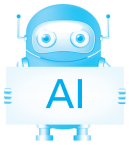
第 5 章



大模型预训练数据

一般情况下，用于预训练的都是具备复杂网络结构、众多参数量，以及足够大数据集的大模型。在自然语言处理领域，预训练模型往往是语言模型（图 5-1），其训练是无监督的，可以获得大规模语料。同时，语言模型又是许多典型自然语言处理任务的基础，如机器翻译、文本生成、阅读理解等。

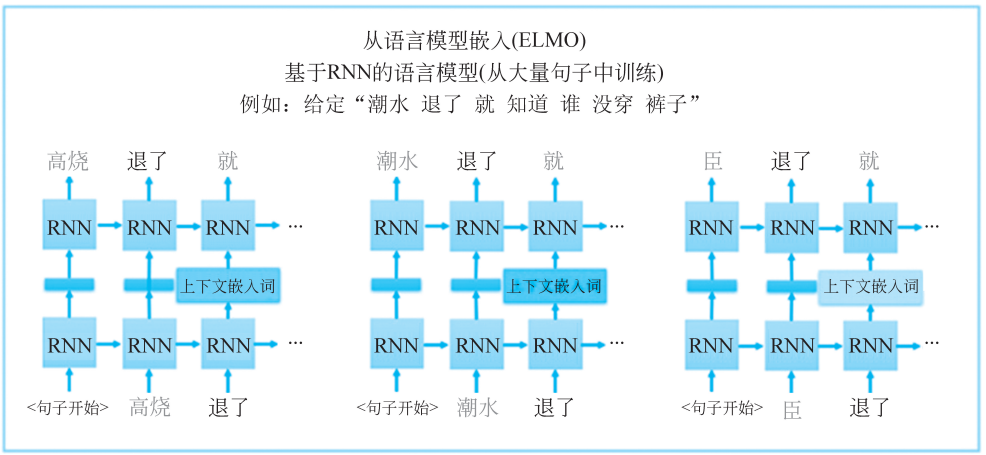


图 5-1　从语言模型嵌入

（1）在 RNN（循环神经网络）模型中，每一个词嵌入的输出要参考前面已经输入过的数据，所以叫作上下文词嵌入。

（2）除了考虑每个词嵌入前文，还要考虑后文，所以再从句尾向句首训练。

（3）使用多层隐藏层后，最终的词嵌入＝该词所有层的词嵌入的加权平均（图 5-2）。

训练大模型需要数万亿各类型数据。如何构造海量“高质量”数据对于大模型的训练至关重要。研究表明，预训练数据是影响大模型效果及样本泛化能力的关键因素之一。大模型预训练数据要覆盖尽可能多的领域、语言、文化和视角，通常来自网络、图书、论文、百科和社交媒体等。

