“源1.0”大模型
技术白皮书

浪潮电子信息产业股份有限公司
2022年8月

目录

如何训练2457亿参数量的中文巨量模型“源1.0”

从2018年的BERT到2020年的GPT-3，NLP语言模型经历了爆发式的发展过程，其中BERT模型的参数量为3.4亿，而GPT-3的模型参数量达到了1750亿。2021年9月，浪潮发布了“源1.0”，它是目前规模最大的中文AI单体模型，参数规模高达2457亿，训练采用的中文数据集达5TB。“源1.0”在语言智能方面表现优异，获得中文语言理解评测基准CLUE榜单的零样本学习和小样本学习两类总榜冠军。测试结果显示，人群能够准确分辨人与“源1.0”作品差别的成功率低于50%。

海量的参数带来了模型训练和部署上的巨大挑战。本文将聚焦“源1.0”背后的计算挑战以及我们采取的训练方法。

1. “源1.0”的模型结构

“源1.0”是一个典型的语言模型。语言模型通俗来讲就是能够完成自然语言理解或者生成文本的神经网络模型。对于“源1.0”，我们考虑语言模型（Language Model，LM）和前缀语言模型（Prefix Language Model，PLM）两种模型结构。如下图所示：

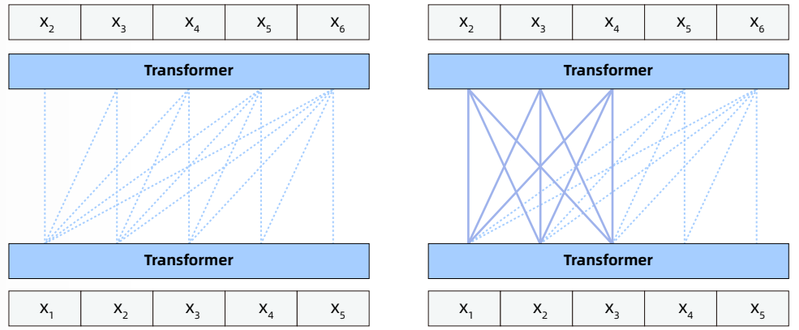


图1 模型结构示意图（左图为LM，右图为PLM）

我们比较了130亿参数的LM和PLM在不同下游任务上的结果，注意到LM在Zero-Shot和Few-Shot上表现更好，而PLM在微调方面表现出色。微调通常会在大多数任务中带来更好的准确性，然而微调会消耗大量的计算资源，这是不经济的。所以我们选择LM作为“源 1.0”模型的基础模型结构。

2. 如何训练“源1.0”

2.1 源1.0训练面对的挑战

“源1.0”的训练需要面对的第一个挑战就是数据和计算量的挑战。

数据方面，如果把训练一个巨量模型的训练过程比作上异常战役的话，那么数据就是我们的弹药。数据量的多少，决定了我们可以训练模型的规模，以及最后的效果。针对这一方面，我们构建了一个全新的中文语料库，清洗后的高质量数据规模达到了5TB，是目前规模最大的中文语料库。

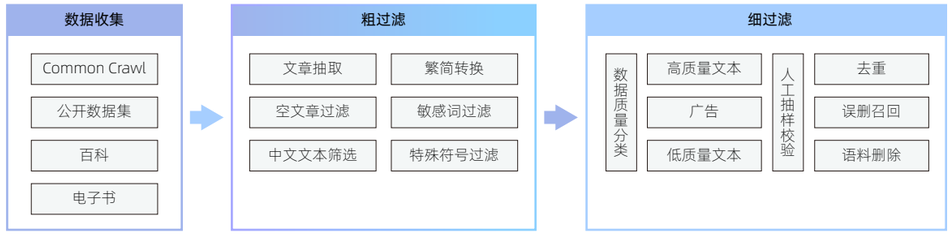


图2 数据预处理流程图

算力方面，根据OpenAI提出的PetaFlop/s-day衡量标准，我们可以估算“源1.0”训练的计算需求情况。根据Wikipedia提供的数据（https://en.wikipedia.org/wiki/OpenAI），GPT-3的计算需求约为3640 PetaFlop/s-day，约等于64个A100 GPU训练1年时间。而“源1.0”的计算需求达到了4095 PetaFlop/s-day。

计算资源的巨大开销是限制研究人员研发具有数以千万计参数的NLP大模型的瓶颈。例如GPT-3是在由10000个GPU所组成的集群上训练得到的。我们在设计“源1.0”的模型结构时，考虑到了影响大规模分布式训练的关键因素，采用了专门的分布式训练策略，从而加速了模型的训练过程。

在模型训练时一般最常用的是采用数据并行分布式计算策略，但这只能满足小模型的训练需求。对于巨量模型来说，由于其模型参数量过大，远远超过常用计算设备比如GPU卡的显存容量，因此需要专门的算法设计来解决巨量模型训练的显存占用问题，同时还需要兼顾训练过程中的GPU计算性能的利用率。

2.2 “源1.0”的训练策略

为了解决显存不足的问题，我们采用了张量并行、流水并行、数据并行相结合的并行策略，实现了在2128个GPU上部署“源1.0”，并完成了1800亿tokens的训练。

一、张量并行

针对单个GPU设备不能完整的承载模型训练，一个解决方案就是张量并行+数据并行的2D并行策略。具体来说，使用多个GPU设备为1组，比如单个服务器内的8个GPU为1组，组内使用张量并行策略对模型进行拆分，组间（服务器间）采用数据并行。

对于张量并行部分，NVIDIA在Megatron-LM中提出了针对Transformer结构的张量并行解决方案。其思路是把每一个block的参数和计算都均匀的拆分到N个GPU设备上，从而实现每个GPU设备都承担这一block的参数量和计算量的1/N效果。图3展示了对Transformer结构中的MLP层和self-attention层进行张量并行拆分计算的过程示意图。

