LLM 101

一起入门大语言模型

https://llm101.top

哔哩哔哩@一万篇论文笔记

一万篇论文笔记

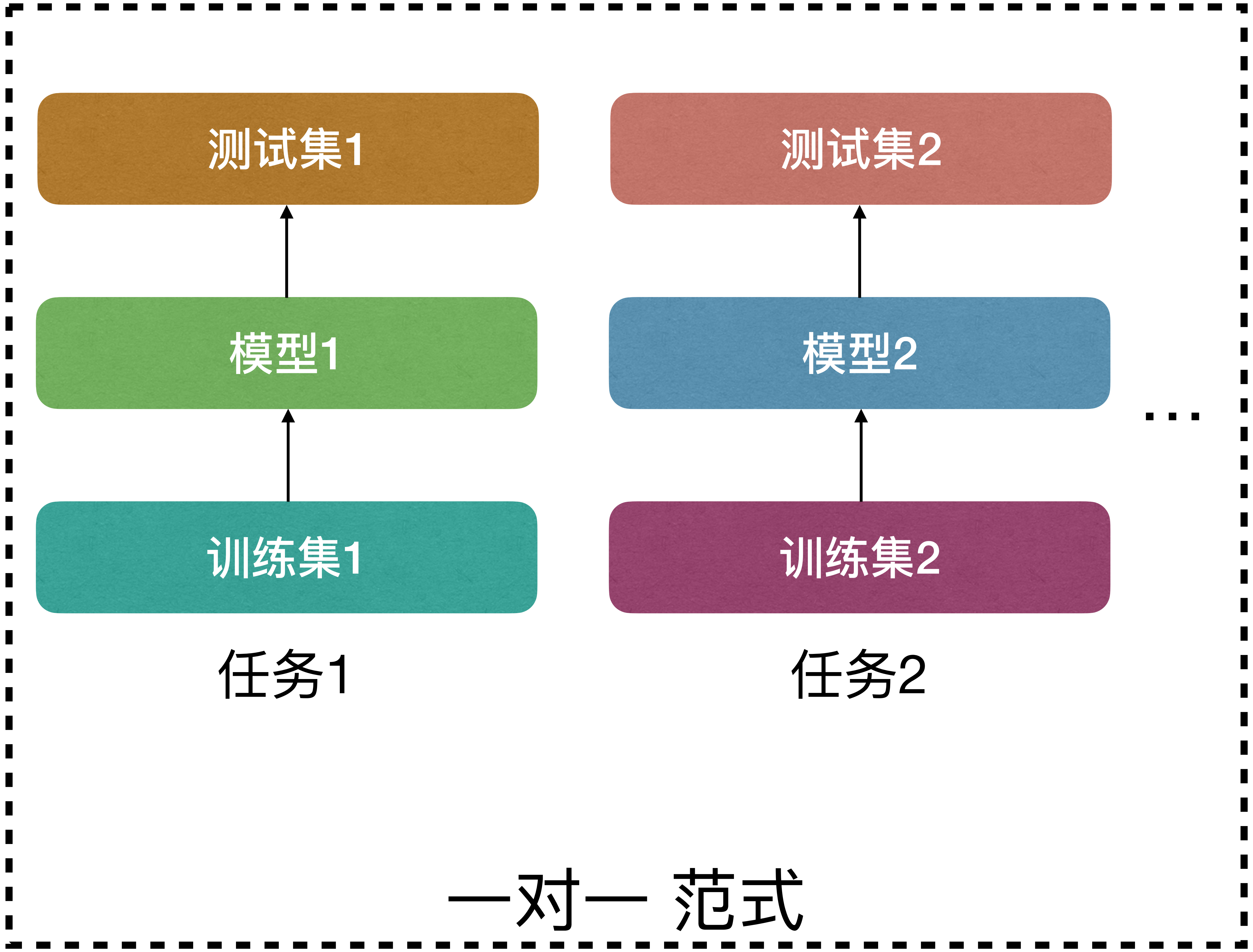
微信扫描二维码，关注我的公众号

2024.12.16

LLM Pre-training and Beyond

- GPT-1 && GPT-2
    - NLP中的预训练-微调范式: CoVe、ELMo、ULMFiT、GPT-1、BERT
    - GPT-1 && GPT-2: Transformer LM + Large scale pre-training ==> zero-shot
    - 编程实践：阅读gpt-1/gpt-2代码；训练124M GPT-2 **llm.c** **Modded-NanoGPT**
- GPT-3 and Beyond
    - Scaling Laws、涌现、幻觉、位置编码、合成数据、提示工程、SLMs ...

