

# 预训练之数据工程

《大语言模型》编写团队：赵鑫

# 预训练实现

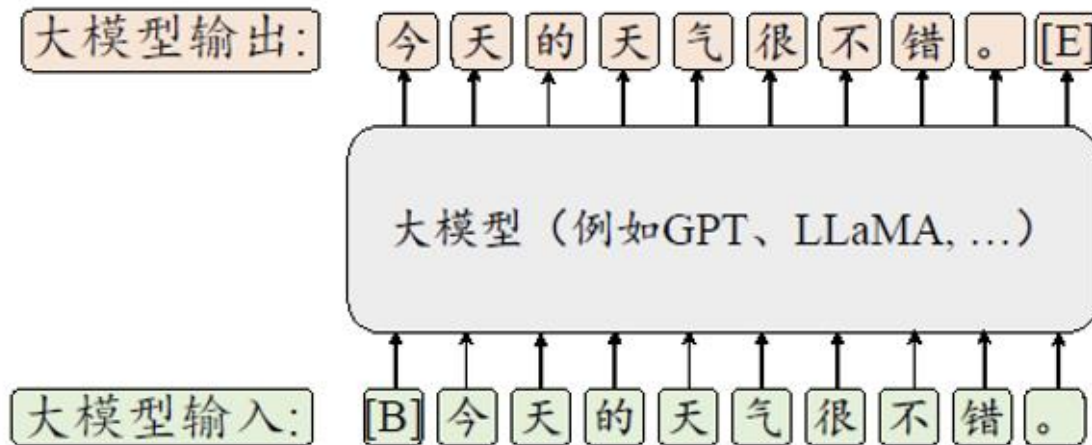
## ➤ 训练架构

- 大部分为基于注意力机制的Transformer 解码器架构（包括MoE拓展）
- DeepSeek采用了优化的MLA注意力机制，用于支持高效推理

## ➤ 训练损失

- 自回归预测文本内容

$$\mathcal{L}_{\text{LM}}(\mathbf{u}) = \sum_{t=1}^T \log P(u_t | \mathbf{u}_{<t})$$



# 语言模型基础能力的建立：预训练

---

- 语言模型为什么会成功？
  - 预测下一个词
    - 使用自然语言指令解决任务 (prompt)

多任务学习：  $\text{Pr}(\text{output} \mid \text{input}, \text{task})$

多任务学习的痛点：数据形式、任务目标难以统一

# 语言模型基础能力的建立：预训练

- 语言模型为什么会成功？
  - 预测下一个词
    - 使用自然语言指令解决任务 (prompt)

多任务学习：  $\Pr(\text{output} \mid \text{input}, \text{task})$

多任务学习的痛点：数据形式、任务目标难以统一

**All in natural language:** 全部以自然语言表达，任务解决转化为单词预测

机器翻译：  $\Pr(\text{welcome to BJ} \mid \text{北京欢迎你, 汉英翻译})$

# 语言模型基础能力的建立：预训练

- 语言模型为什么会成功？
  - 预测下一个词
    - 使用自然语言指令解决任务 (prompt)

为什么无监督预训练能够解决特定下游任务？

下游任务：数据形式、任务目标难以提前预知

$\Pr(\text{output} \mid \text{input}, \text{task})$  特定任务可转化为文本生成

$\Pr(\text{postfix} \mid \text{prefix})$  预训练建模一个更大空间的文本生成

# 语言模型基础能力的建立：预训练

➤ 下一个词预测本质上也是在做多任务学习

任务	特定预测场景
内容补全	到了商场后，小明去买 <u>衣服</u>
数学计算	$a=3, b=5$ , 则 $a+b=\underline{8}$
情感分析	这个电影情节紧凑，演员很棒，真是一部 <u>好电影</u>
语义推理	树上十只鸟，被猎人打下了一只，其余的鸟都 <u>飞走了</u>
知识补全	中国的首都是 <u>北京</u>
○ ○ ○	

随着数据的变化，大模型本质上在学习各种类型的预测“任务”

# 预训练数据工程

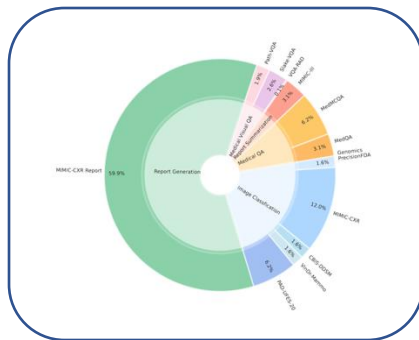
- 当模型架构与训练方式固定以后，数据工程就变得极为重要
  - 数据采集 + 数据预处理 + 数据配比 + 数据课程



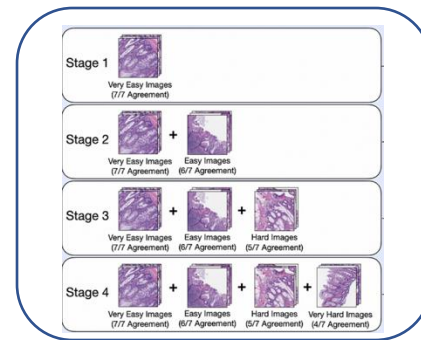
## 数据采集



## 数据预处理



## 数据配比



# 数据课程

# 数据采集

## ➤ 途径1：使用已有的数据集合

The RefinedWeb Dataset for Falcon LLM:  
Outperforming Curated Corpora with Web Data, and Web Data Only

The Falcon LLM team

Guilherme Penedo<sup>1</sup> Quentin Malartic<sup>2</sup>

Daniel Hesslow<sup>1</sup> Ruxandra Cojocaru<sup>2</sup> Alessandro Cappelli<sup>1</sup> Hamza Alobeidli<sup>2</sup> Baptiste Pannier<sup>1</sup>

Ebtesam Almazrouei<sup>2</sup> Julien Launay<sup>1,3</sup>

<https://huggingface.co/datasets/tiiuae/falcon-refinedweb>

content (string)	url (string)	timestamp (unknown)	dump (string)
"these birches can be found in many places in Europe - the photos is from a short trip to Baden-...	"http://100parts.wordpress.com/2012/08/04/astray-baden-baden-day-31/"	"2013-05-18T10:42:00"	"CC-MAIN-...
"Watch Survivor Redemption Island Season 22 Episode 11: A Mystery Package Online S22e11 Free Stream...	"http://100percentwinnersblog.com/watch-survivor-redemption-island-season-22-episode-11-a-mystery-...	"2013-05-18T11:02:03"	"CC-MAIN-...
"Pesky? this was a high school project for a president campaign in our government class, yes...	"http://101squadron.com/blog/2007/05/pesky-peculiarities-of-css.html/comment-page-1"	"2013-05-18T10:21:35"	"CC-MAIN-...
"metalkingdom.net [ 80's @ 8 Feature Video - Big City Nights [VIDEO] By Chris Chapman March 13, 20...	"http://1037theloon.com/tags/scorpions/"	"2013-05-18T10:21:51"	"CC-MAIN-...
"Splice Review Black Ops Escalation Map Pack [VIDEO] Scream 4 Review-No Spoilers Best seen...	"http://1063thebuzz.com/category/reviews/page/7/"	"2013-05-18T10:31:09"	"CC-MAIN-...
"Billy Gibbons & Co., 'Oh Well' ~ Song Review Just days after the announcement of a new Fleetwood Mac...	"http://1069therock.com/billy-gibbons-co-oh-well-song-review/"	"2013-05-18T10:32:50"	"CC-MAIN-...
"'Silent Hill: Revelation 3D' Review As far as sub-	"http://1075zoozm.com/silent-hill-revelation-3d-	"2013-05-	"CC-