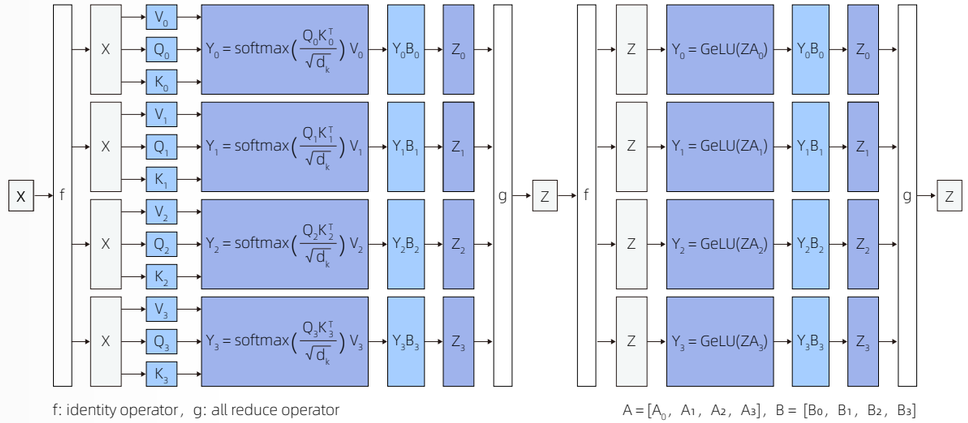


图3 张量并行示意图

在训练过程中，tensor经过每一层的时候，计算量与通信数据量之比f_{tp}如下：

$$f_{tp} = \frac{96t}{8(t\text{-}1)}\left(h + \frac{s}{6}\right)$$

其中，S为输入序列的长度，h为隐藏层的大小（hidden size）。

二、流水并行

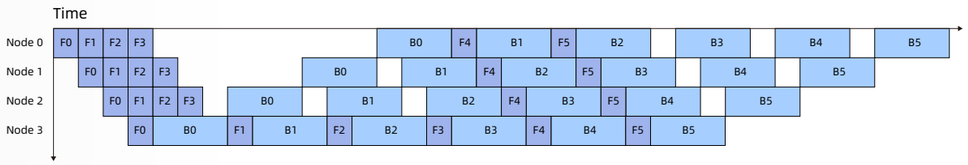


图4 流水线并行示意图

对于具有数千亿参数的语言模型，这些参数很难被存放在单个节点中。流水线并行将LM的层序列在多个节点之间进行分割，以解决存储空间不足的问题，如图5所示。每个节点都是流水线中的一个阶段，它接受前一阶段的输出并将结果过发送到下一阶段。如果前一个相邻节点的输出尚未就绪，则当前节点将处于空闲状态。节点的空闲时间被称为流水线气泡（pipline bubble）。为了提高流水行并行的性能，我们必须尽可能减少在气泡上花费的时间。定义流水线中气泡的理想时间占比 f_{pb}为如下形式：

$$f_{pb} = \frac{(L/l\text{-}1)}{m}$$

根据这一公式，流水线气泡的耗时随着层数L的增加而增加，随着微批次大小（micro-batch-size）的增加而减小。当m» L/l的时候，流水并行过程中的流水线气泡对训练性能的影响几乎可以忽略。

与此同时，在流水并行过程中，节点间的计算量与通信数据量之比f_{pp}为：

$$f_{pp} = \frac{24L}{p}\left(h + \frac{s}{6}\right)$$

根据上面的公式，流水线中节点的计算效率与h和S呈线性关系，这与张量并行类似。

三、数据并行

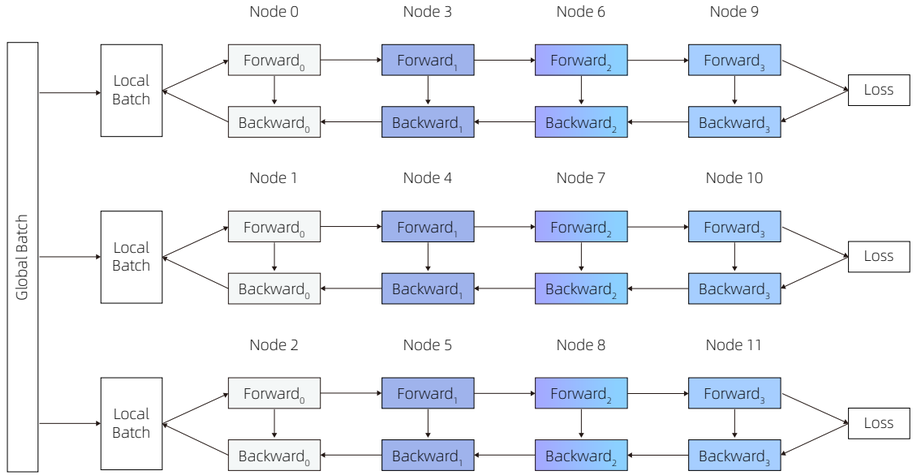


图6 数据并行示意图

采用数据并行时，全局批次大小（global batch size）按照流水线分组进行分割。每个流水线组都包含模型的一个副本，数据在组内按照局部批次规模送入模型副本。数据并行时的计算量与通信数据量的比值f_{dp}可用如下公式近似：

$$f_{dp} \approx \frac{4BSd}{d\text{-}1}$$

当d» 1时，上面公式可以进一步简化成：

$$f_{dp} \approx 4BS$$

根据这一公式，我们可以看出数据并行的计算效率与全局批次大小B和序列长度S呈正比关系。由于模型对内存的需求与S的平方成正比，与B成线性关系，因此增加全局批次大小可以更有效的提升数据并行的效率。

当全局批次大小过大的时候，模型很容易出现不收敛的问题，为了保证模型训练过程的稳定性，我们将全局批次大小限制在了10^7个token内。

根据以上的理论分析，我们确定了设计“源1.0”巨量模型结构的基本原则：

·尽可能增加序列长度，因为它有利于张量并行、流水线并行和数据并行。由于内存占用与序列长度的平方成正比，因此有必要在反向传播时重新计算激活函数，以节省内存开销。

·语言模型中层数太多会对性能产生负面影响，因为这会增加在流水线气泡上的时间消耗。

·增加隐藏层大小可以提高张量并行和流水线并行的性能。

·增加节点中的微批次大小可以提高流水线并行效率，增加全局批次大小可以提升数据并行的效率。

在这一设计原则的基础上，我们设计的“源1.0”的模型结构以及分布式策略的设置如下表所示：

Model	Layers	Hidden size	Global BS	Micro BS	Sequence Length	t	p	d	GPUs
Yuan 1.0	76	16384	3360	1	2048	8	38	7	2128

结合模型结构的特性以及我们使用集群的硬件特性，我们如下的节点配置和分布式策略选择：

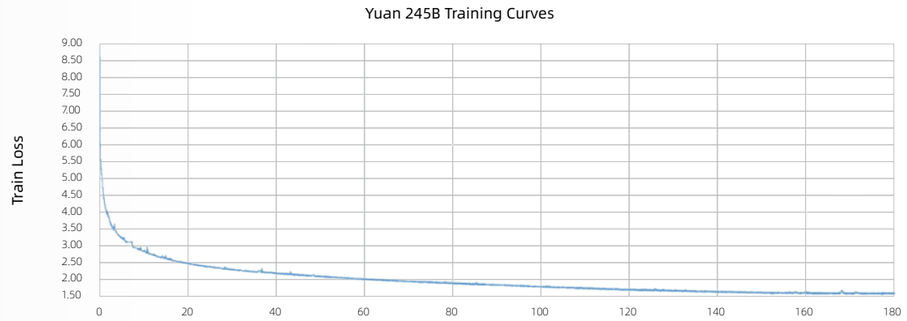
· “源1.0”模型在训练过程中共使用了2128个GPU

· 模型分成了7组，每组38台AI服务器，里面放置一个完整的“源1.0”模型，7组之间采用数据并行

· 每组的38个服务器，采用流水并行每个服务器放置1/38的模型（2个Transformer Layer），一共76层

· 在每台服务器内采用张量并行，按照Transformer结构的每一层进行均匀切分

模型收敛曲线如下图：



关于“源1.0”的更多信息，大家可以参照浪潮发布在arxiv上的论文：https://arxiv.org/abs/2110.04725。

另外，我们也整理了更详细的《NLP模型训练解决方案白皮书》。感兴趣的读者可以关注“浪潮AIHPC”公众号，回复“NLP白皮书”即可下载。

中文巨量模型“源1.0”的小样本学习优化方法

最近，浪潮发布了中文巨量模型“源1.0”，参数量达2457亿，超越美国OpenAI组织研发的GPT-3。“源1.0”在语言智能方面表现优异，获得中文语言理解评测基准CLUE榜单的零样本学习（zero-shot）和小样本学习（few-shot）两类总榜冠军。在零样本学习榜单中，“源1.0”超越业界最佳成绩18.3%，在6项任务中获得冠军；在小样本学习的4项任务获得冠军。在成语阅读理解填空项目中，源1.0的表现已超越人类得分。