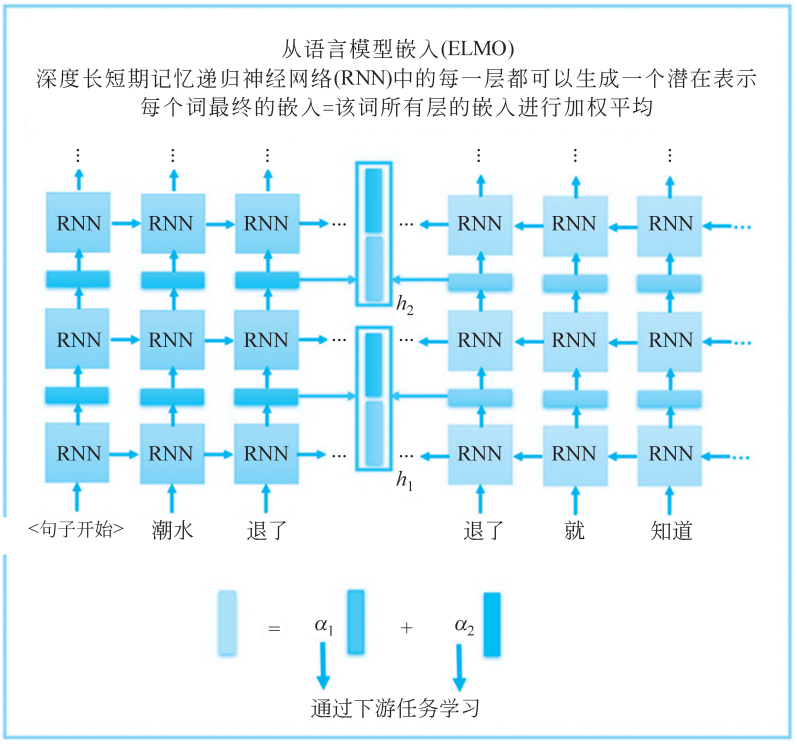


图 5-2 从句子中训练

5.1 数据来源

OpenAI 训练 GPT-3 使用的主要数据来源，包含经过过滤的 CommonCravwl、WebText 2、Books1、Books2 及英文维基百科等数据集。其中 CommonCrawl 的原始数据有 45TB，过滤后仅保留了 570GB 的数据。通过词元方式对上述数据进行切分，大约包含 5000 亿个词元。为了保证模型使用更多高质量数据进行训练，训练 GPT-3 时，根据数据来源的不同，设置不同的采样权重。在完成 3000 亿个词元的训练时，英文维基百科的数据平均训练轮数为 3.4 次，而 CommonCrawl 和 Books2 仅有 0.44 次和 0.43 次。举另一个例子，由于 CommonCrawl 数据集的过滤过程烦琐复杂，Meta 公司的研究人员在训练 OPT 模型时，采用了混合 RoBERTa、Pile 和 PushShift.io Reddit 数据的方法。由于这些数据集中包含的绝大部分数据都是英文数据，因此 OPT 也从 CommonCrawl 数据集中抽取了部分非英文数据加入训练数据。

大模型预训练所需的数据来源大体上分为通用数据和专业数据两大类。

5.1.1 通用数据

通用数据在大模型训练数据中的占比非常高，主要包括网页、图书、新闻、对话文本等不同类型的数据，具有规模大、多样性和易获取等特点，因此支持大模型的语言建模和泛化能力。

网页是通用数据中数量最多的一类。随着互联网的日益普及，人们通过网站、论坛、博客、App 创造了海量的数据。网页数据使语言模型能够获得多样化的语言知识，并增强其

泛化能力。爬取和处理海量网页内容并不是一件容易的事情，因此一些研究人员构建了ClueWeb09、ClueWeb12、SogouT-16、CommonCrawl等开源网页数据集。虽然这些爬取的网络数据包含大量高质量的文本（如维基百科），但也包含非常多低质量的文本（如垃圾邮件等）。因此，过滤并处理网页数据以提高数据质量，对大模型训练非常重要。

图书是人类知识的主要积累方式之一，从古代经典到现代学术著作，承载了丰富多样的人类思想。图书通常包含广泛的词汇，包括专业术语、文学表达及各种主题词汇。利用图书数据进行训练，大模型可以接触多样化的词汇，从而提高其对不同领域和主题的理解能力。相较于其他数据库，图书也是最重要的，甚至是唯一的长文本书面语的数据来源。图书提供了完整的句子和段落，使大模型可以学习到上下文之间的联系。这对于模型理解句子中的复杂结构、逻辑关系和语义连贯性非常重要。图书涵盖了各种文体和风格，包括小说、科学著作、历史记录等。用图书数据训练大模型，可以使模型学习不同的写作风格和表达方式，提高大模型在各种文本类型上的能力。受限于版权因素，开源图书数据集很少，现有的开源大模型研究通常采用Pile数据集中提供的Books 3和BookCorpus 2数据集。

对话文本是指有两个或更多参与者交流的文本内容。对话文本包含书面形式的对话、聊天记录、论坛帖子、社交媒体评论等。研究表明，对话文本可以有效增强大模型的对话能力，并潜在地提高大模型在多种问答任务上的表现。对话文本可以通过收集、清洗、归并等过程从社会媒体、论坛、邮件组等处构建。相较于网页数据，对话文本数据的收集和处理会困难一些，数据量也少很多。常见的对话文本数据集包括PushShift.io Reddit、Ubuntu Dialogue Corpus、Douban Conversation Corpus、Chromium Conversations Corpus等。此外，还提出了使用大模型自动生成对话文本数据的UltraChat方法。

5.1.2　专业数据

专业数据包括多语言数据、科学文本数据、代码及领域特有资料等。虽然专业数据在大模型中所占的比例通常较低，但是其对改进大模型在下游任务上的特定解决能力有着非常重要的作用。专业数据种类非常多，大模型使用的专业数据主要有3类。