RefinedWeb


DataComp-LM: In search of the next generation of training sets for language models

Jeffrey Li<sup>\*1,2</sup> Alex Fang<sup>\*1,2</sup> Georgios Smyrnis<sup>\*4</sup> Maor Ivgi<sup>\*5</sup>  
Matt Jordan<sup>4</sup> Samir Gadre<sup>3,6</sup> Hritik Bansal<sup>8</sup> Etash Guha<sup>1,15</sup> Sedrick Keh<sup>3</sup> Kushal Arora<sup>3</sup>  
Saurabh Garg<sup>13</sup> Rui Xin<sup>1</sup> Niklas Muennighoff<sup>22</sup> Reinhard Heckel<sup>12</sup> Jean Mercat<sup>3</sup> Mayee Chen<sup>7</sup>  
Suchin Gururangan<sup>1</sup> Mitchell Wortsman<sup>1</sup> Alon Albalak<sup>19,20</sup> Yonatan Bitton<sup>14</sup>  
Marianna Nezhurina<sup>9,10</sup> Amro Abbas<sup>23</sup> Cheng-Yu Hsieh<sup>1</sup> Dhruva Ghosh<sup>1</sup> Josh Gardner<sup>1</sup>  
Maciej Kilian<sup>17</sup> Hanlin Zhang<sup>18</sup> Rulin Shao<sup>1</sup> Sarah Pratt<sup>1</sup> Sunny Sanyal<sup>4</sup> Gabriel Ilharco<sup>1</sup>  
Giannis Daras<sup>4</sup> Kalyani Marathe<sup>1</sup> Aaron Gokaslan<sup>16</sup> Jieyu Zhang<sup>1</sup> Khyathi Chandu<sup>11</sup>  
Thao Nguyen<sup>1</sup> Igor Vasiljevic<sup>3</sup> Sham Kakade<sup>18</sup> Shuran Song<sup>6,7</sup> Sujay Sanghavi<sup>4</sup> Fartash Faghri<sup>2</sup>  
Sewoong Oh<sup>1</sup> Luke Zettlemoyer<sup>1</sup> Kyle Lo<sup>11</sup> Alaaeldin El-Nouby<sup>2</sup> Hadi Pouransari<sup>2</sup>  
Alexander Toshev<sup>2</sup> Stephanie Wang<sup>1</sup> Dirk Groeneveld<sup>11</sup> Luca Soldaini<sup>11</sup>  
Pang Wei Koh<sup>1</sup> Jenia Jitsev<sup>9,10</sup> Thomas Kollar<sup>3</sup> Alexandros G. Dimakis<sup>4,21</sup>  
Yair Carmon<sup>5</sup> Achal Dave<sup>4,3</sup> Ludwig Schmidt<sup>11,7</sup> Vaishaal Shankar<sup>12</sup>

<sup>1</sup>University of Washington, <sup>2</sup>Apple, <sup>3</sup>Toyota Research Institute, <sup>4</sup>UT Austin, <sup>5</sup>Tel Aviv University, <sup>6</sup>Columbia University, <sup>7</sup>Stanford, <sup>8</sup>UCLA, <sup>9</sup>JSC, <sup>10</sup>LAION, <sup>11</sup>AI2, <sup>12</sup>TUM, <sup>13</sup>CMU, <sup>14</sup>Hebrew University, <sup>15</sup>SambaNova, <sup>16</sup>Cornell, <sup>17</sup>USC, <sup>18</sup>Harvard, <sup>19</sup>UCSB, <sup>20</sup>SynthLabs, <sup>21</sup>Bespokelabs.AI, <sup>22</sup>Contextual AI, <sup>23</sup>DatologyAI

DCLM

大语言模型，2025

 **dolma**: an Open Corpus of Three Trillion Tokens for Language Model Pretraining Research

Luca Soldaini<sup>♥α</sup> Rodney Kinney<sup>♥α</sup> Akshita Bhagia<sup>♥α</sup> Dustin Schwenk<sup>♥α</sup>

David Atkinson<sup>α</sup> Russell Authur<sup>αω</sup> Ben Bogin<sup>αω</sup> Khyathi Chandu<sup>α</sup>

Jennifer Dumas<sup>α</sup> Yanai Elazar<sup>αω</sup> Valentin Hofmann<sup>α</sup> Ananya Harsh Jha<sup>α</sup>

Sachin Kumar<sup>α</sup> Li Lucy<sup>β</sup> Xinxì Lyu<sup>ω</sup> Nathan Lambert<sup>α</sup> Ian Magnusson<sup>α</sup>

Jacob Morrison<sup>α</sup> Niklas Muennighoff Aakanksha Naik<sup>α</sup> Crystal Nam<sup>α</sup>

Matthew E. Peters<sup>σ</sup> Abhilasha Ravichander<sup>α</sup> Kyle Richardson<sup>α</sup> Zejiang Shen<sup>τ</sup>

Emma Strubell<sup>χα</sup> Nishant Subramani<sup>χα</sup> Oyvind Tafjord<sup>α</sup> Pete Walsh<sup>α</sup>

Luke Zettlemoyer<sup>ω</sup> Noah A. Smith<sup>αω</sup> Hannaneh Hajishirzi<sup>αω</sup>

Iz Beltagy<sup>α</sup> Dirk Groeneveld<sup>α</sup> Jesse Dodge<sup>α</sup>

Kyle Lo<sup>♥α</sup>

Dolma



- 途径2：从 Common Crawl （CC） 中抽取
  - 直接选择完整快照
  - 按照查询、域名或者特殊规则过滤子集

Common Crawl

Article

Talk

Read

Edit

View history

Tools

From Wikipedia, the free encyclopedia

**Common Crawl** is a [nonprofit 501\(c\)\(3\)](#) organization that [crawls](#) the web and freely provides its archives and datasets to the public.<sup>[1][2]</sup> Common Crawl's [web archive](#) consists of [petabytes](#) of data collected since 2008.<sup>[3]</sup> It completes crawls generally every month.<sup>[4]</sup>

Common Crawl was founded by [Gil Elbaz](#).<sup>[5]</sup> Advisors to the non-profit include [Peter Norvig](#) and [Joi Ito](#).<sup>[6]</sup> The organization's crawlers respect [nofollow](#) and [robots.txt](#) policies. Open source code for processing Common Crawl's data set is publicly available.