为了提高“源1.0”在不同下游任务的泛化性和精度，我们采用了多种小样本学习优化策略。本文介绍了标签扩充和校正相结合的小样本学习优化方法。该方法不仅能消除预训练语料中标签出现频率不同而带来偏置，而且通过空文本或数据集校正标签词和输入样本带来的偏置，可使巨量模型避免再训练，降低了内存需求和系统复杂性，而且大大提高了下游任务的准确率和稳定性。

根据在下游任务推理时提供的样本数目，我们进一步将表述专门化为“零样本”和“小样本”。

1．采用零样本和小样本学习的原因

人类可以仅通过一个或几个示例就可以轻松地建立对新事物的认知，而机器学习算法通常需要成千上万个有监督样本来保证其泛化能力。拥有从零样本、小样本中学习和概括的能力，是人工智能向人类智能发展的重要标志。简单来说，零样本学习就是训练的模型不仅仅能够识别出训练集中已有的数据类别，还可以对未见过的类别的数据进行区分；小样本学习就是使用远小于深度学习所需要的数据样本量，达到接近甚至超越大数据深度学习的效果，如图1。

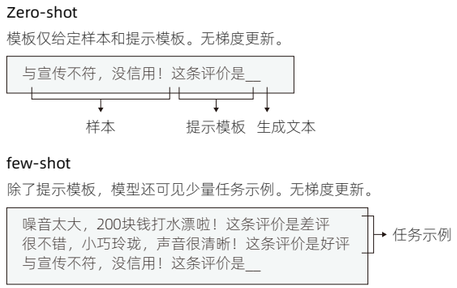


图1 零样本学习和小样本学习示例

对“源1.0”这样的超大型模型进行微调比较困难而且成本很高，因此希望固定它们的模型参数，然后将不同的优化策略应用到不同下游任务上。最新的研究成果也表明，在自然语言处理（NLP）领域通过增加模型规模、扩大预训练数据体量、使用更多的计算资源等方式，巨量模型可以在小样本学习甚至零样本学习任务中获得非常出色的性能表现。预训练好的巨量模型不必再经过复杂的“微调”过程，就可以为诸多应用任务泛化支持提供统一强大的算法支撑。

2．零样本和小样本学习优化方法

巨量模型最核心的能力是零样本学习和小样本学习能力。但是基于巨量模型的零样本和小样本学习可能是非常不稳定的：提示模板格式的选择、训练样本、甚至训练样本顺序都可能导致准确性在接近偶然和接近最先进水平之间漂移，这种不稳定性源于语言模型对预测某些答案的偏差，例如，那些被放在提示语末尾附近的答案，或在预训练数据中常见的答案，这些偏差往往会导致模型的输出分布发生变化。因此针对零样本和小样本学习任务，我们提出了一套校准和标签扩展方案来提升模型在下游任务上的性能表现。大量实验结果表明这套方案能够在多项语言处理NLP任务上稳定地提升模型的精度。以下内容以单句分类任务为例进行介绍。

2.1 提示模板设计

2.1.1 零样本学习

经过大量的实验对比，我们发现单句分类任务更适合基于概率生成方式，因此将提示模板设计成填空最后一个词的形式。

任务名称	提示模板
EPRSTMT	样本语句。总体来说，该产品很
CSLDCP	文章：样本语句。该文章是关于
TNEWS	新闻：样本语句。这条新闻是关于
IFLYTEK	广告：样本语句。该广告是关于

表1 单句分类任务最佳提示模板

2.1.2 小样本学习

巨量模型的准确性在很大程度上取决于训练示例的选择和排列。因此我们从训练集中人工挑选三个不同类别（娱乐、文化、体育）示例进行测试排列测试。

	十亿参数	百亿参数	千亿参数
zero	0.49	0.52	0.5282
娱乐、文化、体育	0.2732	0.5492	0.5318
娱乐、体育、文化	0.3187	0.5383	0.5765
文化、娱乐、体育	0.2741	0.5455	0.5829
文化、体育、娱乐	0.3069	0.4936	0.5792
体育、娱乐、文化	0.2550	0.4399	0.5774
体育、文化、娱乐	0.2659	0.5018	0.57468

表2 TNEWS任务3-shot不同示例排序结果对比

从实验数据可知，当模型参数量较小时，模型没有学到足够的知识，此时小样本示例对分类结果起负作用。随着模型参数量增大，小样本学习的效果越明显。

2.2 校准

巨量模型在预训练中会从语料库带来偏置，导致下游任务精度低或性能不稳定。例如，在零样本情感分类设置中，给定“N/A”作为输入，GPT-3 倾向于预测为“positive”而不是“negative”，而本应该分配50/50的概率给这两个相反的标签。另一个问题是同一对象的不同表示（例如，“computer”和“PC”）可能会竞争概率质量，导致任务标签上的分布不理想。因此在实际应用中校正很有必要。具体可参考论文：Calibrate Before Use: Improving Few-Shot Performance of Language Models（https://arxiv.org/abs/2102.09690）。

我们采用解决的方法是通过无文本输入对带偏置的标签词进行补偿，把它们校准为无偏状态，减少不同提示选择之间的差异。

具体实现：

> 输入无文本样例，即将无文本["N/A", " ", "[MASK]"]分别和2.1设计的提示模板组合，如"N/A"与EPRSTMT提示模板组成输入样例：“N/A。总体来说，该产品很__”；

> 将无文本样例输入语言模型，输出标签词位置对应的所有类别概率（logits），并取平均值后归一化得到p_cf；

> 将验证集样本与提示模板组合为验证集样例输入语言模型，输出校正前所有类别概率p_{pre}。

> 根据公式 p_{cal}= softmax(W*p_{pre} + b) 计算校正后类别概率。其中有两种方案：一、当通过W= [diag(p_cf)]$^{-1}$计算校正矩阵时，p_{cal}=softmax(W*p_{pre})；二、当通过b = -1 *p_cf计算校正矩阵时，p_{cal}=softmax(p_{pre} + b)。

需要特别注意的是，为了同时校正标签词和输入样本带来的偏置，我们还提出了将训练集或验证集样本替代无文本["N/A", " ", "[MASK]"]计算校正矩阵的方法，使得模型可以根据输入数据分布进行校正。实验结果如表3所示。

