

# LLaMA：开放高效的基础语言模型

Hugo Touvron\*、Thibaut Lavril\*、Gautier Izacard\*、Xavier Martinet Marie-Anne Lachaux、Timothée Lacroix、Baptiste Rozière、Naman Goyal Eric Hambro、Faisal Azhar、Aurelien Rodriguez、Armand Joulin Edouard Grave\*、Guillaume Lampe\*。

元人工智能

## 摘要

我们介绍了 LLaMA，这是一个由 7B 到 65B

参数的基础语言模型集合。我们在数万亿个词库上训练我们的模型，并证明了完全使用公开可用的数据集训练最先进的模型是可能的，而无需求助于专有的和不可访问的数据集。我们向研究界发布了我们的所有模型<sup>1</sup>。

在推断性能方面，训练时间更长的小模型最终会更便宜。例如，尽管霍夫曼等人（2022 年）建议在 200B 标记上训练一个 10B 模型，但我们发现，即使在 1T

标记之后，7B 模型的性能也会继续提高。这项工作的重点是训练一系列语言模型，通过训练比通常使用的更多的词库，在各种推理预算下达到最佳性能。训练出的模型称为 LLaMA，参数范围从 7B 到 65B 不等，与现有的最佳 LLM

相比，性能极具竞争力。例如，LLaMA-13B 在大多数基准测试中都优于 GPT-3，尽管其体积小了 10 倍。我们相信，由于该模型可以在单个 GPU 上运行，它将有助于实现 LLM 访问和研究的民主化。在更高端的规模上，我们的 65B 参数模型与 Chinchilla 或 PaLM-540B 等最好的大型语言模型相比也具有竞争力。

与 Chinchilla、PaLM 或 GPT-3

不同的是，我们只使用公开可用的数据，这使得我们的工作与开源兼容，而大多数现有模型依赖的数据要么不公开可用，要么没有记录（例如“书籍 - 2TB”或“社交媒体对话”）。但也有一些例外，尤其是 OPT（Zhang 等人，2022 年）、GPT-NeoX（Black 等人，2022 年）、BLOOM（Scaoet 等人，2022 年）和 GLM（Zeng 等人，2022 年），但它们都无法与 PaLM-62B 或 Chinchilla 竞争。在本文的其余部分，我们将概述我们对转换器架构（Vaswani 等人，2017 年）所做的修改，以及我们的训练方法。然后，我们报告了我们模型的性能，并在一组标准基准上与其他 LLM 进行了比较。最后，我们利用负责任的人工智能社区的一些最新基准，揭示了我们的模型中编码的一些偏差和毒性。

## 1 引言

在海量文本语料库中训练的大型语言模型（LLMs）已经证明，它们有能力根据文本指令或少量示例完成新任务（Brown 等人，2020

年）。在将模型扩展到足够大的规模时，首次出现了这些寥寥无几的特性（Kaplan 等人，2020

年），由此产生了一系列侧重于进一步扩展这些模型的工作（Chowdhery 等人，2022 年；Rae 等人，2021

年）；然而，Hoffmann 等人（2022 年）的最新研究表明，对于给定的计算预算，最佳性能不是由最大的模型实现的，而是由在更多数据上训练的较小模型实现的。在这种情况下，给定一个目标性能水平，首选模型不是训练速度最快的，而是推理速度最快的：`{htouvron,thibautlav,gizacard,egrave,glample}@meta.com` <https://github.com/facebookresearch/llama>

2023227

## 2方法

我们的训练方法与之前的工作 (Brown 等人, 2020 年; Chowdhery 等人, 2022 年) 中描述的方法类似, 并受到了钦奇拉缩放定律 (Hoffmann 等人, 2022 年) 的启发。

### 2.1 预训练数据

我们的训练数据集由表 1

所列的多个数据源混合而成, 涵盖了各种不同的任务。在大多数情况下, 我们重复使用已用于训练其他 LLM 的数据源, 但仅限于使用可公开获得的数据, 并且与开源兼容。因此, 我们使用了以下混合数据以及它们在训练集中所占的比例: 我们使用 CCNet 管道 (Wenzek 等人, 2020 年) 预处理了从 2017 年到 2020 年的五份 CommonCrawl

转储数据。该流程在行级别对数据进行重复处理, 使用 fastText 线性分类器进行语言识别, 删除非英文页面, 并使用 n-gram 语言模型过滤低质量内容。此外, 我们还训练了一个线性模型, 将维基百科中用作参考的页面与随机抽样的页面进行分类, 并丢弃未被分类为参考的页面。

在探索性实验中, 我们发现使用不同的预处理 Com-monCrawl

数据集可能比我们的公开数据集更差, 因此我们将公开的数据集也包含重复数据删除和语言识别步骤: 与 CCNet 的主要区别在于质量过滤, 它主要依赖于标点符号的存在或网页中的单词和句子数量。我们只保留了 Apache 和 Wikipedia 项目。我们还删除了 20 个国家和地区的维基百科转储, 覆盖了 20 个国家和地区。

与 CCNet 的预处理也包含重复数据删除和语言识别步骤:

的主要区别在于质量过滤, 它主要依赖于标点符号的存在或网页中的单词和句子数量。

### 采样道具年代 磁盘大小

	CommonCrawl	67.0%	1.10	3.3 TB
C4	15.0%	1.06	783 GB	
Github	4.5%	0.64	328 GB	
Wikipedia	4.5%	2.45	83 GB	
Books	4.5%	2.23	85 GB	
ArXiv	2.5%	1.06	92 GB	
StackExchange	2.0%	1.03	78 GB	

表 1 : 预训练数据。表

1 : 用于预训练的数据混合物, 我们列出了每个子集的采样比例、在 1.4T

词组上训练时在子集上执行的历时数以及磁盘大小。在 1T 标记上的预训练运行具有相同的采样比例。

使用拉丁文或西里尔文的语言: bg、ca、cs、da、de、en、es、fr、hr、hu、it、nl、pl、pt、ro、ru、sl、sr、sv、uk。我们对数据进行处理, 删除超链接、注释和其他格式模板。

我们的训练数据集包括两个图书语料库: 古腾堡项目 (Guten-berg Project), 其中包含公共领域的图书; 以及 TheP-ile 的 Books3 部分 (Gao 等人, 2020 年), 这是一个用于训练大型语言模型的公开数据集。我们在图书层面进行了重复, 删除了内容重合度超过 90% 的图书。

我们对 arXiv Latex

文件进行了处理, 以便为数据集添加科学数据。我们还删除了.tex

文件中的注释, 并对用户编写的定义和宏进行了内联扩展, 以提高不同论文之间的一致性。

我们保留了 28

个最大网站的数据, 重新删除了文本中的 HTML 标记, 并按得分 (从高到低) 对答案进行了排序。

我们使用字节对编码 (BPE) 算法 (Sennrich 等人, 2015 年) 对数据进行标记化, 使用的是 Sentence-Piece 的实现 (Kudo 和 Richardson, 2018 年)。值得注意的是, 我们将所有数字拆分为单个数字, 并回退到字节来分解未知的 UTF-8 字符。