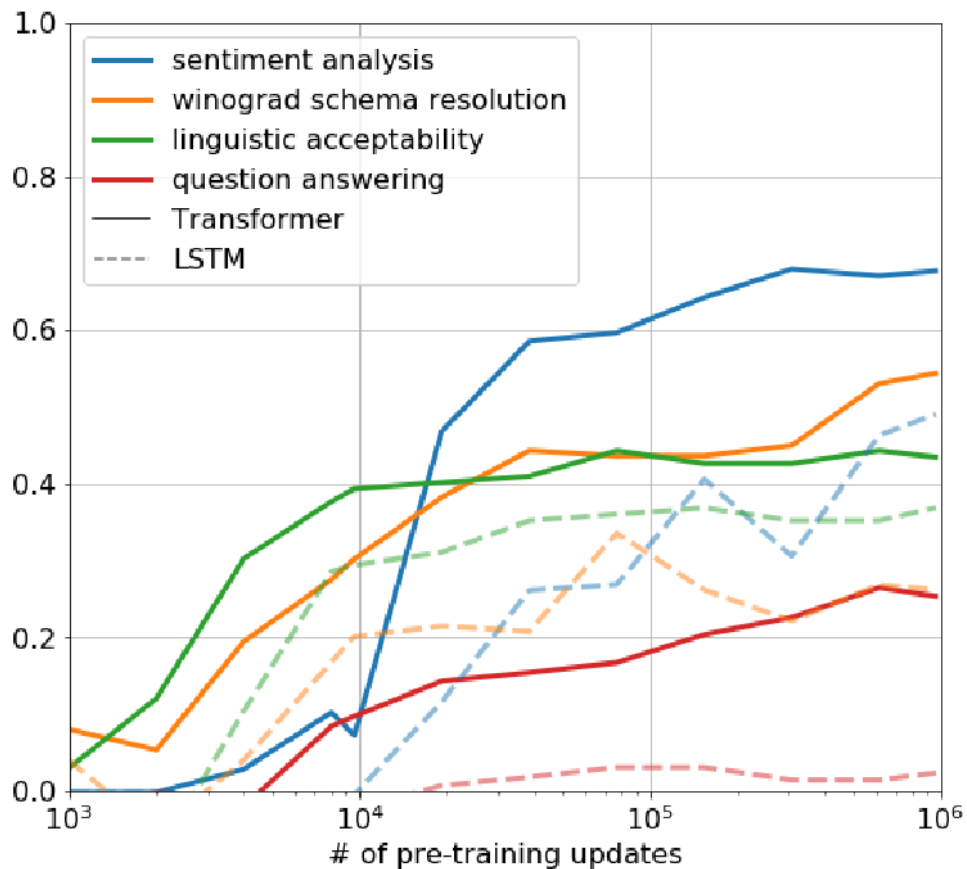
Task	Datasets
Natural language inference	SNLI [5], MultiNLI [66], Question NLI [64], RTE [4], SciTail [25]
Question Answering	RACE [30], Story Cloze [40]
Sentence similarity	MSR Paraphrase Corpus [14], Quora Question Pairs [9], STS Benchmark [6]
Classification	Stanford Sentiment Treebank-2 [54], CoLA [65]

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	-	-	82.1	61.7
Finetuned Transformer LM (ours)	82.1	81.4	89.9	88.3	88.1	56.0

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)	86.5	62.9	57.4	59.0

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

Pre-trained model zero-shot performance



Observan mejoras durante el pre-entrenamiento en distintas tareas (sin fine-tuning)

Zero-shot lo definen con heurísticas para cada tarea (ej. para sentiment agregan la palabra 'very' al final y toman las probabilidades de las palabras 'positive' y 'negative')

Estudio de ablación

El estudio de ablación muestra la ganancia obtenida con el pre-entrenamiento.

La importancia del LM en el fine-tuning loss es dudosa :)

Method	Avg. Score	CoLA (mc)	SST2 (acc)	MRPC (F1)	STSB (pc)	QQP (F1)	MNLI (acc)	QNLI (acc)	RTE (acc)
Transformer w/ aux LM (full)	74.7	45.4	91.3	82.3	82.0	70.3	81.8	88.1	56.0
Transformer w/o pre-training	59.9	18.9	84.0	79.4	30.9	65.5	75.7	71.2	53.8
Transformer w/o aux LM	75.0	47.9	92.0	84.9	83.2	69.8	81.1	86.9	54.4
LSTM w/ aux LM	69.1	30.3	90.5	83.2	71.8	68.1	73.7	81.1	54.6

(*mc*= Mathews correlation, *acc*=Accuracy, *pc*=Pearson correlation)

LSTM es 1 capa 2048 unidades.