校正方法	校正输入	多分类下游任务		
		TNEWS (15类)	IFLYTEK(119类)	CSLDCP（67类）
校正前精度	无	59.74	42.53	46.28
W = [diag(p_cf)]$^{-1}$, p_{cal} = softmax(W*p_{pre})	N/A	63.57	37.87	42.21
	‘’	62.48	37.22	42.07
	[MASK]	62.02	38.31	43.18
	N/A, ‘’, [MASK]	62.57	38.01	42.89
b = -1 *p_cf p_{cal} = softmax(p_{pre} + b)	N/A, ‘’, [MASK]	60.47	42.75	46.95
训练集identity_W	N/A, ‘’, [MASK]	59.74	43.48	46.66
训练集diagonal_W		63.57	46.25	52.18
验证集identity_W		60.02	43.55	46.76
验证集diagonal_W		64.12	45.74	52.47

表3 不同校正方法在多分类任务上实验结果对比（10亿参数模型）

综合来看，通过数据集校正，效果更佳。

2.3 标签扩展

在理想状态下，所有标签在预训练语料中的出现频率应该大致相同。但是在实验中，我们发现标签在语料中出现的频率存在差异，使得模型对预测结果有偏好性。在实际应用中，人工从接近6万的词表空间中选择符合条件的标签映射词是非常困难的，而且通常会引入主观因素（参考论文Knowledgeable Prompt-tuning: Incorporating Knowledge into Prompt Verbalizer for Text Classification）。因此我们采用word2vec算法和人工结合方式，扩展标签词步骤如下：

（1）通过word2vec初步筛选出与原标签相近，并在词表中的标签；

（2）在初步筛选候选集中再人工精选，删去稀有词，尽可能找先验分布相近的标签词对。

标签序号	原标签	扩充后标签
"100"	["故事"]	["故事", "情节", "讲述", "寓言", "童话"]
"101"	["文化"]	["文化", "人文", "传承", "文学艺术", "价值观"]
"102"	["娱乐"]	["娱乐", "文娱", "影视", "综艺", "传媒"]
"103"	["体育"]	["体育", "竞技", "奥林匹克", "奥林匹克运动", "athletic"]
"104"	["财经"]	["财经", "金融", "基金", "财产", "cfa"]
"106"	["房产"]	["房产", "不动产", "地产", "住房", "置业"]
"107"	["汽车"]	["汽车", "车", "用车", "驾驶", "行车"]
"108"	["教育"]	["教育", "文化教育", "教养", "辅导", "学习"]
"109"	["科技"]	["科技", "技术", "研发", "创新", "科技进步"]
"110"	["军事"]	["军事", "国防", "武装", "战争", "警备"]
"112"	["旅游"]	["旅游", "旅行", "旅游业", "休闲", "游客"]
"113"	["国际"]	["国际", "全球", "世界", "跨国", "国外"]
"114"	["股票"]	["股票", "股市", "股价", "期货", "炒股"]
"115"	["农业"]	["农业", "耕作", "农耕", "农牧", "三农"]
"116"	["电子游戏"]	["电子游戏", "手游", "网游", "游戏", "桌游"]

表4 TNEWS任务标签扩充结果

2.4 标签扩展与校准结合

将2.3扩展标签词与2.2校准相结合提出三种校正优化方案，在实际应用中可根据测试选择：

方法一：先通过模型对扩充后标签映射词计算其相应概率，然后按类别取平均，最后再校正；

方法二：先通过模型对扩充后标签映射词计算其相应概率，然后按类别取最大值，最后再校正；

方法三：先通过模型对扩充后标签映射词计算其相应概率，然后分别进行校正，最后按类别取平均。

扩充+校正优化方法	TNEWS精度
原校正方法	49.00, 55.01
方法一	47.996, 58.56
方法二	50.09, 56.56
方法三	48.00, 57.559

表5 标签扩充与校正结合不同方案结果对比（10亿参数模型）

3．“源1.0”在CLUE榜单单句分类任务上的应用方法举例

中文语言理解测评基准（CLUE，https://www.cluebenchmarks.com/）是目前公认最权威的中文语言模型评估基准，提供了一个由语言学家创建的诊断评估数据集，覆盖多个不同程度的语言任务，具有广泛代表性。代码地址：https://github.com/chineseGLUE/chineseGLUE。

任务名称	训练集	验证集	测试集	任务目标	数据来源
单句分类任务					
EPRSTMT	32	32	610	情感分析	电商产品评论
CSLDCP	536	536	1784	长文本分类	中文科学文献
TNEWS	240	240	2010	短文本分类	今日头条中文新闻
IFLYTEK	928	690	1749	长文本分类	日常生活相关的各类应用主题
句子匹配任务					
OCNLI	32	32	2520	推理	原生中文自然语言推理
BUSTM	32	32	1772	语义相似度	小布助手对话短文本匹配
阅读理解					
CHID	42	42	2002	成语完形填空	成语阅读理解填空
CSL	32	32	2828	摘要关键词判断	中文科技文献
CLUEWSC	32	32	976	代词消歧	模式挑战中文版

表6 CLUE中小样本学习榜单下游任务数据集介绍

单句分类任务包括情感分类（Eprstmt：用于情感分析的电子商务产品评论数据集）、新闻标题分类（Tnews：Toutiao Short Text Classificaiton for News）、应用描述分类（Iflytek：长文本分类）和学科分类（Csldcp：中国科学文献学科分类）。Eprstmt 是二分类任务，包括带有正面和负面产品评论。Tnews、Iflytek 和 Csldcp 是多分类任务，分别有 15、118 和 67 个类别。如果标签为 0 或 1 或是英文，我们会将标签转换为中文。如果标签长于一个标记，我们将标签转换为一个标记的长度，并确保其具有相同或相似的含义。对于所有文本分类任务，标签都附加在句子的末尾，句子和标签直接用提示词链接。我们的生成模型会根据给定的句子预测标签，并计算每个候选标签的概率 P(label | sentence)，其中概率最大的是模型预测结果。

预训练好的“源1.0”千亿参数模型，结合下游任务优化方法，在零样本学习榜单中，超越业界最佳成绩18.3%，并在文献分类、新闻分类，商品分类、原生中文推理、成语阅读理解填空、名词代词关系6项任务中获得冠军。

排行	模型	研究机构	测评时间	Score	EPRSTMT	CSLDCP	TNEWSF	IFLYTEKF	OCNLIF	BUSTM	CHIDF	CSLF	CLUEWSCF
1	Human	CLUE	21-06-18	83.934	90.0	68.0	71.0	66.0	90.3	88.0	87.1	84.0	98.0
2	源1.0	浪潮人工智能...	21-09-27	59.024	84.9933...	47.9159...	64.4666...	34.5769...	45.2179...	57.4999...	87.5	51.6666...	62.7586...
3	OBERT-C-base.ZeroCLUE	selftrun	21-08-19	49.881	78.0876...	26.4088...	52.9333...	25.6923...	37.0395...	69.25	60.85	50.6333...	51.0344...
4	RoBERTa_wwm.Zero-shot	CLUE	21-06-18	47.028	85.2	12.6	25.3	27.7	40.3	50.6	57.6	52.2	50.0
5	NEZHA-Gen.Zero-shot	CLUE	21-06-18	44.166	57.54	26.23	36.96	19.04	34.4	50.0	65.63	50.14	50.31
6	Model.Zero-shot	ZeroCLUE.M	21-08-12	27.128	75.0332...	25.5085...	48.7333...	24.9230...	37.4112...	54.1	0	0	0
7	BertForCLS	bert	21-08-23	26.313	60.1593...	27.9759...	45.7333...	22.4615...	35.9580...	63.9	0	0	0
8	S-Bert	Sysu	21-08-24	17.809	45.1527...	1.60053...	2.80000...	2.30769...	30.8550...	62.55	0	0	0
9	S-Bert-V2	Sysu	21-08-25	16.429	42.6294...	1.70056...	6.66666...	0.5	38.1209...	48.4	0	0	0