多语言数据对于增强大模型的语言理解和生成多语言能力具有至关重要的作用。当前的大模型训练除了需要目标语言中的文本，通常还要整合多语言数据库。例如，BLOOM的预训练数据包含46种语言的数据，PaLM的预训练数据中甚至包含高达122种语言的数据。研究发现，通过多语言数据混合训练，预训练模型可以在一定程度上自动构建多语言之间的语义关联。因此，多语言数据混合训练可以有效提升翻译、多语言摘要和多语言问答等任务能力。此外，由于不同语言中不同类型的知识获取难度不同，多语言数据还可以有效地增加数据的多样性和知识的丰富性。

科学文本数据包括教材、论文、百科及其他相关资源。这些数据对于提升大模型在理解科学知识方面的能力具有重要作用。科学文本数据的来源主要包括arXiv论文、PubMed论文、教材、课件和教学网页等。由于科学领域涉及众多专业领域且数据形式复杂，通常还需要对公式、化学式、蛋白质序列等采用特定的符号标记，并进行预处理。例如，公式可以用LaTeX语法表示，化学结构可以用简化的分子输入管路输入系统（Simplified Molecular Input Line Entry System，SMILES）表示，蛋白质序列可以用单字母代码或三字母代码表示。这样可以将不同格式的数据转换为统一的形式，使大模型更好地处理和分析科学文本

数据。

代码是进行程序生成任务所必需的训练数据。研究表明，通过在大量代码上进行预训练，大模型可以有效提升代码生成的效果。程序代码除本身之外，还包含大量的注释信息。与自然语言文本不同，代码是一种格式化语言，对应着长程依赖和准确的执行逻辑。代码的语法结构、关键字和特定的编程范式都对其含义和功能起着重要作用。代码的主要来源是编程问答社区和公共软件仓库。编程问答社区中的数据包含了开发者提出的问题、其他开发者的回答及相关代码示例。这些数据提供了丰富的语境和真实世界中的代码使用场景。公共软件仓库中的数据包含了大量的开源代码，涵盖多种编程语言和不同领域。这些代码库中的很多代码经过了严格的代码评审和实际的使用测试，因此具有一定的可靠性。

5.2 数据处理

由于数据质量对于大模型的影响非常大，因此，收集各种类型的数据之后，需要对数据进行处理，去除低质量数据、重复数据、有害信息、个人隐私等内容，进行词元切分。

5.2.1 质量过滤

互联网上的数据质量参差不齐，因此，从收集到的数据中删除过滤掉低质量数据是大模型训练中的重要步骤，其方法大致分为两类：基于分类器的方法和基于启发式的方法。

（1）**基于分类器的方法**。目标是训练文本质量判断模型，利用该模型识别并过滤低质量数据。GPT-3、PaLM 和 GLaM 模型在训练数据构造时都使用了基于分类器的方法。例如，基于特征哈希的线性分类器，可以非常高效地完成文本质量判断。该分类器使用一组精选文本（维基百科、书籍和一些选定的网站）进行训练，目标是给予训练数据类似的网页较高分数。利用这个分类器可以评估网页的内容质量。在实际应用中，还可以通过使用 Pareto 分布对网页进行采样，根据其得分选择合适的阈值，从而选定合适的数据集。然而，一些研究发现，基于分类器的方法可能会删除包含方言或者口语的高质量文本，从而损失一定的多样性。

（2）**基于启发式的方法**。通过一组精心设计的规则来消除低质量文本，BLOOM 和 Gopher 采用了基于启发式的方法。一些启发式规则如下。

① 语言过滤：如果一个大模型仅关注一种或几种语言，则可以大幅过滤数据中其他语言的文本。

② 指标过滤：利用评测指标也可以过滤低质量文本。例如，可以使用语言模型对给定文本的困惑度进行计算，利用该值过滤非自然的句子。

③ 统计特征过滤：针对文本内容可以计算包括标点符号分布、符号字比、句子长度在内的统计特征，利用这些特征过滤低质量数据。

④ 关键词过滤：根据特定的关键词集，可以识别并删除文本中的噪声或无用元素。例如，HTML 标签、超链接及冒犯性词语等。

在大模型出现之前，在自然语言处理领域已经开展了很多文章质量判断相关的研究，主要应用于搜索引擎、社交媒体、推荐系统、广告排序及作文评分等任务中。在搜索和推荐系统中，内容结果的质量是影响用户体验的重要因素之一，因此，此前很多工作都是针对用户生成内容的质量进行判断的。自动作文评分也是文章质量判断领域的一个重要子任务，自