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

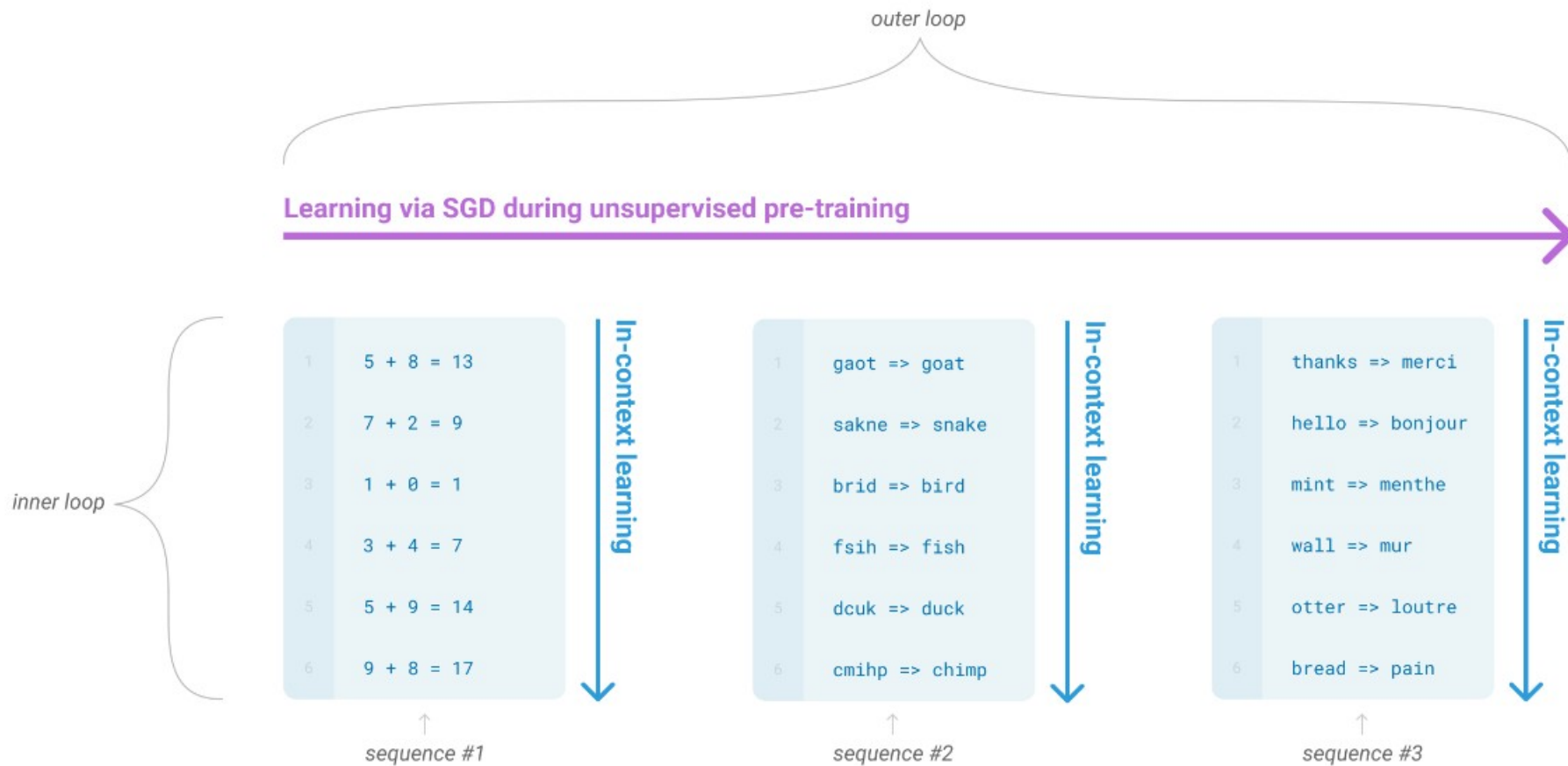
Abstract

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, using a novel word in a sentence, or performing 3-digit arithmetic. At the same time, we also identify some datasets where GPT-3’s few-shot learning still struggles, as well as some datasets where GPT-3 faces methodological issues related to training on large web corpora. Finally, we find that GPT-3 can generate samples of news articles which human evaluators have difficulty