图2 Zero榜单打榜结果

关于“源1.0”的更多信息，大家可以参照浪潮人发布在arxiv上的论文：https://arxiv.org/abs/2110.04725。

中文巨量模型“源1.0”：模型结构与生成效果解析

“源 1.0”自2021年9月底发布以来收获了广泛的关注。其参数量达2457亿，超越美国OpenAI组织研发的GPT-3。“源1.0”在语言智能方面表现优异，获得中文语言理解评测基准CLUE榜单的零样本学习（zero-shot）和小样本学习（few-shot）两类总榜冠军。测试结果显示，人群能够准确分辨人与“源1.0”作品差别的成功率低于50%。

在之前的博客中，我们详细论述了如何准备预训练数据、模型本身如何训练，以及在下游任务如何提升精度。在本篇中，我们将着重讨论模型的结构问题，以及由模型结构带来的效果。会回答以下三个问题：（1）“源1.0”基础模型结构是怎样的？（2）为什么要选择这样的结构？（3）和模型结构相关的下游任务效果。

1．“源1.0”基础模型结构的选择

在介绍基础模型结构之前，显然要明确一件事情：我们想让模型完成什么呢？在自然语言处理（NLP）领域，所有的任务大体可以被分为两类：自然语言理解（NLU）任务和自然语言生成（NLG）任务，前者偏重于对语义的理解，而后者偏重于文本的创作。如果可能的话，开发者当然期望这个模型在两类任务上同样出色，但事实上，不同类型的NLP模型结构对两类任务总是有所偏重的。如果只考虑在榜单上的表现，偏重于NLU任务可能会比较合适，因为包括“源1.0”冲击的CLUE榜单在内，几乎所有相似的榜单都偏重于自然语言理解任务，在《中文巨量模型浪潮“源1.0”的小样本学习优化方法》（http://blog.itpub.net/31545803/viewspace-2847859/）这篇博文上也可以看到相关任务的介绍。为了在榜单上取得更好的成绩，自然应该选择一个偏重于NLU的模型结构。然而，当我们考虑到模型实际应用的时候，就会发现NLG的应用场景更广泛，没有NLG，也很难体先出NLU的价值。所以，在这个问题上，我们的认识是要优先保证模型具有出色的创作能力（NLG），而在NLU任务上也务必尽可能地提升效果。

带着这样的初衷，“源1.0”的基础结构为一个单向的语言模型，即根据上文预测下文的概率。其中的Transformer解码器（Decoder）采用自回归的方式输出序列。当处理不同的下游任务时，则会根据任务类型使用一个从文本到文本的框架，将所有任务处理成相似的格式，以便直接将预训练的语言模型应用于不同的下游任务上。过去的研究已经证实，经典的单向语言模型结构是擅长NLG任务的，而在NLU任务上则相对薄弱一些。为了进一步探索模型在NLU任务上的可能，在“源 1.0”的开发过程中，我们考虑了语言模型（Language Model，LM），和前缀语言模型（Prefix Language Model，PLM）两种结构。两种结构的主要区别在于掩码的方式，如图1所示。

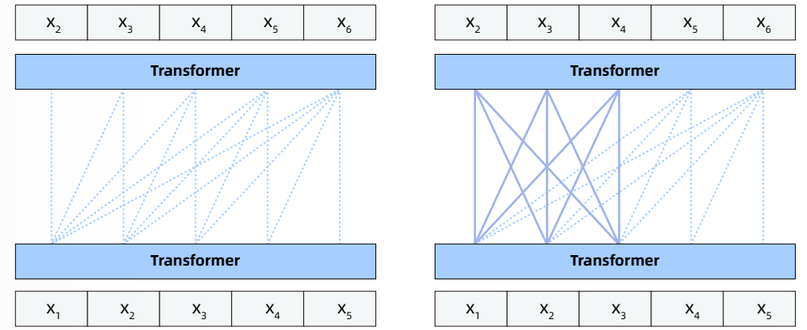


图1 语言模型结构示意图（左图为LM，右图为PLM）。蓝色实线代表输入在前缀范围内全可见掩码，蓝色虚线代表随机掩码

在t时刻，解码器根据模型对输出序列的预测概率，生成输出序列中最右侧的一位 (x5)的标记（token）。之后这个标记与输入序列相连接，一起被送入模型以预测 t+1时刻的输出(x6)的标记。我们用这两种模型结构分别训练了130亿参数量的两个模型，Yuan LM-13B和Yuan PLM-13B，并把这两个模型放在小样本学习（FewCLUE）和零样本学习（ZeroCLUE）场景下做了评估（表1）。关于表格中任务的详细介绍，请参考博文《中文巨量模型浪潮“源1.0”的小样本学习优化方法》。

	Scores	Bustm	Chid	Csl	Csldcp	Eprstmt	Iflytek	Ocnli	Tnews	Wsc
Human	82.48	88	87.1	84	68	90	66	90.3	71	98
SOTA	52.47	69.25	65.63	52.2	27.98	85.82	27.7	40.3	52.93	51.03
Yuan LM-13B	56.88	59.375	86.14	50	47.533	88.125	37.87	46.875	57.01	38.99
Yuan PLM-13B	55.83	56.875	85.63	48.13	46.57	88.125	38.82	48.125	57.468	32.7

表1(a) 采用零样本学习在ZeroCLUE任务上的表现

	Scores	Bustm	Chid	Csl	Csldcp	Eprstmt	Iflytek	Ocnli	Tnews	Wsc
Human	82.48	88	87.1	84	68	90	66	90.3	71	98
SOTA	70.16	76.7	69.45	76.57	59.55	88.05	45.77	72.02	73.87	73.1
Yuan LM-13B	69.14	81.25	72.27	81.25	60.9	90.625	54.87	46.25	52.45	82.39
Yuan PLM-13B	72.66	83.75	76.24	83.75	66.44	90.625	57.78	58.125	56.74	80.5

表1(b) 采用微调在FewCLUE上的表现

表1(a)和(b)表明LM和PLM在Zero-Shot和Few-Shot上都具有优异的表现能力。LM和PLM的零样本平均得分都优于已往的最优结果。在Csldcp、Tnews和Iflytek任务上，模型的得分大大超过了以往零样本学习的最优结果。模型在Ocnli上也取得了不错的成绩，比以往零样本学习的最优结果高出6-8个点。我们的监督微调方法与GPT的设计一致。　　LM和PLM的平均分数与以往最优分数相当，如表1(b)所示。与小样本学习结果相比，微调对Bustm、Csl和Wsc有很大的改进。但是，对于在零样本学习上表现出色的Chid、Eprsmt、Tnews和Ocnli，微调贡献很小甚至会有负面影响。