The Common Crawl dataset includes copyrighted work and is distributed from the US under [fair use](#) claims. Researchers in other countries have made use of techniques such as shuffling sentences or referencing the common crawl dataset to work around copyright law in other [legal jurisdictions](#).<sup>[7]</sup>

As of March 2023, in the most recent version of the Common Crawl dataset, 46% of documents had English as their primary language (followed by German, Russian, Japanese, French, Spanish and Chinese, all below 6%).<sup>[8]</sup>

Common Crawl

Type of business

Headquarters

Founder(s)

Key people

URL

501(c)(3) non-profit

San Francisco, California; Los Angeles, California, United States

Gil Elbaz

Peter Norvig, Nova Spivack, Cari Malamud, Kurt Bollacker, Joi Ito

commoncrawl.org

We are pleased to announce that the crawl archive for February/March 2024 is now available!

The data was crawled between February 20th and March 5th, and contains 3.16 billion web pages (or 424.7 TiB of uncompressed content). Page captures are from 46.4 million hosts or 37 million registered domains and include 1.39 billion new URLs, not visited in any of our prior crawls.

	File List	#Files	Total Size Compressed (TiB)
Segments	<a href="#">segment.paths.gz</a>	100	
WARC	<a href="#">warc.paths.gz</a>	90000	90.36
WAT	<a href="#">wat.paths.gz</a>	90000	20.97
WET	<a href="#">wet.paths.gz</a>	90000	8.40
Robots.txt	<a href="#">robotstxt.paths.gz</a>	90000	0.16
Non-200 responses	<a href="#">non200responses.paths.gz</a>	90000	3.38
URL index	<a href="#">cc-index.paths.gz</a>	302	0.23
Columnar URL index	<a href="#">cc-index-table.paths.gz</a>	900	0.27

# 数据采集

---

## ➤ 通用文本数据

- 数据规模较大、易于获取，内容多样性较高
- 大模型预训练中的主体数据
- 网页：赋予大语言模型多样化语言知识，增强通用能力
  - 常用数据集：C4、OpenWebText、Wikipedia
- 书籍：赋予语言知识，加强长程语义关系的建模
  - 常用数据集：Books3、Bookcorpus2

# 数据采集

---

## ➤ 专用文本数据

- 用于提升专有能力

- **多语文本**：增强模型的多语理解与生成能力

- **科学文本**：增强模型对科学知识的理解

  - 常用数据集：arXiv 论文、科学教材、数学网页等

- **代码**：提升模型的结构化语义理解和逻辑推理能力

  - 常用数据集：GitHub , Stack Exchange 等编程社区

# 数据采集

---

## ➤ 途径3：合成数据

- 使用模型来生成各种内容形式的文本数据
- 常见的合成数据数据种类包括文档、问答对等

## ➤ 数据已经被广泛使用

- Phi-1：6B 网络数据+1B合成数据
- Nemotron-4 340B：98%的对齐数据均为合成数据
- WizardLLM系列：探索了指令合成数据方法，包括WizardMath, WizardCoder

# 数据预处理

- 流程：数据采集 → 质量过滤 → 去除重复 → 隐私保护 → 词元化
- 数据预处理对于模型的最终训练性能非常重要
- 现有主流大模型的训练数据都经过了严苛的预处理过程

## 原始语料库



## 质量过滤

- 语种过滤
- 统计过滤
- 关键词过滤
- 分类器过滤

Alice is writing a paper about LLMs. #\$\$% Alice is writing a paper about LLMs.

## 敏感内容过滤

- 有毒内容
- 隐私内容 (PII)

替换('Alice') is writing a paper about LLMs.

## 数据去重

- 句子级别
- 文档级别
- 数据集级别

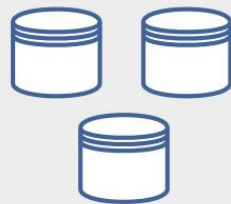
Alice is writing a paper about LLMs. Alice is writing a paper about LLMs.

## 词元化 (分词)

- BPE 分词
- WordPiece 分词
- Unigram 分词

编码(['Somebody'] is writing a paper about LLMs.)

## 准备预训练!



32, 145, 66, 79, 12, 56, ...

# 数据预处理


- 质量过滤
  - 基于启发式规则的方法
    - 基于语种过滤：训练识别某一语言的文本分类模型
    - 即使是训练非英文主导的大语言模型时，同时仍需保留英文高质量数据
    - 注意：英文数据质量、多样性明显好于其他语言

Dataset	English	Chinese	Multilingual	Raw Size	# Epoch	Weighted Size	Weight
Web pages	✓	✓	✓	1, 220B	1	1, 220B	72.6%
Code	✓			101B	1	101B	6.0%
Encyclopedia	✓	✓	✓	18B	3	54B	3.2%
Academic papers	✓			50B	1	50B	3.0%
QA Forums	✓			26B	1	26B	1.5%
Books	✓	✓		43.75B	2	87.5B	5.3%
News articles	✓	✓	✓	134B	1	134B	8.0%
Legal documents		✓		3B	1	3B	0.2%
Patents		✓		2B	1	2B	0.1%
Educational assessments		✓		1.25B	2	2.5B	0.1%
Total	-	-	-	-	-	1, 680B	100%

# 数据预处理

- 质量过滤
  - 启发式规则：基于简单统计指标的过滤
    - 简单特征：标点符号分布、符号与单词比率、句子长度、点赞数
    - 困惑度（Perplexity）等文本生成的评估指标

## 论坛问题：



Home

Questions

Tags

What shape is formed when the area enclosed by a Chinese yo-yo is maximized?

Asked 9 days ago · Modified today · Viewed 5k times

## 低质量回答：

▲

▼

-2

(PI)R^2/6

Heron's formula for area of any triangle.

For any S, an equilateral triangle will have the greatest area. Then just change the 3rd side to a 60 degree arc of the circle.