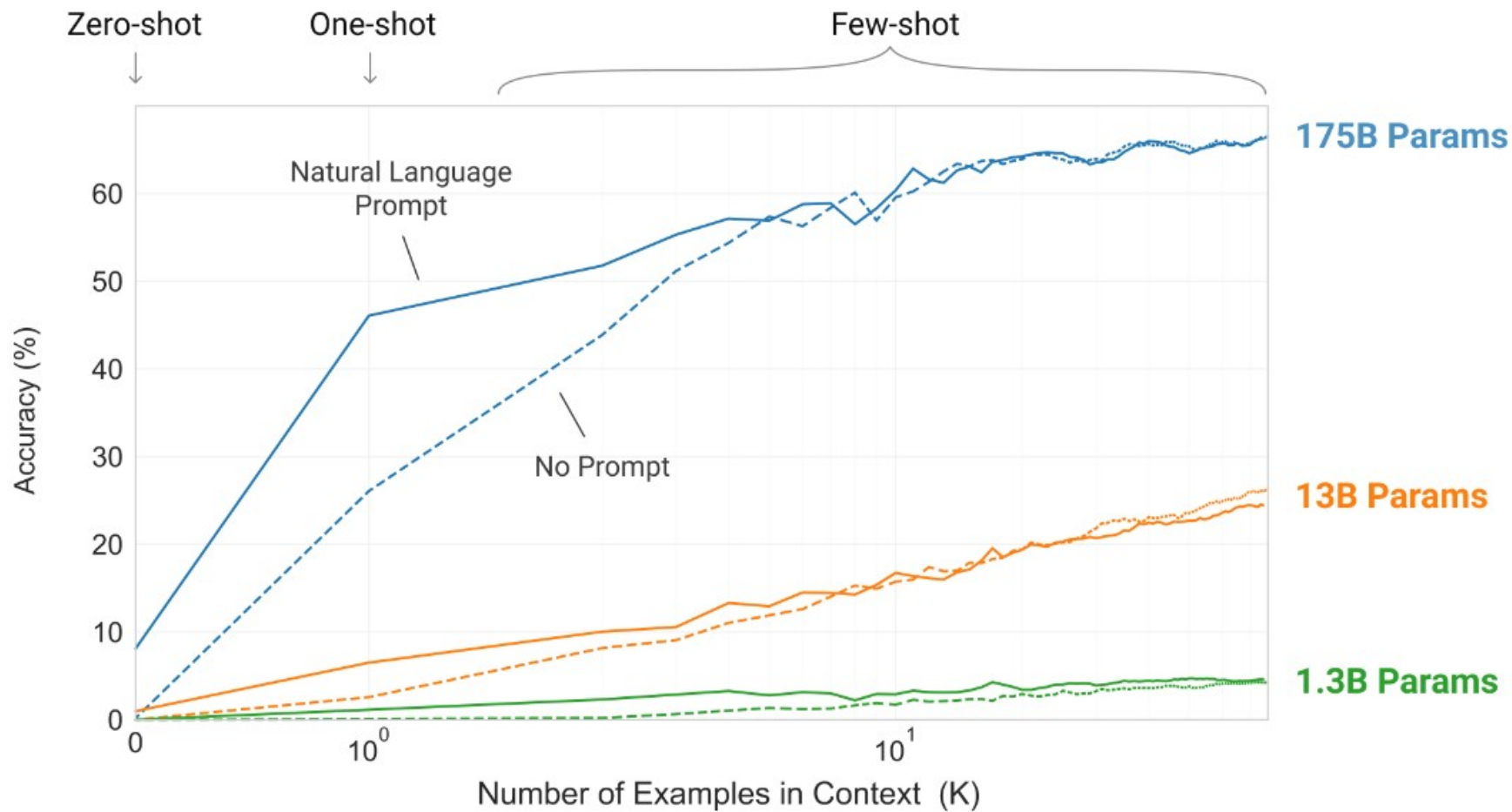
Recent work has demonstrated substantial gains on many NLP tasks and benchmarks by pre-training on a large corpus of text followed by fine-tuning on a specific task. While typically task-agnostic in architecture, this method still requires task-specific fine-tuning datasets of thousands or tens of thousands 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, using a novel word in a sentence, or performing 3-digit arithmetic. At the same time, we also identify some datasets where GPT-3's few-shot learning still struggles, as well as some datasets where GPT-3 faces methodological issues related to training on large web corpora. Finally, we find that GPT-3 can generate samples of news articles which human evaluators have difficulty distinguishing from articles written by humans. We discuss broader societal impacts of this finding and of GPT-3 in general.

In-context learning (meta-learning)

(inner loop vs outer loop)



Cant. parámetros y few-shot in-context learning



GPT-1 to GPT-3

2018

GPT-1 / 117M params / 600B corpus.

2019

GPT-2 / 1.5B params / 450B corpus (mejorado).

2020

GPT-3 / 175B params / 300B training (1.5T ?)