1998 年提出使用贝叶斯分类器进行作文评分预测以来，基于 SVM、CNN-RNN、BERT 等方法的作文评分算法相继提出，并取得了较大的进展。这些方法都可以应用于大模型预训练数据过滤。由于预训练数据量非常大，并且对质量判断的准确率要求并不很高，因此一些基于深度学习和预训练的方法还没有应用于低质过滤中。

5.2.2　冗余去除

研究表明，大模型训练数据库中的重复数据会降低大模型的多样性，并可能导致训练过程不稳定，从而影响模型性能。因此，需要对预训练数据库中的重复数据进行处理，去除其中的冗余部分。文本冗余发现也被称为文本重复检测，是自然语言处理和信息检索中的基础任务之一，其目标是发现不同粒度上的文本重复，包括句子、段落、文档、数据集等不同级别。在实际产生预训练数据时，冗余去除需要从不同粒度着手，这对改善语言模型的训练效果具有重要作用。

在句子级别上，包含重复单词或短语的句子很可能造成语言建模中引入重复的模式。这对语言模型来说会产生非常严重的影响，使模型在预测时容易陷入重复循环。重复循环对语言模型生成的文本质量的影响非常大，因此在预训练数据中需要删除这些包含大量重复单词或者短语的句子。

在文档级别上，大部分大模型依靠文档之间的表面特征相似度（例如 n-gram 重叠比例）进行检测并删除重复文档。LLaMA 采用 CCNet 处理模式，先将文档拆分为段落，并把所有字符转换为小写字符，将数字替换为占位符，删除所有 Unicode 标点符号和重音符号，对每个段落进行规范化处理。然后，使用 SHA-1 方法为每个段落计算一个哈希码，并使用前 64 位数字作为键。最后，利用每个段落的键进行重复判断。RefinedWeb 先去除页面中的菜单、标题、页脚、广告等内容，仅抽取页面中的主要内容。在此基础上，在文档级别进行过滤，使用 n-gram 重复程度来衡量句子、段落及文档的相似度。如果超过预先设定的阈值，则会过滤重复段落或文档。

此外，数据集级别上也可能存在一定数量的重复情况，比如很多大模型预训练数据集都会包含 GitHub、维基百科、C4 等。需要特别注意预训练数据中混入测试数据，造成数据集污染的情况。

5.2.3　隐私消除

由于绝大多数预训练数据源于互联网，因此不可避免地会包含涉及敏感或个人信息的用户生成内容，这可能会增加隐私泄露的风险。因此，有必要从预训练数据库中删除包含个人身份信息的内容。

删除隐私数据最直接的方法是采用基于规则的算法，BigScience ROOTS Corpus 在构建过程中就采用了基于命名实体识别的方法，利用算法检测姓名、地址、电话号码等个人信息内容，并进行删除或者替换。该方法被集成在 muliwai 类库中，使用了基于 Transformer 的模型，并结合机器翻译技术，可以处理超过 100 种语言的文本，消除其中的隐私信息。

5.2.4　词元切分

传统的自然语言处理通常以单词为基本处理单元，模型都依赖预先确定的词表，在编码输

入词序列时，这些词表示模型只能处理词表中存在的词。因此，使用时如果遇到不在词表中的未登录词，模型无法为其生成对应的表示，只能给予这些未登录词一个默认的通用表示。

在深度学习模型中，词表示模型会预先在词表中加入一个默认的“[UNK]”标识表示未知词，并在训练的过程中将［UNK］的向量作为词表示矩阵的一部分一起训练，通过引入相应机制来更新［UNK］向量的参数。使用时，对全部未登录词使用［UNK］向量作为表示向量。此外，基于固定词表的词表示模型对词表大小的选择比较敏感。当词表过小时，未登录词的比例较高，影响模型性能；当词表过大时，大量低频词出现在词表中，这些词的词向量很难得到充分学习。在理想模式下，词表示模型应能覆盖绝大部分输入词，并避免词表过大造成的数据稀疏问题。