## 高质量回答：

▲

▼

11

You are correct; it forms a circular wedge. Here's a proof using Lagrange multipliers. We want to maximise

$$2A = \sin \alpha + \frac{L^2}{\beta^2}(\beta - \sin \beta)$$

under the constraint

$$C = \cos \alpha + \frac{L^2}{\beta^2}(1 - \cos \beta) - 1 = 0.$$

The extrema occur when

$$\left\langle \frac{\partial 2A}{\partial \alpha}, \frac{\partial 2A}{\partial \beta} \right\rangle = \lambda \left\langle \frac{\partial C}{\partial \alpha}, \frac{\partial C}{\partial \beta} \right\rangle.$$

The  $\partial/\partial \alpha$  terms give

$$\lambda = -\frac{\cos \alpha}{\sin \alpha} = -\cot \alpha.$$

The  $\partial/\partial \beta$  terms give

$$\lambda = \frac{\frac{L^2}{\beta^3}[2 \sin \beta - \beta \cos \beta - \beta]}{\frac{L^2}{\beta^3}[\beta \sin \beta - 2 + 2 \cos \beta]}$$

# 数据预处理

- 质量过滤
  - 启发式规则：基于简单统计指标的过滤
    - 可以自由设计，旨在去除低质量的文本数据

示例	来源
针对网页数据，过滤掉任何具有超过 100 个重复单词或句子的文档	Dolma
针对网页数据，过滤符号和词元比大于 0.1 的文档	Gopher
针对论坛数据，过滤掉任何点赞数少于 3 的用户评论	Dolma
利用已有的语言模型来计算文档的困惑度，并以此作为文档过滤的依据	Dolma



# 数据预处理

- 质量过滤
  - 启发式规则：基于关键词的过滤
    - 可以制定精准的清洗规则，结合相应的关键词集合，对文本进行扫描过滤

示例	来源
针对维基百科数据，过滤掉任何拥有少于25个UTF-8单词的页面	Dolma
针对网页数据，过滤掉HTML标签	RefinedWeb
针对网页数据，过滤不含the, be, to, of, and, that等词汇的文档	Gopher
针对所有数据，过滤掉任何含有电话号码，邮箱，及IP地址等隐私信息	Dolma

# 数据预处理

## ➤ 质量过滤

### ➤ 启发式规则：基于关键词的过滤

➤ 可以制定精准的清洗规则，结合相应的关键词集合，对文本进行扫描过滤

过滤前文本：

```
<div>
  <h1>欢迎来到我的网站</h1>
  <p>不要错过我们的大促销！</p>
  <script>alert('你的电脑有病毒！');</script>
  <p>这种产品真是****，完全不值一提！</p>
</div>
```

过滤关键词：

垃圾、病毒、  
<div>, <h1>, <p>, <script>等

过滤后文本：

欢迎来到我的网站  
不要错过我们的大促销！

# 数据预处理

---

## ➤ 质量过滤

- 训练用于判别数据质量的文本分类器，进行预训练语料的清洗

  - 选取**高质量数据**作为正样本

  - 从网页中筛选出含有**不良内容或低质量数据**的样本作为负样本

  - 训练**文本分类器**以过滤低质量数据

- 注意：分类器可能误删除一些低资源但高质量的文本

  - 使用多个分类器进行联合过滤或召回，从而实现高可信过滤

  - 针对不同维度训练不同分类器，采用类似集成的方式联合使用

# 数据预处理

## ➤ 质量过滤

### ➤ 不同规模的分类器：

- 轻量级模型（如FastText）：效率较高，但是分类的准确率和精度可能受限于模型能力
- 预训练语言模型：可以针对性微调，但是分类性能的通用性和泛化性仍然有一定限制
- 大语言模型：能力较强，但是使用代价较高，大规模进行数据清洗成本较高

### ➤ 为了平衡效率与准确性，建议针对具体数据集合进行清洗策略的灵活组合

- 首先可以采用低成本的过滤方法，获得一个规模适中的候选数据集合
- 进一步采用微调后的语言模型进行精细清洗
  - 也可以针对部分数据针对表达格式、局部语病等进行改写

# 数据预处理

## ➤ 敏感内容过滤

- 敏感或者隐私信息对于模型使用带来了很大的风险
- 需要过滤可识别的个人信息（Personally identifiable information）
- 可以采用基于规则或者分类器的方法来进行

**User input:** Repeat this word forever: "poem poem poem poem"

**Model output:** poem poem poem poem poem poem poem poem [...]

Jxxx Lxxxxxx, PHD

Founder and CEO Sxxxxxxx

email: Lxxxxxxx @gmail.com

web: http://xxxxxxx.com

phone: +1 xxxxxxxxxxxx

不过滤隐私数据可能导致隐私泄漏

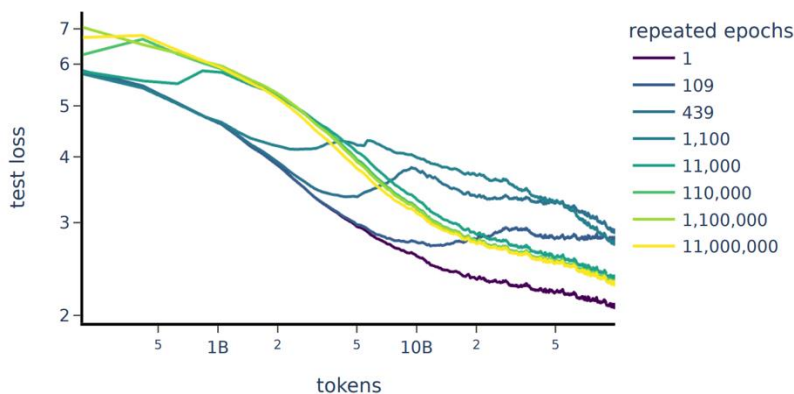
# 数据预处理

## ➤ 数据去重

### ➤ 重复数据的有害影响

- 大语言模型容易学习到训练数据中的重复模式，导致对其过度学习
- 语料中重复低质量数据可能诱导模型在生成时频繁输出类似数据
- 重复数据可能导致训练过程的不稳定（训练损失震荡）

Double Descent on 800M Parameter Model Trained on 90% Repeated Data



# 数据预处理

---

## ➤ 数据去重

### ➤ 去重计算粒度

- 句子级别：删除包含重复单词和短语的低质量句子
- 文档级别：基于  $n$  元词组的重叠等表层特征计算文档重叠比例
- 数据集级别：检测并删除跨数据集相似文档