2022

GPT-3.5 / 175B params / larger corpus than GPT-3

2023

GPT-4 / ~100T params / 100T corpus / multimodal

GPT-3

175B params, training for 300B tokens, 2048 tokens de contexto

Se evaluaron 3 enfoques:

- a. few-shot** (10 a 100 ejemplos) + task_description
- b. one-shot** + task_description
- c. zero-shot**, solamente una descripción de la tarea

Few, One and Zero-shot learning

Few-shot

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

```
1 Translate English to French: ← task description
2 sea otter => loutre de mer ← examples
3 peppermint => menthe poivrée ←
4 plush girafe => girafe peluche ←
5 cheese => ..... ← prompt
```

One-shot

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

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

Zero-shot

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

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

GPT-3

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}

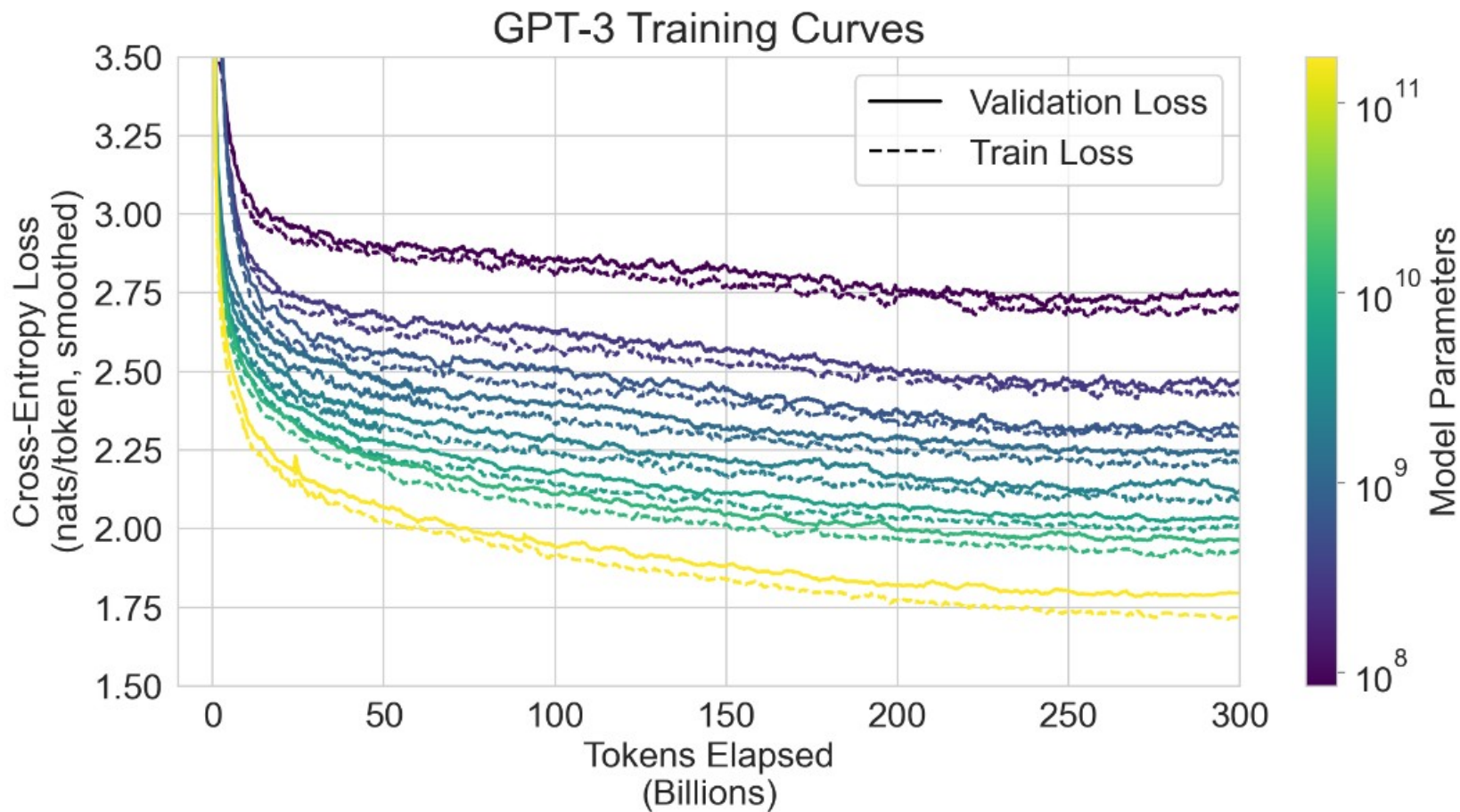
Table 2.1: Sizes, architectures, and learning hyper-parameters (batch size in tokens and learning rate) of the models which we trained. All models were trained for a total of 300 billion tokens.