一对一 范式



- 每个任务都需要专门设计的模型结构、数据集和模型训练
- 不同任务之间的模型独立

预训练-微调 范式


测试集1
测试集2
模型1
模型2
训练集1
训练集2
任务1
任务2
...
一对一 范式
测试集1
测试集2
预训练模型
训练集1
训练集2
任务1
任务2
...
预训练-微调 范式

预训练-微调 范式

- 预训练-微调**词向量**范式
  - 静态(static)词向量：Word2vec(2013)、Glove(2014)
  - 动态(contextualized)词向量：CoVe(2017)、ELMo(2018) 笔记

笔记

预训练-微调 范式

- 预训练-微调**词向量**范式
  - 静态(static)词向量：Word2vec(2013)、Glove(2014)
  - 动态(contextualized)词向量：CoVe(2017)、ELMo(2018) 笔记

笔记

特征

- 预训练-微调**模型**范式

笔记

  - 任务特定(task-specific)：LM-LSTM(2015)、byte mLSTM(2017)
  - 通用(general): ULMFiT(2018)

笔记

预训练-微调 范式

- 预训练-微调**词向量**范式
  - 静态(static)词向量：Word2vec(2013)、Glove(2014)
  - 动态(contextualized)词向量：CoVe(2017)、ELMo(2018) 笔记

笔记

- 预训练-微调**模型**范式

笔记

  - 任务特定(task-specific)：LM-LSTM(2015)、byte mLSTM(2017)
  - 通用(general): ULMFiT(2018)

笔记

特征

LSTM LM

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

How scaling ?

预训练-微调 范式

- 预训练-微调**词向量**范式
  - 静态(static)词向量： Word2vec(2013)、Glove(2014)
  - 动态(contextualized)词向量： CoVe(2017)、ELMo(2018) 笔记

    笔记
- 预训练-微调**模型**范式

    笔记
  - 任务特定(task-specific)： LM-LSTM(2015)、byte mLSTM(2017)
  - 通用(general): ULMFiT(2018)

    笔记

特征

LSTM LM

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

LSTM MoE(2017)

预训练-微调 范式

- 预训练-微调**词向量**范式
  - 静态(static)词向量：Word2vec(2013)、Glove(2014)
  - 动态(contextualized)词向量：CoVe(2017)、ELMo(2018) 笔记

    笔记
- 预训练-微调**模型**范式
  - 笔记

    任务特定(task-specific)：LM-LSTM(2015)、byte mLSTM(2017)
  - 通用(general): ULMFiT(2018)、GPT-1(2018)、BERT(2018)

    笔记 Decoder-only Transformer Transformer Encoder

特征

LSTM LM

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

LSTM MoE(2017)

GPT-1 vs BERT

- 预训练数据集，BookCorpus(~800M words) vs BookCorpus(800M words) + English Wikipedia(2500M words)
- 参数量，GPT-1(~117M) 和$BERT_{BASE}$(~110M)都是12层$\{MHSA, FFN\}$

GPT-1 vs BERT 参数量

- 上下文窗口长度，均是512
- 位置编码，均是学习的
- 分词，char-level BPE (40478) vs wordpiece (30522)
- batch size, 64 * 512 = 32K vs 256 * 512 = 128K
- 预训练tokens数量，100 epoch * 800M = 80B(words) vs 1M step(~40 epoch) * 3.3B = 132B(words
- 预训练时间，1m on 8GPUs vs 4d on16 TPUs
- 下游任务微调，效果BERT > GPT-1

System	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	**Average**
	392k	363k	108k	67k	8.5k	5.7k	3.5k	2.5k	-
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
$BERT_{BASE}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$BERT_{LARGE}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

GPT-1 vs BERT

- 预训练数据集，BookCorpus(~800M words) vs BookCorpus(800M words) + English Wikipedia(2500M words)
- 参数量，GPT-1(~117M) 和$BERT_{BASE}$(~110M)都是12层$\{MHSA, FFN\}$
- 上下文窗口长度，均是512
- 位置编码，均是学习的
- 分词，char-level BPE (40478) vs wordpiece (30522)
- batch size, 64 * 512 = 32K vs 256 * 512 = 128K
- 预训练tokens数量，100 epoch * 800M = 80B(words) vs 1M step(~40 epoch) * 3.3B = 132B(words
- 预训练时间，1m on 8GPUs vs 4d on16 TPUs
- ~~下游任务微调~~，效果BERT > GPT-1