为了缓解未登录词问题，一些工作通过利用亚词级别的信息构造词表示向量。一种直接的解决思路是为输入建立字符级别表示，并通过字符向量的组合获得每个单词的表示，以解决数据稀疏问题。然而，单词中的词根、词缀等构词模式往往跨越多个字符，基于字符表示的方法很难学习跨度较大的模式。为了充分学习这些构词模式，研究人员提出了子词元化方法，以缓解未登录词问题。词元表示模型会维护一个词元词表，其中既存在完整的单词，也存在形如“c”“re”“ing”等单词的部分信息，称为子词。词元表示模型对词表中的每个词元计算一个定长向量表示，供下游模型使用。对于输入的词序列，词元表示模型将每个词拆分为词表内的词元。例如，将单词“reborn”拆分为“re”和“born”。模型随后查询每个词元的表示，将输入重新组成词元表示序列。当下游模型需要计算一个单词或词组的表示时，可以将对应范围内的词元表示合成需要的表示。因此，词元表示模型能够较好地解决自然语言处理系统中未登录词的问题。词元分析是将原始文本分割成词元序列的过程。词元切分也是数据预处理中至关重要的一步。

字节对编码是一种常见的子词词元算法。该算法采用的词表包含最常见的单词及高频出现的子词。使用时，常见词通常位于字节对编码词表中，而罕见词通常能被分解为若干个包含在字节对编码词表中的词元，从而大幅度减小未登录词的比例。字节对编码算法包括以下两部分。

（1）词元词表的确定。

（2）全词切分为词元及词元合并为全词的方法。

5.3 数据影响分析

过去，自然语言处理是一个任务用标注数据训练一个模型，而现在可以在大量无标注的语料上预训练出一个在少量有监督数据上微调就能做很多任务的模型。这其实就比较接近人类学习语言的过程。例如，参加某个考试测试英文能力的好坏，里面有听说读写等各式各样的任务，有填空和选择等很多题型。但我们学习英文的方法并不是去做大量的选择题，而是背大量的英文单词，理解它的词性、意思，阅读大量的英文文章，掌握它在段落中的用法，你只需做少量的选择题，就可以通过某个语言能力的测试。这便是自然语言处理领域所追求的目标。我们期待可以训练一个模型，它真的了解人类的语言，需要解决各式各样任务的时候，只需要稍微微调一下，它就知道怎么做了（图 5-3）。

大模型训练需要大量计算资源，通常不可能进行多次。有千亿级参数量的大模型进行

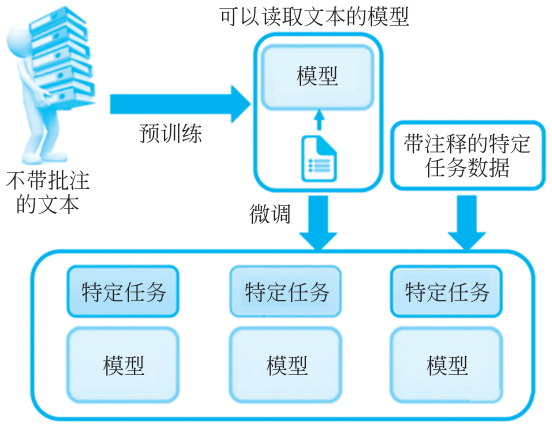


图 5-3　在预训练基础上微调

一次预训练就需要花费数百万元的计算成本。因此，训练大模型之前，构建一个准备充分的预训练数据库尤为重要。

5.3.1　数据规模

随着大模型参数规模的增加，为了有效地训练模型，需要收集足够数量的高质量数据。在针对模型参数规模、训练数据量及总计算量与模型效果之间关系的研究被提出之前，大部分大模型训练所采用的训练数据量相较于 LLaMA 等新的大模型都少很多。

DeepMind 的研究人员描述了他们训练 400 多个语言模型后得出的分析结果(模型的参数量从 7000 万个到 160 亿个，训练数据量从 5 亿个词元到 5000 亿个词元)。研究发现，如果希望模型训练达到计算最优，则模型大小和训练词元数量应该等比例缩放，即模型大小加倍，则训练词元数量也应该加倍。为了验证该分析结果，他们使用与 Gopher 语言模型训练相同的计算资源，根据上述理论预测了 Chinchilla 语言模型的最优参数量与词元量组合。最终确定 Chinchilla 语言模型具有 700 亿个参数，使用了 1.4 万亿个词元进行训练。通过实验发现，Chinchilla 在很多下游评估任务中都显著地优于 Gopher(280B)、GPT-3(175BJurassic-1(178B)及 Megatron-Turing NLG(530B)。