GPT-3 Training corpus

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 Training curve



GPT-3 Training time

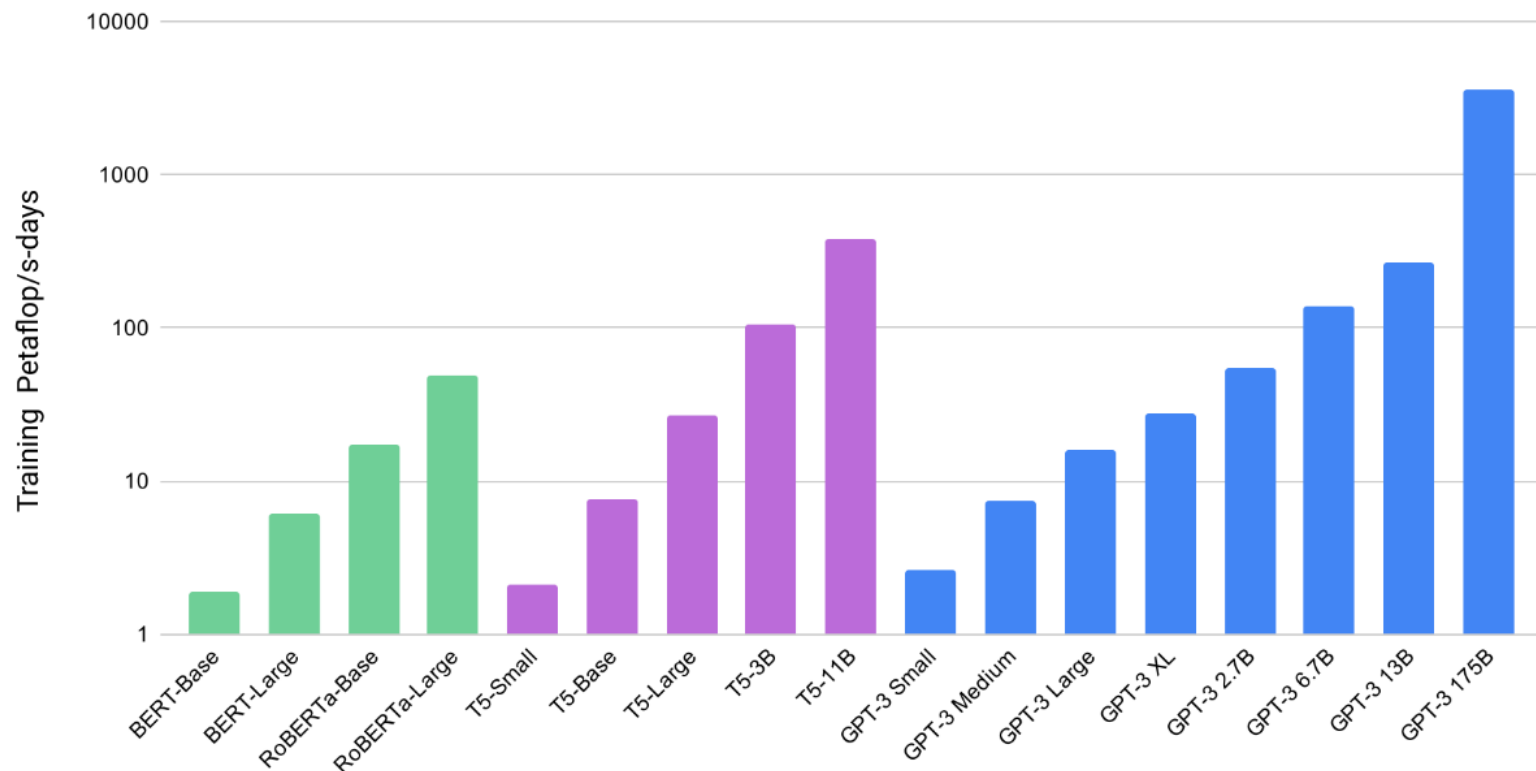


Figure 2.2: Total compute used during training. Based on the analysis in Scaling Laws For Neural Language Models [KMH⁺20] we train much larger models on many fewer tokens than is typical. As a consequence, although GPT-3 3B is almost 10x larger than RoBERTa-Large (355M params), both models took roughly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