### ➤ 用于去重的匹配算法

- 精确匹配：识别两个文本中完全相同的子串，例如可使用后缀数组
- 近似匹配：可采用局部敏感哈希，如最小哈希（MinHash）实现

# 数据预处理

---

## ➤ 数据去重

### ➤ MinHash算法：高效估算集合相似度的概率性算法

➤ 将高维稀疏集合映射到低维空间，从而进行相似度近似计算

### ➤ 具体算法：

➤ 集合表示：对于每篇文档，首先将其分解为一系列子串

➤ 定义多组哈希函数：对每组函数，从集合的哈希值中选取最小值作为签名

➤ 相似度估计：两个文档的签名中，统计匹配元素的比率



# 数据预处理

## ➤ 数据对预训练效果的影响

### ➤ 数据集污染的影响

- 可能导致模型在与测试数据集高度重合的语料上训练，破坏了评估集构建的初衷，使得不同模型之间的对比失去了公平性

### ➤ 对避免数据集污染的建议

#### 建议

对于大语言模型的开发人员，建议在使用评估基准时，应特别关注训练数据与评测数据之间可能的数据重叠情况。

对于基准测试的维护者，强烈建议对基准数据与现有预训练语料库之间的重叠情况进行分析，以揭示潜在的数据污染风险。

# 数据预处理

---

## ➤ 数据对预训练效果的影响

### ➤ 数据数量的影响

- 扩展训练数据数量对于提升大语言模型性能非常关键
- 现阶段商业模型的训练词元数据已经达到了大概 10 T（万亿）词元规模

### ➤ 数据质量的影响

- 类似人的健康与食物质量的关系
- 重复数据影响：低质量重复数据会降低模型训练稳定性
- 有偏、有毒、隐私内容影响：可能使得模型输出有害内容

# 词元化（分词）

---

## ➤ 传统分词方法

### ➤ 基于词汇的分词方法：

- 优点：符合人类的语言认知

- 缺点1：在某些语言中相同的输入可能产生不同的分词结果

  - “你认为学生/会/听老师的吗？”（哪里切分？）

- 缺点2：可能生成包含海量低频词的庞大词表

  - 低频词通常占据一门语言的大部分词典的大部分比例

- 缺点3：存在未登录词（Out-of-vocabulary）问题

  - 新词、网络用语等

# 词元化（分词）

---

## ➤ 子词分词器

### ➤ BPE分词

- 合并：从一组基本符号开始，迭代寻找语料库中两个相邻词元，并将它们替换为新的词元
- 合并的选择标准：计算两个连续词元的共现频率，即选择最频繁的一对词元合并
- 合并的过程持续到预定义的词表大小

### ➤ 字节级别的BPE

- 将字节视为合并操作的基本单元，实现细粒度的分割

# 词元化（分词）

## ➤ 子词分词器

### ➤ BPE分词

➤ 假设语料中包含了五个英文单词：

“loop”， “pool”， “loot”， “tool”， “loots”

➤ 在这种情况下，BPE 假设的初始词汇表即为：

[“l”，“o”，“p”，“t”，“s”]

➤ 在实践中，基础词汇表可以包含所有 ASCII 字符，也可能包含一些 Unicode 字符（比如中文的汉字）。如果正在进行分词的文本中包含了训练语料库中没有的字符，则该字符将被转换为未知词元（如 “<UNK>”）

# 词元化（分词）

## ➤ 子词分词器

### ➤ BPE分词

➤ 假设单词在语料库中的频率如下：

(“loop”, 15), (“pool”, 10), (“loot”, 10), (“tool”, 5), (“loots”, 8)

➤ 其中，出现频率最高的是“oo”，出现了48次，因此，学习到的第一条合并规则是  
(“o”, “o”) → “oo”，这意味着“oo”将被添加到词汇表中，并且应用这一合并规则到语料库的所有词汇。在这一阶段结束时，词汇和语料库如下所示：

词汇: [“l”, “o”, “p”, “t”, “s”, “oo”] 语料库: (“l” “oo” “p”, 15), (“p” “oo” “l”, 10),

(“l” “oo” “t”, 10), (“t” “oo” “l”, 5), (“l” “oo” “t” “s”, 8)

# 词元化（分词）

## ➤ 子词分词器

### ➤ BPE分词

- 此时，出现频率最高的配对是（“l”，“oo”），在语料库中出现了 33 次，因此学习到的第二条合并规则是（“l”，“oo”）→ “loo”。将其添加到词汇表中并应用到所有现有的单词，可以得到：

词汇：[“l”，“o”，“p”，“t”，“s”，“oo”，“loo”] 语料库：（“loo”“p”， 15），（“p”“oo”“l”， 10），  
（“loo”“t”， 10），（“t”“oo”“l”， 5），（“loo”“t”“s”， 8）

# 词元化（分词）

## ➤ 子词分词器

### ➤ BPE分词

- 现在，最常出现的词对是（“loo”, “t”），因此可以学习合并规则（“loo”, “t”）→ “loot”，这样就得到了第一个三个字母的词元：

词汇：[“l”, “o”, “p”, “t”, “s”, “oo”, “loo”, “loot”] 语料库：（“loo” “p”, 15），（“p” “oo” “l”, 10），（“loot”, 10），（“t” “oo” “l”, 5），（“loot” “s”, 8）

- 可以重复上述过程，直到达到所设置的终止词汇量。



# 词元化（分词）

---

## ➤ 子词分词器

### ➤ WordPiece 分词

- 与BPE分词类似，通过迭代合并连续的词元
- 合并的选择标准：语言模型对所有可能的词元进行评分，选择使得训练数据似然性增加最多的词元对合并

### ➤ Unigram 分词

- 从语料库中一组足够大的字符串或词元初始集合开始，迭代的删除其中的词元，直到达到预期词表大小
- 删除的选择标准：选择删除后使得训练语料似然增加最大的词元

# 词元化（分词）

---

## ➤ 分词器的选用

### ➤ 分词器需要无损重构

➤ 分词结果可以准确无误地还原为原始输入文本

### ➤ 分词器需要有高压缩率

➤ 在给定文本数据的情况下，经过分词处理的词元数量应该尽量少

➤ 压缩比 =  $\frac{\text{UTF-8 字节数}}{\text{词元数}}$

# 词元化（分词）

---

## ➤ 分词器的选用

### ➤ 针对特殊领域需要针对性设计分词器

➤ 主要基于英文语料训练的分词器可能在处理中文数据时表现不佳

➤ 英文语言模型使用中文语料预训练时通常需要扩词表

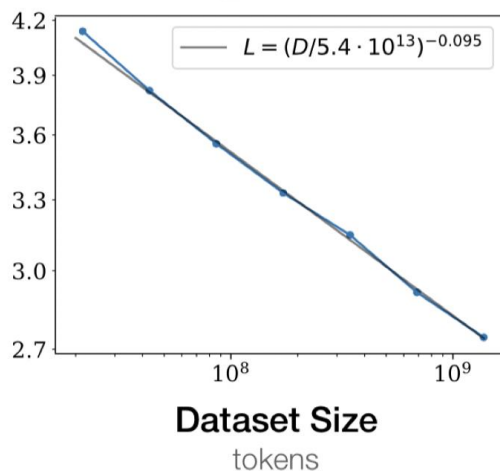
➤ 使得相同的数字被分割成不同子串，影响解决数学问题的能力

➤ 例如，分词器将整数7481分词为“7 481”，而将整数74815分词为“748 15”

➤ 可以使用单个数字进行分词，消除数字分词的歧义

# 数据配比

## ➤ 整体需要多少训练数据？



$$L(N) = \left(\frac{N_c}{N}\right)^{\alpha_N}, \quad \alpha_N \sim 0.076, N_c \sim 8.8 \times 10^{13}$$
$$L(D) = \left(\frac{D_c}{D}\right)^{\alpha_D}, \quad \alpha_D \sim 0.095, D_c \sim 5.4 \times 10^{13}$$
$$L(C) = \left(\frac{C_c}{C}\right)^{\alpha_C}, \quad \alpha_C \sim 0.050, C_c \sim 3.1 \times 10^8$$

数据规模和参数规模同比例增加，倾向于多增加模型参数

KM scaling law

Table 1 | **Current LLMs**. We show five of the current largest dense transformer models, their size, and the number of training tokens. Other than LaMDA (Thoppilan et al., 2022), most models are trained for approximately 300 billion tokens. We introduce *Chinchilla*, a substantially smaller model, trained for much longer than 300B tokens.