5.3.2　数据质量

数据质量是影响大模型训练效果的关键因素之一。大量重复的低质量数据会导致训练过程不稳定，模型训练不收敛。研究表明，训练数据的构建时间、噪声或有害信息、数据重复率等因素，都对语言模型性能产生较大影响，在经过清洗的高质量数据上训练数据可以得到更好的性能。

Gopher 语言模型在训练时针对文本质量进行相关实验，具有 140 亿个参数的模型在 OpenWebText、C4 及不同版本的 MassiveWeb 数据集上训练得到模型效果对比。他们分别测试了利用不同数据训练得到的模型在 Wikitext103 单词预测、CuraticCorpus 摘要及 Lambada 书籍级别的单词预测三个下游任务上的表现。从结果可以看到，使用经过过滤和去重的 MassiveWeb 数据集训练得到的语言模型，在三个任务上都远好于使用未经处理的数据训练所得到的模型。使用经过处理的 MassiveWeb 数据集训练得到的语言模型在下游

任务上的表现也远好于使用 OpenWebText 和 C4 数据集训练得到的结果。

构建 GLaM 语言模型时，也对训练数据质量的影响进行了分析。实验结果可以看到使用高质量数据训练的模型在自然语言生成和理解任务上表现更好。特别是，高质量数据对自然语言生成任务的影响大于自然语言理解任务。这可能是因为自然语言生成任务通常需要生成高质量的语言，过滤预训练数据库对语言模型的生成能力至关重要。预训练数据的质量在下游任务的性能中也扮演着关键角色。

来自不同领域、使用不同语言、应用于不同场景的训练数据具有不同的语言特征，包含不同语义知识。通过使用不同来源的数据进行训练，大模型可以获得广泛的知识。

5.4 典型的开源数据集

随着基于统计机器学习的自然语言处理算法的发展，以及信息检索研究的需求增加，特别是对深度学习和预训练语言模型研究的深入，研究人员构建了多种大规模开源数据集，涵盖网页、图书、论文、百科等多个领域。构建大模型时，数据的质量和多样性对于提高模型性能至关重要。为了推动大模型研究和应用，学术界和工业界也开放了多个针对大模型的开源数据集。

5.4.1 Pile 数据集

Pile 数据集是一个用于大模型训练的多样性大规模文本数据库，由 22 个不同的高质量子集构成，包括现有的和新构建的，主要来自学术或专业领域。这些子集包括 Pile-CC（清洗后的 CommonCrawl 子集）、Wikipedia、OpenWebText2、ArXiv、PubMed Central 等。Pile 数据集的特点是包含大量多样化文本，涵盖不同领域和主题，从而提高了训练数据集的多样性和丰富性。Pile 数据集包含 825GB 英文文本，其数据类型的主要构成如图 5-4 所示，所占面积大小表示数据在整个数据集中所占的规模。