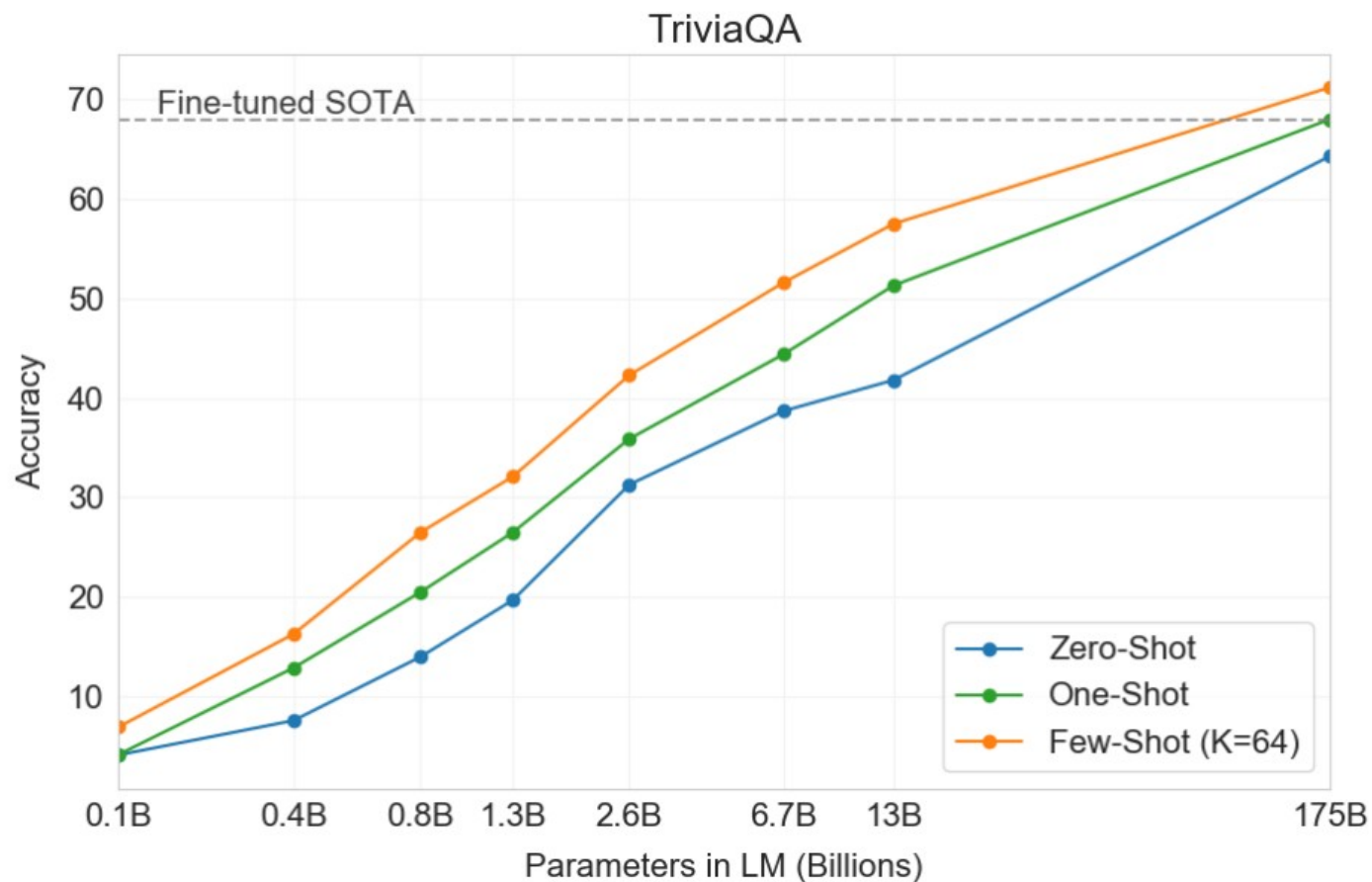
GPT-3 Resultados QA

Open-Domain QA: realiza una consulta para obtener texto donde pueda estar la respuesta

Closed-Book QA: se genera la respuesta sin texto auxiliar

Setting	NaturalQS	WebQS	TriviaQA
RAG (Fine-tuned, Open-Domain) [LPP+20]	44.5	45.5	68.0
T5-11B+SSM (Fine-tuned, Closed-Book) [RRS20]	36.6	44.7	60.5
T5-11B (Fine-tuned, Closed-Book)	34.5	37.4	50.1
GPT-3 Zero-Shot	14.6	14.4	64.3
GPT-3 One-Shot	23.0	25.3	68.0
GPT-3 Few-Shot	29.9	41.5	71.2