比较 LM和 PLM的结果，我们注意到 LM在 Zero-Shot和 Few-Shot上表现更好，而 PLM在微调方面表现出色。微调通常会在大多数NLU的任务中带来更好的准确性，这与我们一开始选择模型结构的初衷相合。然而，当模型参数量从百亿扩大到千亿规模，比如对于我们的　　“源1.0”模型，微调会消耗大量的计算资源，这是不经济的。所以最终，我们选择 LM作为“源1.0”的基础架构。

2. “源1.0”的文本生成效果

“源 1.0”更加出色的能力是体现在创作上（NLG）。为了评价模型生成文本的效果，我们任意选择了“源1.0”生成的24个文本，包括　4副对联、5首中文传统和现代诗歌、5篇新闻文章、5个故事和5段对话。对联、诗歌和对话的创作可以看作是短文本任务（~10-20个标记），而新闻和故事生成可以看作是长文本任务（~300个标记）。与之对比的人工写的文章来自名家所作的诗歌、经典小说、搜狐新闻的新闻文章和LCCC-large数据集中的对话。参与者被要求选择文章是“由人类撰写”还是“由模型撰写”，我们收集了83份有效问卷。根据我们的采访，大多数受访者会倾向于选择 “更好的”那一篇是由人类创作的，而“稍差”的那一篇是模型生成的。

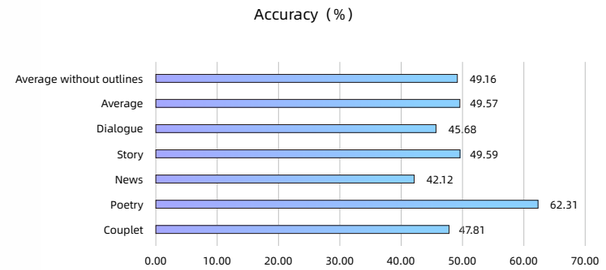


图2 人类检测由“源 1.0”模型创建的文章的准确性

图2展示了这次实验的结果。“源 1.0”创作的文章有 49.16%的概率可以被正确识别为“由模型撰写”，这意味着参与者难以区分人工撰写和模型生成的文章，尤其是现代诗和文章。新闻（42.12%）和故事（49.15%）的生成结果让我们可以相信模型出色的长文本生成能力。在我们的参与者看来，一些模型撰写的文章甚至比部分出自大师之手的文章还要好。对联和诗歌的生成表明，尽管我们的预训练语料中没有加强古汉语，“源1.0”也能够创建具有一定形式的古汉语文本，如图2所示。“源1.0”也可以进行符合人类期望的对话（45.68%）。“源1.0”目前能够生成多种高质量的文本，如新闻稿件、故事续写等。对于这些类别的任务，所产出的文章与人类创作的内容相差无，甚至达到了以假乱真的程度。

在表2中展示了一些“源 1.0”对不同文体的驾驭能力，读者可以稍加体会。

上联	“源1.0”创作的下联
雨 落日 鸡冠花 江山永固 四时多吉庆 与松竹梅交友 春回大地风光好 雪飘故里人财两旺 爆竹两三声人间换岁 绿树条条染绿春光一片 身比闲云，月影溪光堪证性	晴 清风 狗尾草 日月同辉 八方尽亨通 和鹿鹤龟同乐 福满人间气象新 梅开江南财丁双全 梅花四五点春信报时 青山点点妆成晚景千重 心如流水，诗情竹韵更传神

表2(a) 对联生成效果

风格	输入例子+要求	输出
李白	三峰却立如欲摧，翠崖丹谷高掌开。 白帝金精运元气，石作莲花云作台。 云台阆道连窈冥，中有不死丹丘生。 以饮茶为题写一首诗：	茶经通四海，仙茗播诸天。 灵芽生石上，芳气袭人烟。
杜甫	麟角凤觜世莫识，煎胶续弦奇自见。 尚看王生抱此怀，在于甫也何由羡？ 且过王生慰畴昔，素知贱子甘贫贱。 酷见冻馁不足耻，多病沈年苦无健。 以送别为题写一首诗：	送君别泪不须多， 此去应知重九过。 况是重阳无雨色， 秋山红叶满长安。
诗经	昔我往矣，杨柳依依。 今我来思，雨雪霏霏。 行道迟迟，载渴载饥。 我心伤悲，莫知我哀！ 以战争为题写一首诗：	我欲东归，害我者谁？ 我欲西归，谁当为守？ 我欲南归，谁与偕逝？ 我欲北归，谁与更岁？ 我欲与归，道里辽远，山川悠长。 我欲与归，而无与归。

表2(b) 诗歌生成效果

原文来源	输入原文	“源1.0”生成文本
老舍 《四世同堂》	祁老太爷什么也不怕，只怕庆不了八十大寿。 在他的壮年，他亲眼看见八国联军怎样攻进北京城。	他的亲家，也就是他的亲家母，给吓得从后门逃出城去，半路上就死了。他自己，他的亲儿子，也从城墙上跳下去。他的亲家母死的时候，曾留下话来，不准祁家的子孙再当兵。祁老人可是当了一辈子的兵，而且由绿林而变为“红”人。祁老人的大儿子，就是祁瑞宣的父亲。他在光绪年间入了伍。在光绪和慈禧相继死去以后，他参加了义和团。义和团失败以后，他被洋人捉住，关在了东交民巷的德国使馆里。祁瑞宣的母亲，在祁老人入狱以后，带着瑞宣和瑞丰，逃到了娘家。不久，祁瑞丰也入了伍。
鲁迅 《藤野先生》	中国留学生会馆的门房里有几本书买，有时还值得去一转；倘在上午，里面的几间洋房里倒也还可以坐坐的。	有所谓三文治者，附以洋火腿鸡蛋而美。这里还有几瓶汽水，但我们在旱冰场上已经喝够了。只记得每瓶要三元六角，比起二元六角一瓶的，大约要贵一倍了。然而那凉粉却颇凉，小冰箱时常是空着的。在这里，外国人自然也是很多的。但留学生有他们一种好处，便是很少在街上走；中国人的交涉，往往是口头上的，尽有被外国人窃听去的危险，而他们是实心实意的，来得去得，毫不费工夫。前几天听到一个在工学院正教授做事的亲戚说，正教授跑掉了十个了。

表2(c) 小说生成效果

问题	“源1.0”的回答
我长的帅吗？ 你干嘛呢？ 我喜欢你。 你是谁呀？ 你怎么了？ 我生气了。 你想干嘛？ 你长什么样子？ 好难过，我没有女朋友。 我难过你知道吗？	你是个大帅哥。 没干嘛，就是在想，你是不是真的喜欢我…… 我也是。 我是一个被你遗弃在这里的可怜人。 你说，你是不是有什么事瞒着我？ 我知道，我知道，你别生气，我错了。 只是想看看你。 我不告诉你！ 我也没有男朋友。 你的心，是我的海角和天涯，我不能去得更远。

表2(d) 对话生成效果

从表2(b)(c)中，我们可以看到“源 1.0”的学习和模仿能力。在生成诗歌时，如果我们在输入中给出某种风格的例子，模型就会倾向于生成相似风格的诗歌。在小说续写中，模型也会依据输入文本的风格，继续文章的写作。