GPT-1 vs BERT 参数量

System	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	Average
	392k	363k	108k	67k	8.5k	5.7k	3.5k	2.5k	-
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
$BERT_{BASE}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$BERT_{LARGE}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

GPT-1 vs BERT

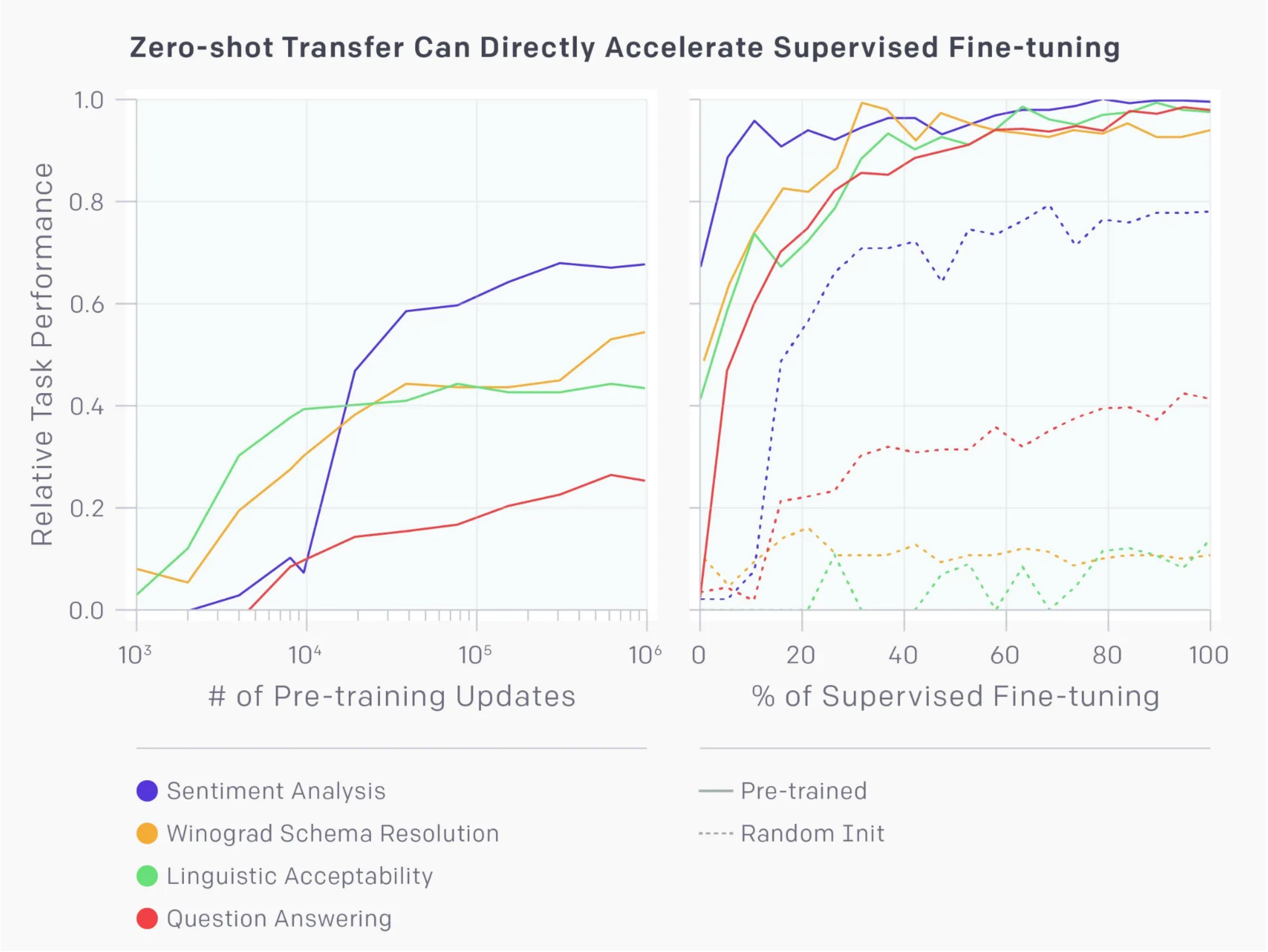
- 预训练数据集，BookCorpus(~800M words) vs BookCorpus(800M words) + English Wikipedia(2500M words)
- 参数量，GPT-1(~117M) 和$BERT_{BASE}$(~110M)都是12层$\{MHSA, FFN\}$