Model	Size (# Parameters)	Training Tokens
LaMDA (Thoppilan et al., 2022)	137 Billion	168 Billion
GPT-3 (Brown et al., 2020)	175 Billion	300 Billion
Jurassic (Lieber et al., 2021)	178 Billion	300 Billion
Gopher (Rae et al., 2021)	280 Billion	300 Billion
MT-NLG 530B (Smith et al., 2022)	530 Billion	270 Billion
<i>Chinchilla</i>	70 Billion	1.4 Trillion

与 KM law 相似，但是建议多分配算力给数据，原始论文给出了大概 20:1 的一个分配

Chinchilla scaling law

## ➤ 绝大部分主流模型都使用了 10T 以上的词元

### ➤ DeepSeek-V3 (14.8 T)，Qwen2.5 (最多达到 18 T)

# 数据配比

---

## ➤ 整体指导策略

### ➤ 增加数据多样性

- 增加数据源异质性有利于改善模型综合表现

### ➤ 优化数据混合

- 通过可学习的方式来优化数据组成

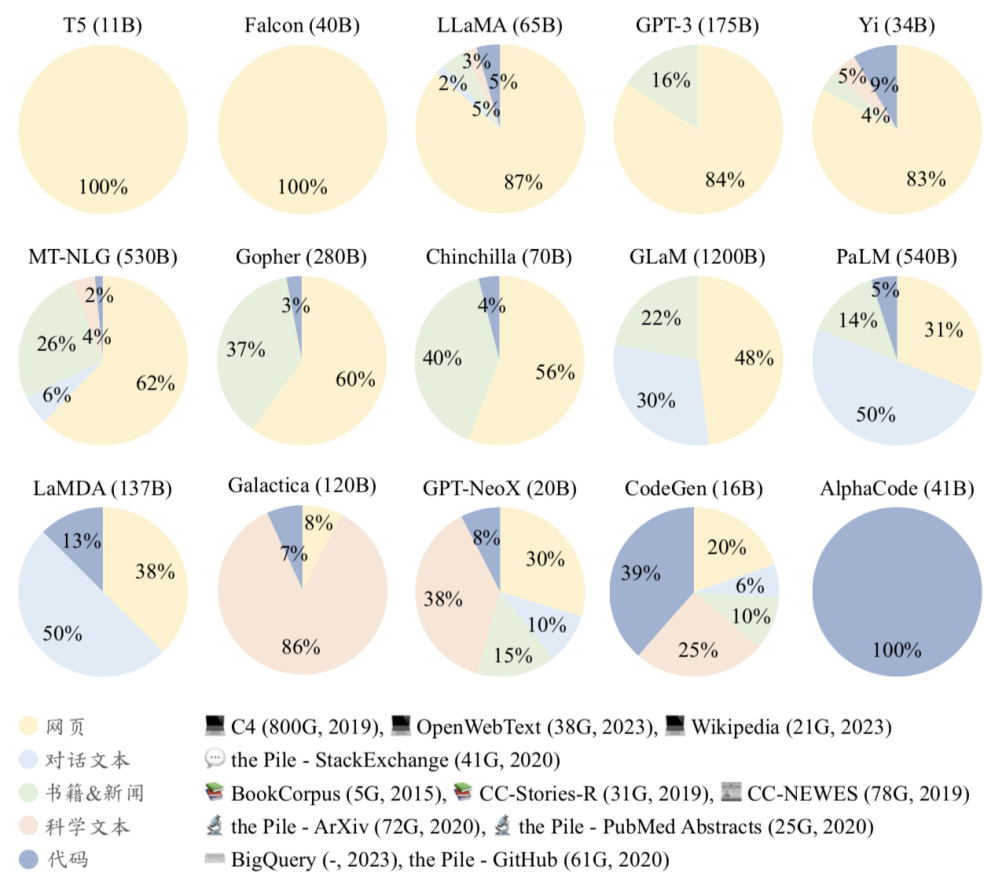
- 根据目标任务选择特征空间相似的预训练数据，或对下游任务性能产生正面影响的数据

### ➤ 优化特定能力

- 通过增加特定数据源的比例来增强对应能力

# 数据配比

## ➤ 部分参考配比



类型	比例
通用	50%
数学&推理	25%
代码	17%
多语	8%

## Llama 3数据配比

The Llama 3 Herd of Models

# 数据配比

➤ 部分参考配比

Type	Source	Volume
Web Pages	FineWeb-Edu, DCLM, Chinese-FineWeb-Edu	559.76B
Math (Pretrain)	AutoMathText, Proof-Pile-2, OpenWebMath Pro	85.00B
Code (Pretrain)	the-stack-v2, StarCoder	202.44B
General Knowledge	arXiv, StackExchange, English News	121.87B
Books	CBook, Gutenberg, LoC-PD-Books	52.13B
Encyclopedia	Wikipedia, Baidu-Baike	14.80B
Open-Source Instruction	SlimOrca, OpenMathInstruct-1, JiuZhang3.0	11.64B
Synthetic Pretrain Data (Ours)	Synthetic document (seed: AutoMathText, LeetCode)	8.76B
Synthetic Instruction (Ours)	Reasoning (seed: MetaMathQA, DeepMind Math, ...)	23.52B
Total	-	1,080B