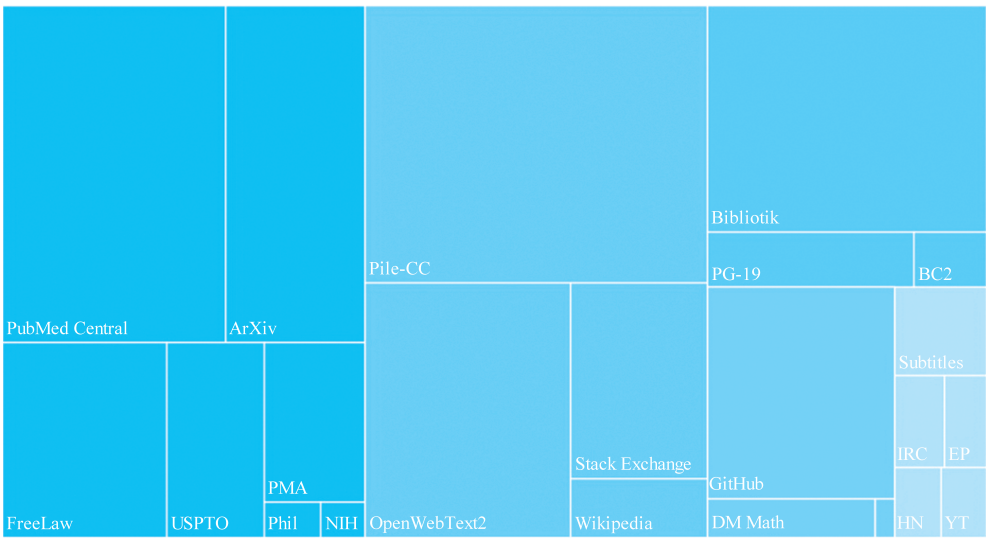


图 5-4 Pile 数据集的主要构成

Pile 数据集的部分子集简单介绍如下。

(1) Pile-CC：通过在 Web Archive 文件上使用 jusText 方法提取，比直接使用 WET 文件产生更高质量的输出。

(2) PubMed Central(PMC)：是由美国国家生物技术信息中心(NCBI)运营的 PubMed 生物医学在线资源库的一个子集，提供对近 500 万份出版物的开放全文访问。

(3) OpenWebText2(OWT2)：是一个基于 WebText1 和 OpenWobTextCorpts 的通用数据集，它包括来自多种语言的文本内容、网页文本元数据，以及多个开源数据集和开源代码库。

(4) ArXiv：是一个自 1991 年开始运营的研究论文预印版本发布服务平台。论文主要集中在数学、计算机科学和物理领域。ArXiv 上的论文是用 LaTeX 编写的，其中公式、符号、表格等内容的表示非常适合语言模型学习。

(5) GitHub：是一个大型的开源代码库，对于语言模型完成代码生成、代码补全等任务具有非常重要的作用。

(6) FreeLaw：是一个非营利项目，为法律学术研究提供访问和分析工具。CourtListener 是 FreeLaw 项目的一部分，包含美国联邦和州法院的数百万法律意见，并提供批量下载服务。

(7) Stack Exchange：是一个围绕用户提供问题和答案的网站集合，其中 Stack Exchange Data Dump 包含了网站集合中所有用户贡献的内容的匿名数据集。它是最大的问题—答案对数据集之一，包括编程、园艺、艺术等主题。

(8) USPTO：是美国专利商标局授权专利背景数据集，源于其公布的批量档案。该数据集包含大量关于应用主题的技术内容，如任务背景、技术领域概述、建立问题空间框架等。

(9) Wikipedia(English)：是维基百科的英文部分。维基百科旨在提供各种主题的知识，是世界上最大的在线百科全书之一。

(10) PubMed：是由 PubMed 的 3000 万份出版物的摘要组成的数据集。它是由美国国家医学图书馆运营的生物医学文章在线存储库，它还包含 1946 年至今的生物医学摘要。

(11) OpenSubtitles：是由英文电影和电视的字幕组成的数据集。字幕是对话的重要来源，并且可以增强模型对虚构格式的理解，对创造性写作任务(如剧本写作、演讲写作、交互式故事讲述等)有一定作用。

(12) DeepMind Mathematics(DM Math)：以自然语言提示形式给出，由代数、算术、微积分、数论和概率等一系列数学问题组成的数据集。大模型在数学任务上的表现较差，这可能是由于训练集中缺乏数学问题。因此，Pile 数据集中专门增加数学问题数据集，期望增强通过 Pile 数据集训练的语言模型的数学能力。

(13) PhilPapers：由国际数据库中的哲学出版物组成，它涵盖了广泛的抽象、概念性话语，文本写作质量也非常高。

(14) NIH：包含 1985 年至今获得 NIH 资助的项目申请摘要，是高质量的科学写作实例。

Pile 中不同数据子集所占比例及训练时的采样权重有很大不同，高质量的数据会有更高的采样权重。例如，Pile-CC 数据集包含 227.12GB 数据，整个训练周期中采样 1 轮，虽然维基百科(英文)数据集仅有 6.38GB 的数据，但是整个训练周期中采样 3 轮。

5.4.2　ROOTS 数据集

ROOTS(Responsible Open-science Open-collaboration Text Sources)，即负责任的开

放科学、开放协作文本源数据集，是 Big-Science 项目在训练具有 1760 亿个参数的 BLOOM 大模型时使用的数据集，其中包含 46 种自然语言和 13 种编程语言，整个数据集约 1.6TB。