GPT-1 vs BERT 参数量

- 上下文窗口长度，均是512
- 位置编码，均是学习的
- 分词，char-level BPE (40478) vs wordpiece (30522)
- batch size, 64 * 512 = 32K vs 256 * 512 = 128K
- 预训练tokens数量，100 epoch * 800M = 80B(words) vs 1M step(~40 epoch) * 3.3B = 132B(words
- 预训练时间，1m on 8GPUs vs 4d on16 TPUs
- **zero-shot**

GPT-1 Zero-shot

- CoLA (linguistic acceptability),
  - example scored as the average token log-p
  - predictions are made by thresholding
- SST-2 (sentiment analysis)
  - append the token “very” to each example
  - restrict LM’s output only “positive”“negative”
- ……



图片来源

语言模型作为预训练任务的优势

GPT-2: bigger GPT-1 + totally zero-shot

Language Models are Unsupervised Multitask Learners

- 预训练数据集，Bookcorpus(~5GB, 800M words) —> WebText(8M pages, ~40GB, 10B tokens)
- 参数量，117M —> 124M、355M、774M、1.5B
- 上下文窗口长度，512 —> 1024
- 位置编码，均是学习的
- 分词：
  - Char-level BPE —> byte-level BPE
  - 4w merges(40478) —> 5w merges(50257)
- batch size, 64 * 512 = 32K —> 512 * 1024 = 512K
- layernorm位置，$LayerNorm(x + SubLayer(x))$ $x + SubLayer(LayerNorm(x))$

Just a language model,

predicts everything

Byte-level BPE

- 每个char(字母,汉字,符号,😊…)都有一个code point
  - > 130000
- 根据utf-8编码规则，将code point转为二进制格式(1B~4B)

Code point ↔ UTF-8 conversion

First code point	Last code point	Byte 1	Byte 2	Byte 3	Byte 4
U+0000	U+007F	0yyyzzzz			
U+0080	U+07FF	110xxxyy	10yyzzzz		
U+0800	U+FFFF	1110wwww	10xxxxyy	10yyzzzz	
U+010000	U+10FFFF	11110uvv	10vvwwww	10xxxxyy	10yyzzzz

Byte-level BPE

- 每个char(字母,汉字,符号,😊...)都有一个code point
  - ＞130000
- 根据utf-8编码规则，将code point转为二进制格式(1B~4B)

Code point ↔ UTF-8 conversion

First code point	Last code point	Byte 1	Byte 2	Byte 3	Byte 4
U+0000	U+007F	0yyyzzzz			
U+0080	U+07FF	110xxxyy	10yyzzzz		
U+0800	U+FFFF	1110wwww	10xxxxyy	10yyzzzz	
U+010000	U+10FFFF	11110uvv	10vvwwww	10xxxxyy	10yyzzzz

```
>>> "😊".encode("utf-8")
b'\xf0\x9f\x98\x8a'

>>> "我".encode("utf-8")
b'\xe6\x88\x91'

>>> "h".encode("utf-8")
b'h'

>>> ord("😊")
128522
```

```
>>> "😊，哈哈hello？！😭☕️".encode("utf-8")
b'\xf0\x9f\x98\x8a\xef\xbc\x8c\xe5\x93\x88\xe5\x93\x88hello\xef\xbc\x9f\xef\xbc\x81\xf0\x9f\x98\xad\xe2\x98\x95\xef\xb8\x8f'
```