GPT-3 Resultados QA

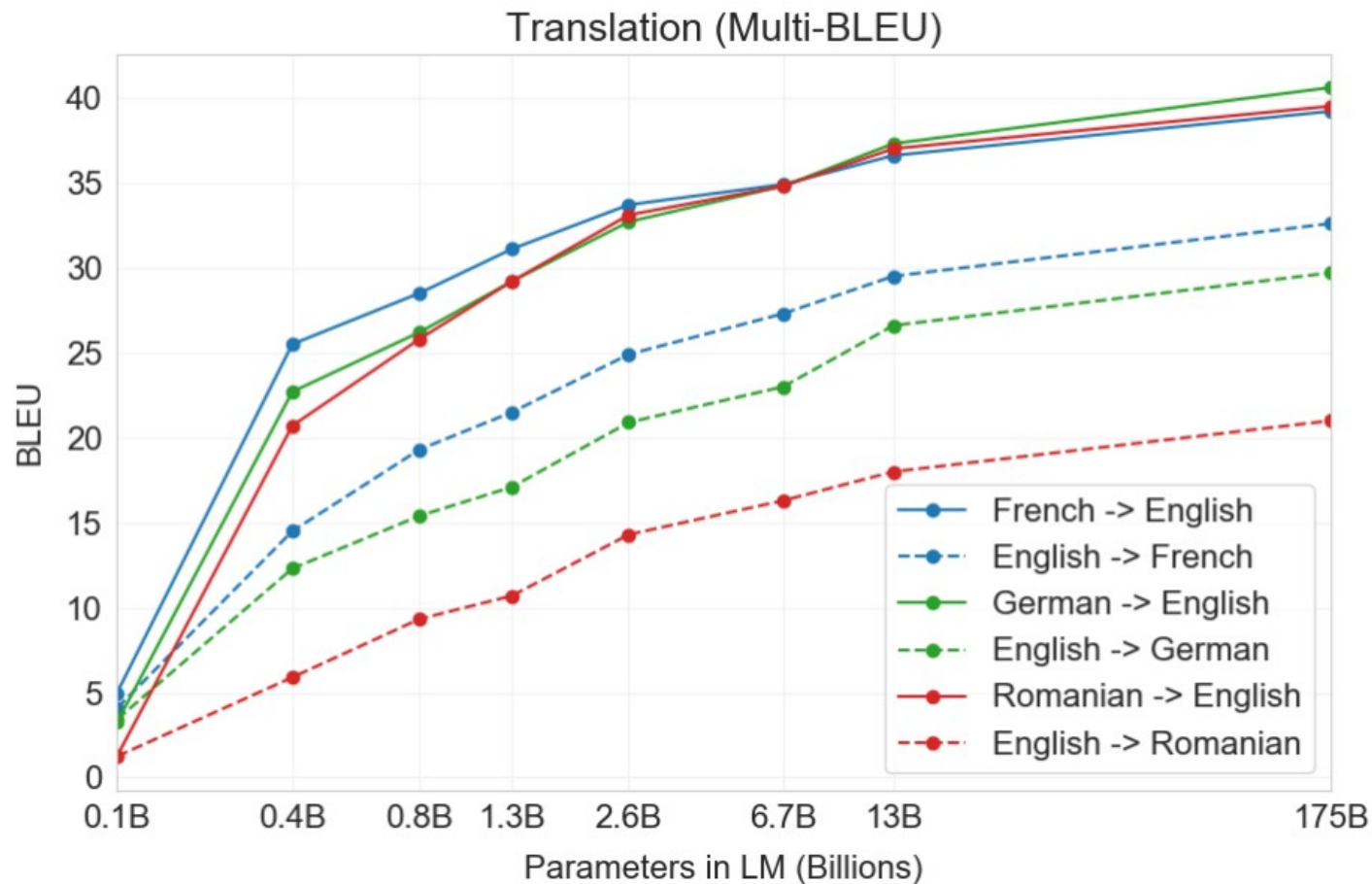


GPT-3 Resultados Traducción

93% (word count) en Inglés, 7% en otros idiomas.

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

GPT-3 Resultados Traducción



GPT-3 Resultados Comprensión Lectora

CoQA free-form conversational dataset

DROP discrete reasoning and numeracy

QuAC structured dialog teacher-student interactions

Setting	CoQA	DROP	QuAC	SQuADv2	RACE-h	RACE-m
Fine-tuned SOTA	90.7^a	89.1^b	74.4^c	93.0^d	90.0^e	93.1^e
GPT-3 Zero-Shot	81.5	23.6	41.5	59.5	45.5	58.4
GPT-3 One-Shot	84.0	34.3	43.3	65.4	45.9	57.4
GPT-3 Few-Shot	85.0	36.5	44.3	69.8	46.8	58.1