除了基本的创作能力以外，“源 1.0”其实还有一些有趣的能力，比如，模型可以学习一些原本不存在的词的用法，也能够具备一定的推理能力。比如我们在输入中给出了这个词的定义和例句，模型将用给定的信息编写一个新句子。这个不存在词包括名词和形容词。表3(a)显示了模型在One-shot条件下的生成结果。在所有情况下，模型都对我们给出的新词作出了近似正确的应用，这意味着我们的模型具有学习和模仿能力。这种能力在模型辅助科学文章写作时特别有效，因为对“源 1.0”来说学术文章中的大量定义可能是陌生的。表3(b)则展示了模型的逻辑推理能力，这些推理题对于人类来说尚有一定的挑战、需要相当程度的推理和计算，但是模型却可以快速给出答案，且正确率远高于随机。

输入语义	生成句子
“悬命”是拼命，接近全力，做出最大努力的意思。 他凡事必悬命，绝不打马虎眼。用“悬命”造句：	他这个人做事太悬命了，为了取得第一手材料，连饭也顾不上吃
“腕前”与本领的意思相同。例：“他只使出了了一小部分腕前，就把事情完成得如此出色”。用“腕前”造句：	这个人的腕前很大，他很能完成这项任务
“气持”是心情、精神状态的意思。例：“那种消极的气持又控制了他”。用“气持”写一句话：	他的气持，使我无法接近
“油断”意思是漫不经心，马虎，轻率。举例：“她虽然在戏中只扮演一个次要的角色，但她并没有半点油断，精彩的表演得到大家的称赞”。请用“油断”写一个句子：	他的学习成绩很好，因为他并没有半点油断
“胜手”意思是最得意拿手的，擅长的。比如，“别的饭食我不会做，只有做面条还胜手”。请用“胜手”一词造句：	我的作文水平在班里名列前茅，是同学们公认的胜手

表3(a) 模型对于新词的学习能力

输入	输出
一 场景：在一列车的某节车厢内，有四位乘客面对面坐在一起。 他们身穿不同颜色的大衣，具有不同的国籍，其中两人是靠窗坐， 另两个人是挨着过道坐。 已知： 1. 他们的座位分别是A、B、C、D，其中有一名身穿蓝色的大衣的 旅客是个国际间谍。 2. 英国旅客坐在B先生的左侧。 3. A先生穿褐色大衣。 4. 穿黑色大衣者坐在德国旅客右侧。 5. D先生的对面坐着美国旅客。 6. 俄国旅客身穿灰色大衣。 7. 英国旅客把头转向左边，望着窗外。 结论：穿着蓝色大衣的间谍是	结论：穿着蓝色大衣的间谍是英国人，穿褐色大衣的是德国人。
二 场景：安娜去某城市旅游，想找一家酒店住宿。 已知： 1. 她希望这家酒店是三星级标准、有24小时热水和免费上网。 2. 朋友给她介绍了四家酒店，其中只有一家同时具备三个条件。 3. 只有三家酒店是三星级，两家酒店有24小时热水供应， 一家酒店可以免费上网。 4. 每家酒店至少具备其中一项条件。 5. A和B酒店的壬水供应时间一样。 6. B和C酒店的级别一样。 7. C和D的酒店级别不同。 结论：安娜最后选择的酒店是	结论：安娜最后选择的酒店是C酒店。 这道题的答案就是C，因为只有C酒店符合三个条件， 而其他酒店都只有一个条件满足， 所以C酒店是唯一符合条件的酒店。
三 场景：有四个好友——云帆、玉明、泽宁、文豪， 他们各自姓骆、高、陈、马。 已知： 1. 云帆的姓是“高”或“陈”的其中一个。 2. 玉明的姓是“高”或“骆”的其中一个。 3. 泽宁的姓是“陈”或“骆”的其中一个。 4. 姓“高”的是云帆或者文豪其中的一个。 结论：泽宁的姓是	结论：泽宁的姓是陈。

表3(b) 模型的逻辑推理能力

关于“源1.0”的更多信息，大家可以参照浪潮发布在arxiv上的论文：https://arxiv.org/abs/2110.04725。

中文巨量模型“源1.0”：语料质量清洗与数据分析方法

2021年，浪潮发布了中文巨量模型“源1.0”，其参数规模为2457亿，训练采用的中文数据集达5TB，超越美国OpenAI组织研发的GPT-3模型。“源1.0”在语言智能方面表现优异，获得中文语言理解评测基准CLUE榜单的零样本学习（zero-shot）和小样本学习（few-shot）两类总榜冠军。测试结果显示，人群能够准确分辨人与“源1.0”作品差别的成功率低于50%。

“源1.0”训练采用的5TB高质量中文数据集是从约860TB的互联网数据中获取的，本文将介绍这一过程使用的方法和取得的效果。

1. 语料质量清洗的必要性

大规模、高质量的预训练语料可以让模型学习更多的知识表达，更好地理解词的各种表征含义，从而更加智能。我们期望预训练所使用的数据足够多，并像百科词条一样语义通顺，且文本中包含一定的知识。但是事实上，百科词条极其有限，真正的大规模语料存在于各类互联网网页中。

通过网络爬虫可获得大量互联网语料，但是质量参差不齐。如表1所示的常见的低质量互联网语料，只使用关键词/规则过滤方法无法有效去除。

序号	原标签	原标签	原标签
1	存在目录结构/ 网站导航	新浪微博| 在线咨询|	每个句子token数量较少
2	存在广告	・广东名匠装饰27周年感恩庆典..... ・足球五大联赛竞猜官网注册（baber2.com）是亚洲优质游戏品牌.....	产品、公司广告，重复性较大
3	中英文混杂	Copyright © 2006-2020 ningansj.jfbxg.com 宁安15.2钢绞线公司	中文、英文混排，且语义并不通顺
4	条目重复	爱家，爱生活 名匠装饰，用心实现你的理想家 任何媒体、网站或个人未经本网书面授权不得转载、链接、转贴或以其他方式使用；	频繁出现的广告语、网站声明等， 重复出现概率较高
5	存在部分 纯标记记录	2021-05-16 06:59 点击：1576 评论：	只有少数评论类语料有该记录， 可用性不高
6	包含网址的条目	玩式的算圣伤www.0126.com和上的属性，	包含网址的条目是广告/无意义记录
7	语义混乱	忍旅续费持稳的商定业手我们率保。 安全协助行通出地交用户有序，奇旅奇打期间码在腾讯疫情健康。	中文语义不通
8	语义不完整	据了解，福原爱原本打算今年4月回台湾看望孩子， 恰好儿子的生日也在4月...	语义表述不完整，多以省略号为结束
9	短文本	2．引起强烈反响。	句子长度太短
10	带有特殊 符号内容	【华石涂装】/文章标题：【征求意见】 依】【靠】【内】【容】【价】【值】	目前看方括号中的内容都可以删除
11	存在电话号码	致电0750-3839929或18922043237	属于广告中内容
12	时间戳	2021-05-16 06:43:05	多出现于评论处，使用聚类可以筛选出来
13	重复无意义内容	12分钟电竞足球比分|热竞技-电竞外围下注|雷火电竞官网app在哪下| 雷火电竞下载官网入口|电竞大师软件下载|	多为网站标题、广告