240	159	152	138	239	188

标点符号分割　　BPE算法

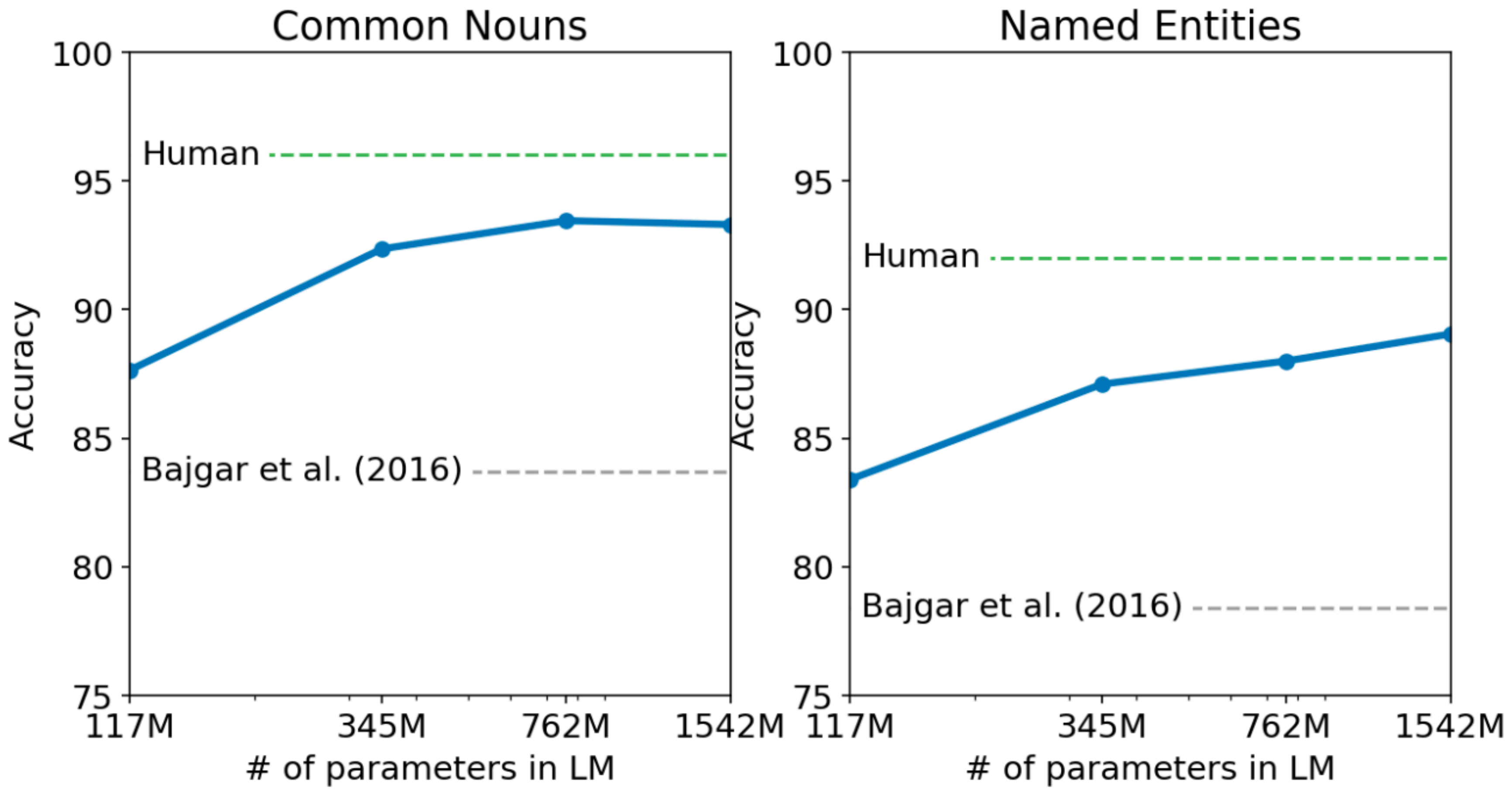
GPT-2: bigger GPT-1 + totally zero-shot

Language Models are Unsupervised Multitask Learners

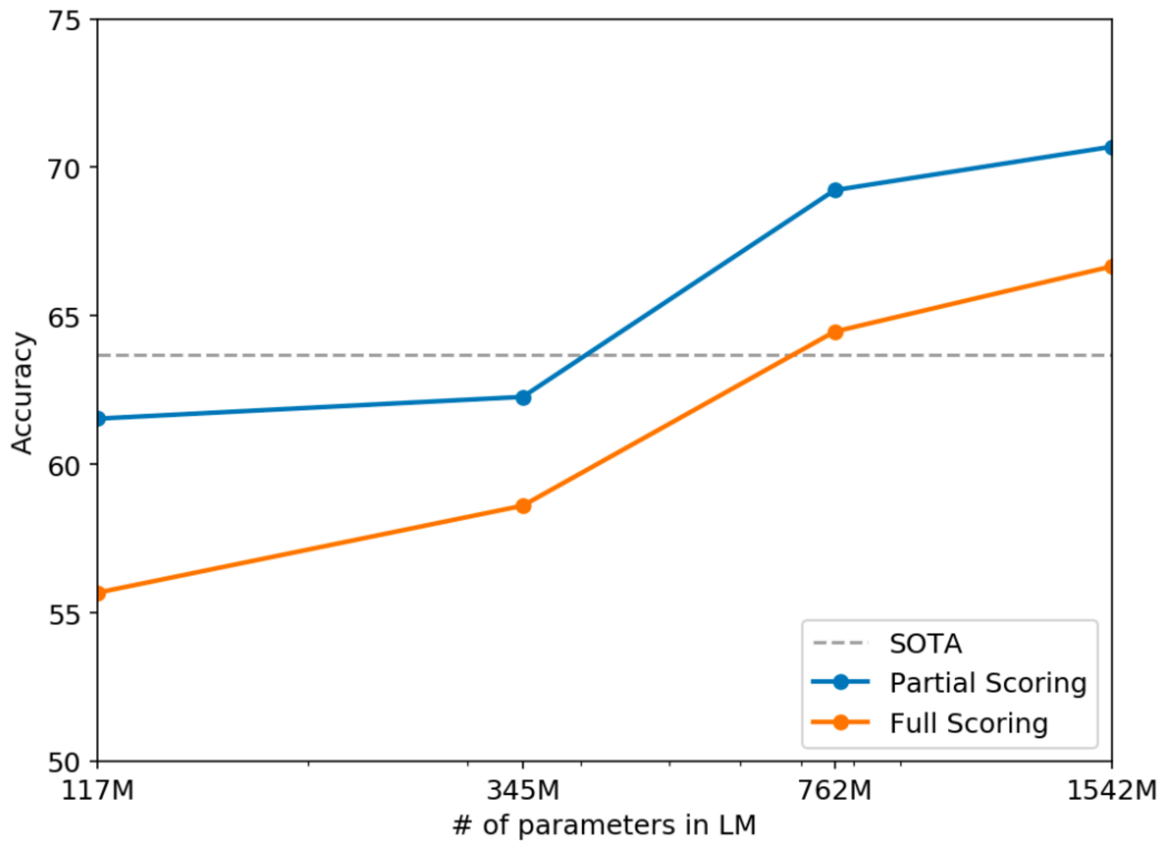
	LAMBADA (PPL)	LAMBADA (ACC)	CBT-CN (ACC)	CBT-NE (ACC)	WikiText2 (PPL)	PTB (PPL)	enwik8 (BPB)	text8 (BPC)	WikiText103 (PPL)	1BW (PPL)
SOTA	99.8	59.23	85.7	82.3	39.14	46.54	0.99	1.08	18.3	**21.8**
117M	**35.13**	45.99	**87.65**	**83.4**	**29.41**	65.85	1.16	1.17	37.50	75.20
345M	**15.60**	55.48	**92.35**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

	PTB	WikiText-2	enwik8	text8	Wikitext-103	1BW
Dataset train	**2.67%**	0.66%	**7.50%**	2.34%	**9.09%**	**13.19%**
WebText train	0.88%	**1.63%**	6.31%	**3.94%**	2.42%	3.75%

Table 6. Percentage of test set 8 grams overlapping with training sets.



CBT数据集



Winograd Schema Challenge

	R-1	R-2	R-L	R-AVG
Bottom-Up Sum	**41.22**	**18.68**	**38.34**	**32.75**
Lede-3	40.38	17.66	36.62	31.55
Seq2Seq + Attn	31.33	11.81	28.83	23.99
GPT-2 `TL;DR:`	29.34	8.27	26.58	21.40
Random-3	28.78	8.63	25.52	20.98
GPT-2 no hint	21.58	4.03	19.47	15.03

Table 4. Summarization performance as measured by ROUGE F1 metrics on the CNN and Daily Mail dataset. Bottom-Up Sum is the SOTA model from (Gehrmann et al., 2018)

GPT-2: bigger GPT-1 + totally zero-shot

- 里程碑
  - Scaling Laws
  - RLHF fine-tuning
  - Open —> ~ Close

编程实践

- gpt-1
- gpt-2
- llm.c
- modded-nanogpt