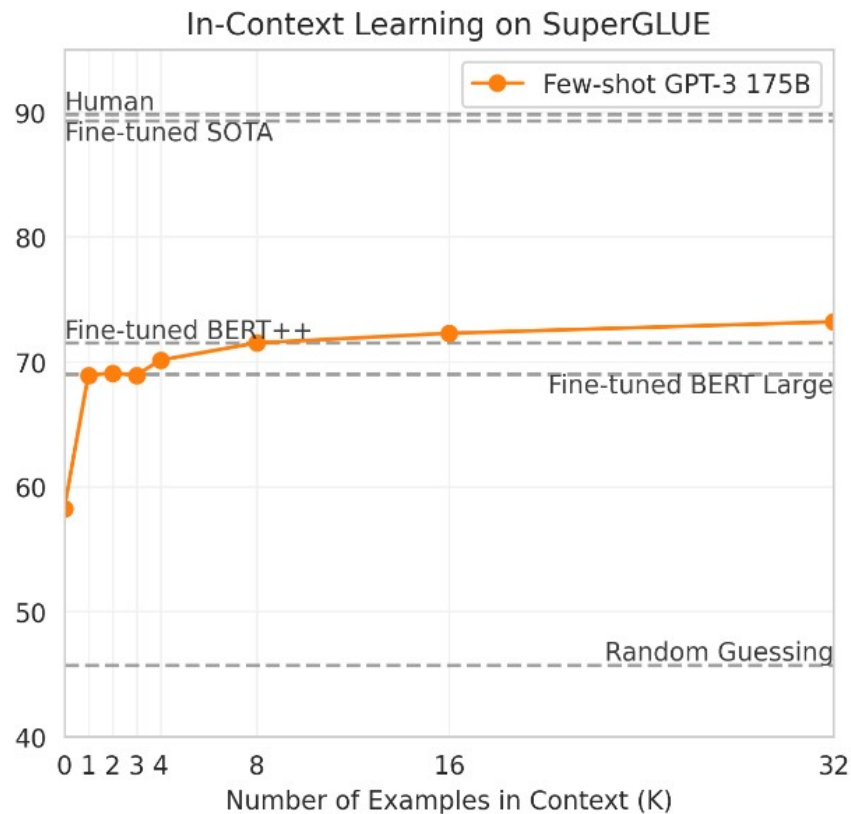
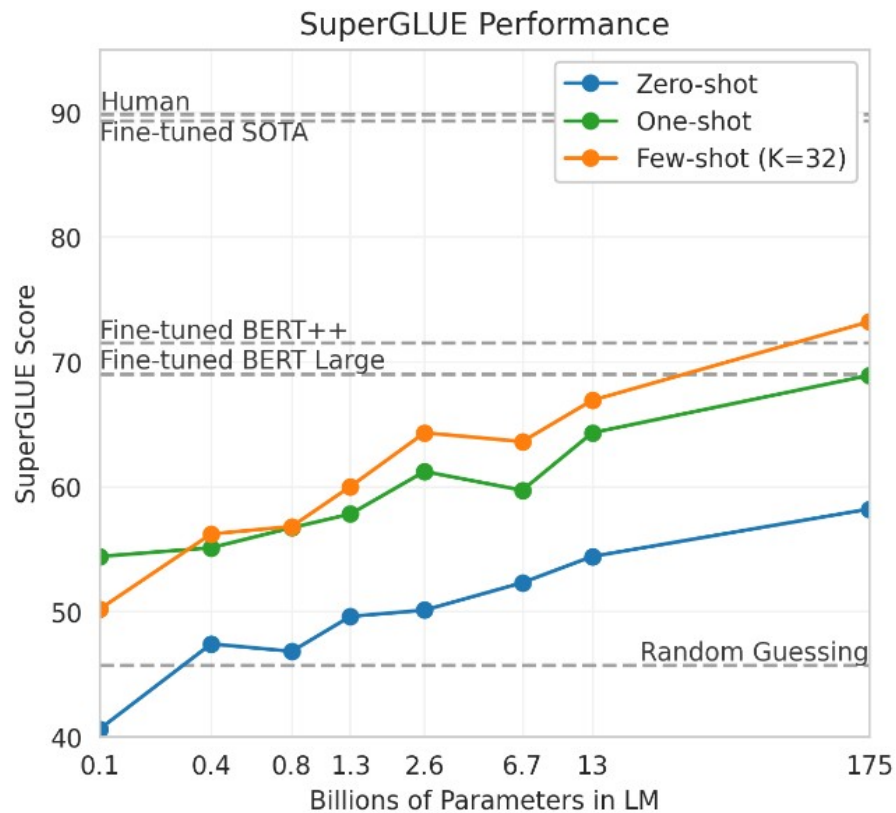
Table 3.7: Results on reading comprehension tasks. All scores are F1 except results for RACE which report accuracy.
^a[JZC⁺19] ^b[JN20] ^c[AI19] ^d[QIA20] ^e[SPP⁺19]

GPT-3 Resultados SuperGLUE

	SuperGLUE Average	BoolQ Accuracy	CB Accuracy	CB F1	COPA Accuracy	RTE Accuracy
Fine-tuned SOTA	89.0	91.0	96.9	93.9	94.8	92.5
Fine-tuned BERT-Large	69.0	77.4	83.6	75.7	70.6	71.7
GPT-3 Few-Shot	71.8	76.4	75.6	52.0	92.0	69.0

	WiC Accuracy	WSC Accuracy	MultiRC Accuracy	MultiRC F1a	ReCoRD Accuracy	ReCoRD F1
Fine-tuned SOTA	76.1	93.8	62.3	88.2	92.5	93.3
Fine-tuned BERT-Large	69.6	64.6	24.1	70.0	71.3	72.0
GPT-3 Few-Shot	49.4	80.1	30.5	75.4	90.2	91.1

GPT-3 Resultados SuperGLUE



GPT-3 Resultados Arithmetico