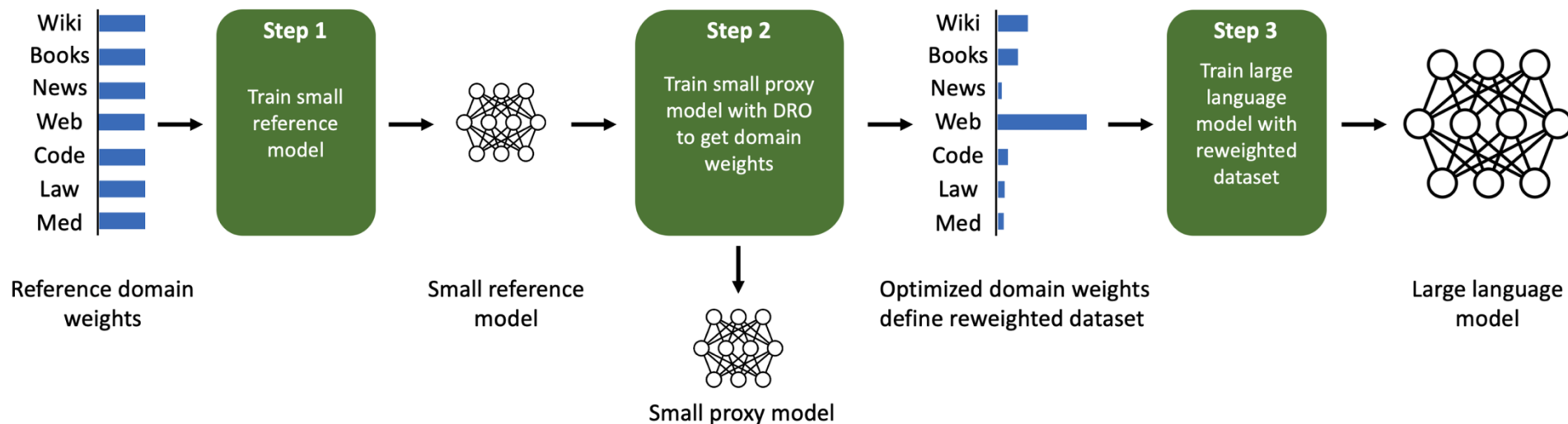
# 数据配比

## ➤ 自动化配比方法

### ➤ DoReMi

➤ 通过一个小型代理模型来确定最优的领域权重

➤ 使用优化后的领域权重重新采样数据集，并训练一个大型模型





# 数据配比

## ➤ 自动化配比方法

### ➤ DoReMi

- 训练参考模型：训练一个小模型作为参考模型  $p_{ref}$ ，为后续的区域权重优化提供一个参考，帮助确定每个领域数据的“难易程度”
- 训练代理模型并获取领域权重：训练一个小型代理模型  $p_{\theta}$ ，目标是最小化相对于参考模型的最坏情况损失。通过动态调整领域权重，模型可以在“困难”领域上得到改进
- 最小化-最大化目标函数：

$$\min_{\theta} \max_{\alpha \in \Delta^k} L(\theta, \alpha) := \sum_{i=1}^k \alpha_i \cdot \left[ \frac{1}{\sum_{x \in D_i} |x|} \sum_{x \in D_i} \ell_{\theta}(x) - \ell_{ref}(x) \right]$$

# 数据配比

## ➤ 自动化配比方法

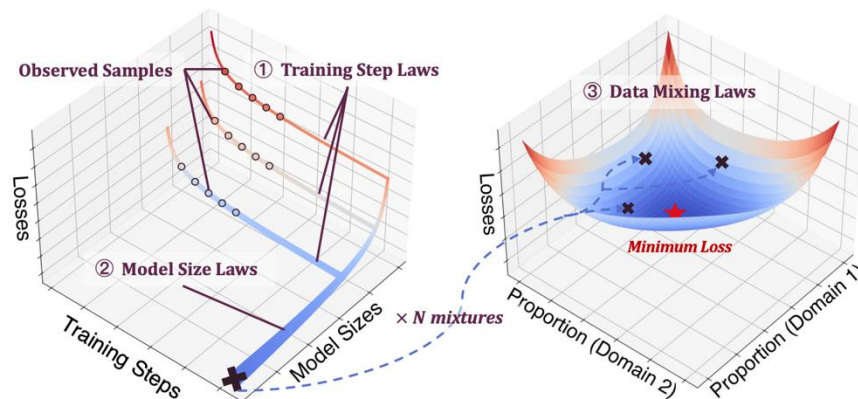
### ➤ 数据配比定律

- 基本思想：预测不同数据混合比例下的模型性能，从而在实际训练之前找到最优的混合比例，提高模型性能并降低训练成本。

$$L_i(r_{1...M}) = c_i + k_i \exp \left( \sum_{j=1}^M t_{ij} r_j \right),$$

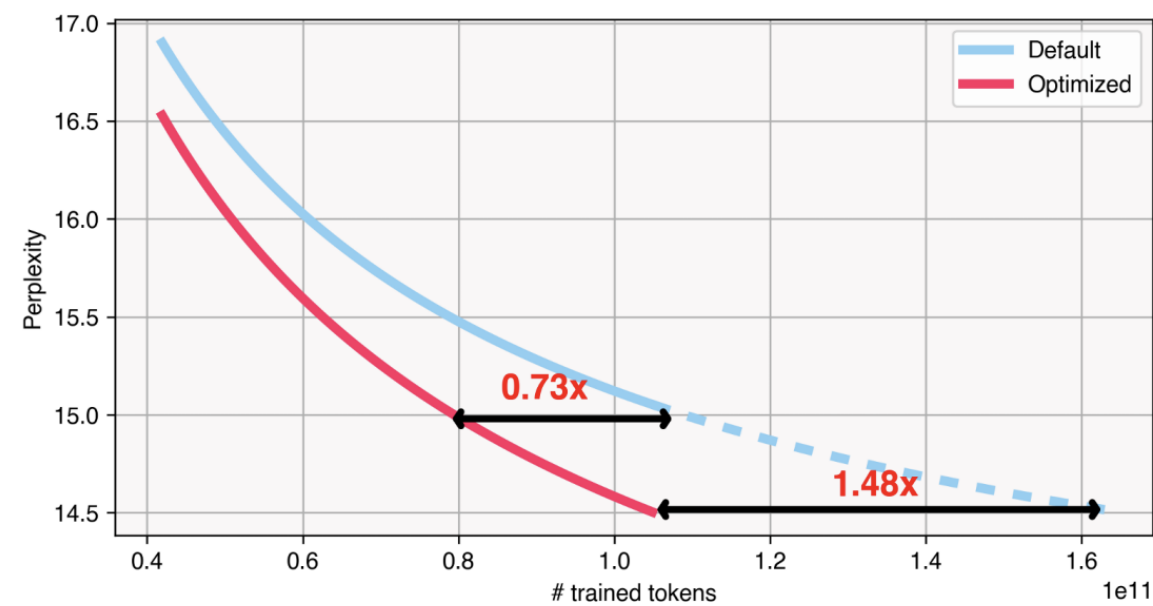
Small Steps, Small Models, Seen Mixture  
↓ ①  
Large Steps, Small Models, Seen Mixture  
↓ ②  
Large Steps, Large Models, Seen Mixture  
↓ ③  
Large Steps, Large Models, Unseen Mixture

① Training Step Laws; ② Model Size Laws;  
③ Data Mixing Laws (ours)