ROOTS 的数据主要来源于 4 方面：公开数据、虚拟抓取、GitHub 代码、网页数据。

（1）在公开数据方面，目标是收集尽可能多的各种类型的数据，包括自然语言处理数据集和各类型文档数据集。在收集原始数据集的基础上，进一步从语言和统一表示方面对收集的文档进行规范化处理。识别数据集所属语言并分类存储，将所有数据都按照统一的文本和元数据结构进行表示。

（2）在虚拟抓取方面，由于很多语言的现有公开数据集较少，因此这些语言的网页信息是十分重要的资源补充。在 ROOTS 数据集中，采用网页镜像，选取了 614 个域名，从这些域名下的网页中提取文本内容补充到数据集中，以提升语言的多样性。

（3）在 GitHub 代码方面，针对程序语言，ROOTS 数据集从 BigQuery 公开数据集中选取文件长度为 100～20 万的字符，字母符号占比在 15%～65%，最大行数为 20～1000 行的代码。

（4）在大模型训练中，网页数据对于数据的多样性和数据量支撑都起到重要作用。ROOTS 数据集中包含了 OSCAR 21.09 版本，对应的是 CommonCrawl 2021 年 2 月的快照，占整体 ROOTS 数据集规模的 38%。

数据准备完成后，还要进行清洗、过滤、去重及隐私信息删除等工作，ROOTS 数据集处理流程如图 5-5 所示。整个处理工作采用人工与自动相结合的方法，针对数据中存在的一些非自然语言的文本，例如预处理错误、SEO 页面或垃圾邮件，构建 ROOTS 数据集时会进行一定的处理。

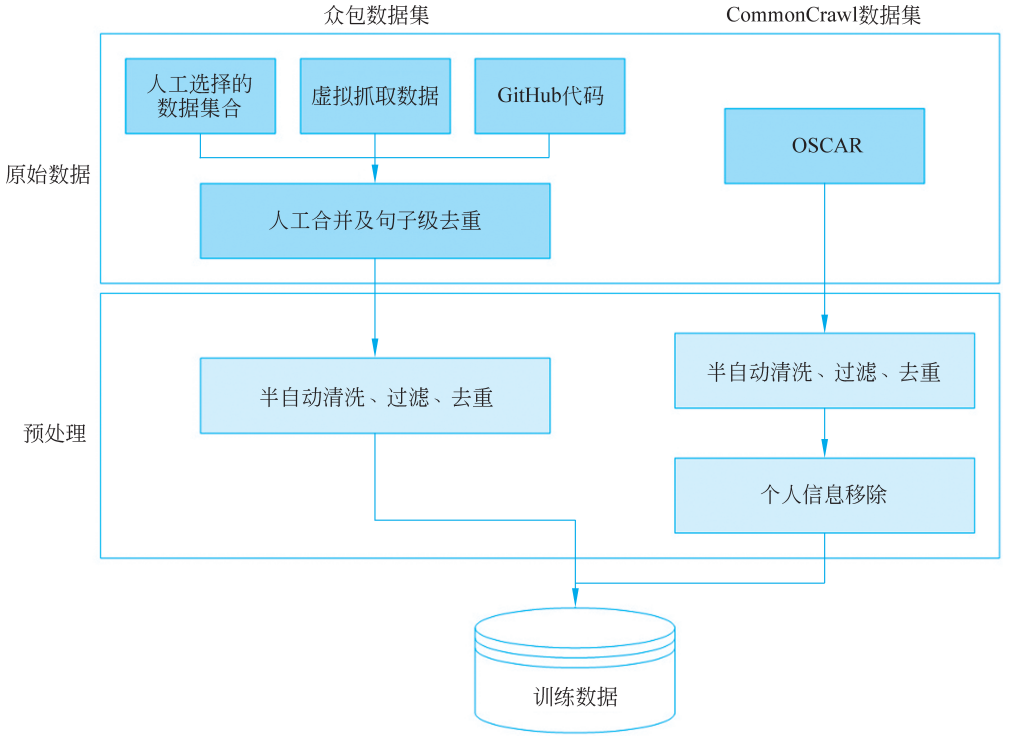


图 5-5 ROOTS 数据集处理流程