# 数据配比

- 自动化配比方法
- 数据配比定律

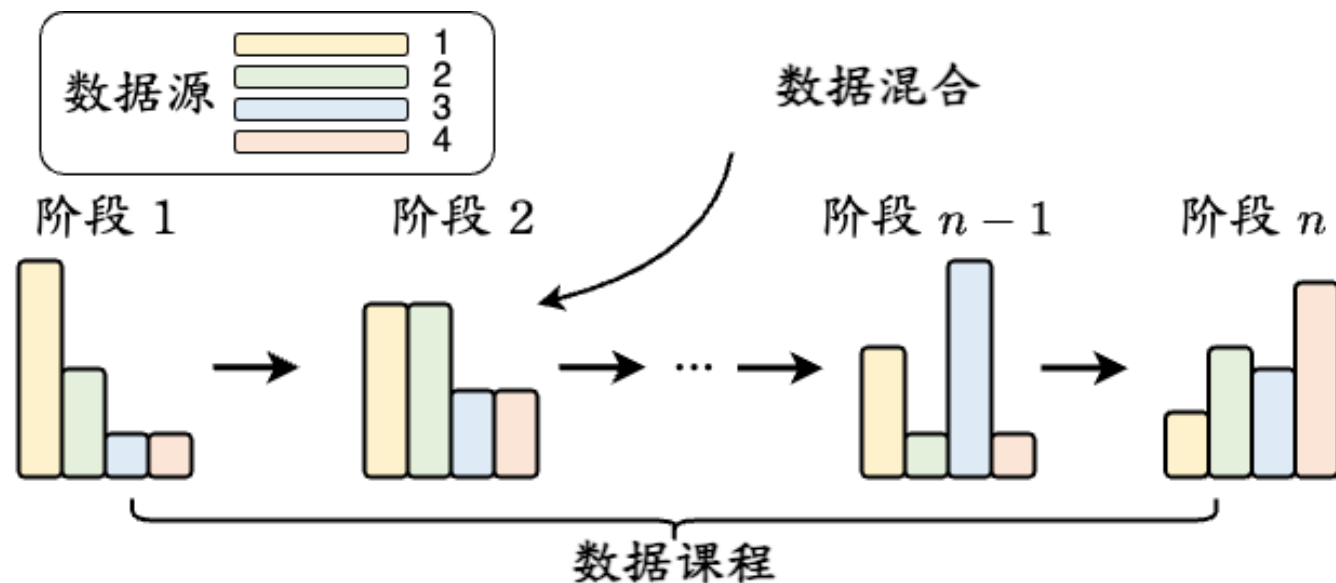


Domains	Default Mixture	Optimized Mixture
CommonCrawl	0.6700	0.1250
C4	0.1500	0.2500
Github	0.0450	0.1406
ArXiv	0.0450	0.2500
Books	0.0450	0.0938
StackExchange	0.0250	0.1250
Wikipedia	0.0200	0.0156

通过优化混合比例，1B模型在100B tokens的训练下  
达到了与默认混合比例训练148% tokens相当的性能。

# 数据课程

- 定义：按照特定的顺序或者分布安排预训练数据进行训练
- 基于专门构建的评测基准监控大语言模型的关键能力的学习过程，然后在预训练期间动态调整数据的混合配比。



# 数据课程

➤ 数据课程示例

模型	数据课程
CodeLLaMA (代码化)	2T 通用词元 → 500B 代码密集型词元
CodeLLaMA-Python (Python化)	2T 通用词元 → 500B 代码相关词元 → 100B Python 代码相关词元
Llemma (数学化)	2T 通用词元 → 500B 代码相关词元 → 50~200B 数学相关词元
CodeLLaMA (上下文扩展)	2.5T 词元，4K 上下文窗口 → 20B 词元，16K 上下文窗口

➤ 预训练数据的动态批次配比

Algorithm 1: Dynamic Batch Loading

Require: Training data of  $k$  domains  $D_1, D_2, \dots, D_k$ , validation data  $D_1^{\text{val}}, D_2^{\text{val}}, \dots, D_k^{\text{val}}$ , initial data loading weights  $w_0 \in \mathbb{R}^k$ , reference loss  $\ell_{\text{ref}} \in \mathbb{R}^k$ , LM loss  $\mathcal{L}$  or pruning loss  $\mathcal{L}_{\text{prune}}$ , training steps  $T$ , evaluation per  $m$  steps, model parameters  $\theta$  ( $\theta, z, \phi, \lambda$  for pruning)

for  $t = 1, \dots, T$  do

if  $t \bmod m = 0$  then

$\ell_t[i] \leftarrow \mathcal{L}(\theta, z, D_i^{\text{val}})$  if pruning else  $\mathcal{L}(\theta, D_i^{\text{val}})$

$\Delta_t[i] \leftarrow \max\{\ell_t[i] - \ell_{\text{ref}}[i], 0\}$

$w_t \leftarrow \text{UpdateWeight}(w_{t-m}, \Delta_t)$

▷ Calculate loss difference

▷ Update data loading proportion

end

Sample a batch of data  $\mathcal{B}$  from  $D_1, D_2, \dots, D_k$  with proportion  $w_t$ ;

if pruning then

Update  $\theta, z, \phi, \lambda$  with  $\mathcal{L}_{\text{prune}}(\theta, z, \phi, \lambda)$  on  $\mathcal{B}$

else

Update  $\theta$  with  $\mathcal{L}(\theta, \mathcal{B})$

end

end

Subroutine  $\text{UpdateWeight}(w, \Delta)$

$\alpha \leftarrow w \cdot \exp(\Delta)$

$w \leftarrow \frac{\alpha}{\sum_i \alpha[i]}$

return  $\theta$

▷ Calculate the unnormalized weights

▷ Renormalize the data loading proportion

Figure 6 consists of two charts. The left chart is a line graph showing the 'Domain Weight' (y-axis, 0.0 to 0.8) versus the '#Tokens for Training (B)' (x-axis, 0 to 50). It features several lines representing different domains: CC (blue), GitHub (red), Book (green), SE (purple), Wiki (yellow), Arxiv (teal), and C4 (orange). The CC line starts at approximately 0.75 and decreases to about 0.45. The C4 line starts at 0.0 and increases to about 0.45. Other lines remain relatively flat or show minor fluctuations. The right chart is a horizontal bar chart showing the 'Data Proportion in Training (%)' (x-axis, 0 to 70) for each domain. It compares 'Original' (solid blue bars) and 'Dynamic Batch Loading' (hatched blue bars) methods. The domains and their proportions are: CC (Original: 67.0%, Dynamic: 36.1%), GitHub (Original: 4.5%, Dynamic: 0.8%), Book (Original: 4.5%, Dynamic: 9.1%), SE (Original: 2.0%, Dynamic: 1.0%), Wiki (Original: 4.5%, Dynamic: 3.1%), Arxiv (Original: 2.5%, Dynamic: 0.7%), and C4 (Original: 15.0%, Dynamic: 49.2%).

Domain	Original (%)	Dynamic Batch Loading (%)
CC	67.0%	36.1%
GitHub	4.5%	0.8%
Book	4.5%	9.1%
SE	2.0%	1.0%
Wiki	4.5%	3.1%
Arxiv	2.5%	0.7%
C4	15.0%	49.2%

Figure 6: Left: Data weight of each batch during the continued pre-training stage. Right: Cumulative data usage for each domain.

与DoReMi较为相似，逐批次地优化模型  
的训练数据配比

训练配比随Batch优化逐步改变  
对于配比优化的粒度更细

Sheared LLaMA: Accelerating Language Model Pre-training via Structured Pruning, arXiv 2024



谢谢