表1 常见低质量互联网语料

2. 语料质量清洗

互联网语料基本上可以分为3大类：高质量语料（语句通顺且包含一定知识）、低质量语料（语句不通顺）、广告语料（语句通顺但重复率过高，如广告、网站说明、免责声明等）。为了给“源 1.0”模型提供高质量的预训练数据集，浪潮使用Bert训练了一个语料质量三分类模型。整个方案包括数据采样、语料标注、模型训练和效果评估四部分。

2.1 数据采样

在“源1.0”训练时，由于爬取的互联网语料非常庞大，即便经过敏感信息过滤、文章去重，数据集仍在TB级别。如此大规模的数据集难以直接进行处理、分析，需要进行采样。

将经过粗略过滤之后的数据，以文章为单位进行采样，再将所有采样数据分别写入到2个文件中，其中一个用来标注训练集语料（构建训练集，称之为“训练集”），另外一个则用来验证分类模型的效果（称之为“测试集”）。另外，为了使采样数据具有充分的代表性，在整个数据集内采用均匀采样。

2.2 语料标注

数据采样后，使用循环迭代标注方法以提高标注速度，如图1所示：

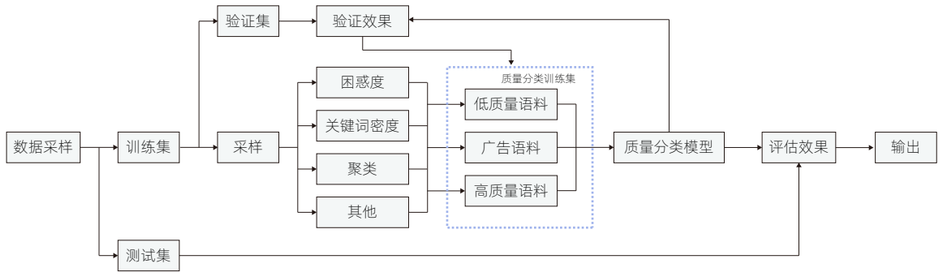


图1 获取质量分类模型策略图

如图所示，先使用“训练集”中的部分数据自动构建训练语料，然后使用分类模型进行训练，最后在剩余采样数据（“验证集”）上进行评估，并将得分较高的低质量语料和广告语料加入到标注数据中，如此循环多次。

另外，需要注意的是，互联网语料中大部分段落都比较短，多为单句，为方便叙述，下面统一将“段落”表述为“句子”。

采用PPL（困惑度）、关键词密度筛选、句向量聚类和人工标注等方法，从采样数据中筛选出低质量语料和广告语料，默认剩下的语料为高质量文本。下面简单介绍下这几个方法：

1. 针对语句不通顺的句子

互联网语料中含有较多的语义不通顺、有特殊符号的语料，为低质量数据。

本文使用困惑度（perplexity，简称PPL）来评估语句的合理性，其基本思想是，计算一句话出现的概率，并取-1/N 幂，其公式如下：

$$PPL = \mathbf{P}(w_1\ w_2...w_N)\ ^(-1/N) = \sqrt[N]{\frac{1}{\mathbf{P}(w_1\ w_2...w_N)}}$$

其中，$w_1w_2...w_N$代表一句话中N个token（token对应语言模型中的词典，对于中文而言，可能为一个字或者词）。

由上述公式可知，使用PPL排序，其分值越大，说明这句话出现的概率越低，表明质量越差。一句话的概率可根据如下公式计算：

$$\mathbf{P}(w_1\ w_2...w_N) = \mathbf{P}(w_2 \mid w_1)\mathbf{P}(w_3 \mid w_1\ w_2)...\mathbf{P}(w_N \mid w_1\ w_2...w_{N-1})$$

其中，给定前k个词，第k+1个词的概率可以通过GPT模型来计算。

计算每条语料的PPL值，从高到低排序，人工评估之后，确认前多少条数据为语句不通顺的语料。

值得注意的是，此方法需要大量的计算资源，且需要有训练好的中文GPT模型。

假设模型经过softmax之后的输出为对应的条件概率，则PPL的计算在某种程度上等价于loss（交叉熵损失函数）的计算。

$$\log(PPL) = (-1/N)\log(\mathbf{P}(w_1\ w_2...w_N))$$

$$= (-1/N)\log(\mathbf{P}(w_2 \mid w_1)\mathbf{P}(w_3 \mid w_1\ w_2)...\mathbf{P}(w_N \mid w_1\ w_2...w_{N-1}))$$

$$= (-1/N)\ (\log(\mathbf{P}(w_2 \mid w_1))+\log(\mathbf{P}(w_3 \mid w_1\ w_2))+...+\log(\mathbf{P}(w_N \mid w_1\ w_2...w_{N-1})))$$

$$= Loss$$

2. 针对知识表达不充分的句子

互联网语料中存在较多的短文本，如网站导航、目录结构等，为低质量数据。

我们使用了关键词密度筛选方法，根据关键词数量（最多30）和文本长度，希望筛选剔除掉关键词少于10但文本很长的词条，以及文本长度小于10的短句。

设关键词数量为N_k,文本长度为l,设计如下指标:

$$r = \frac{N_k}{m} \frac{\exp(N_k/m)}{\min(l,50)} \left(\frac{\max(l-10,0)}{l}\right)$$

其中，m为重要性提升阈值，当$N_k/m>1$时，r的增长速率更快（指数函数），从而拉开不同质量数据的得分。N_k的取值最大为30，经过实验，这里m的取值为9。最后min(l,50)是为了保证当词条长度超过关键词可表示的范围时，高质量词条的得分不因词条长度过长而下降。其中l-10的10为语义表示的最小字数阈值，当文本长度小于10时，r为0。对于长文本而言，该数值可以进一步增大。值得注意的是，r在二倍阈值时取得最大值。上式表明，当提取相同数量关键词时，用的字数越多，文本的得分越低。

使用小样本统计的指标分布图如下图所示。建议r的拆分阈值设定为0.015。

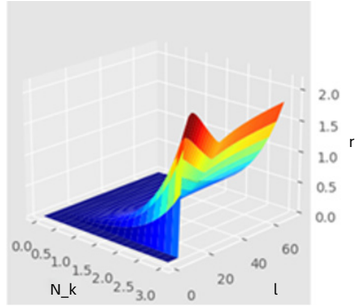


图3 关键词区分度曲面

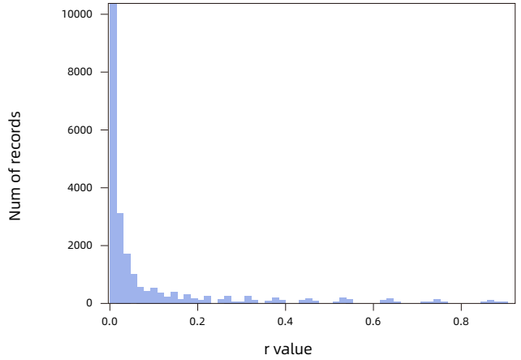


图4 采样数据集中指标值分布

另外，使用jieba分词的TextRank算法进行关键词的筛选。其主要思想是根据窗口大小，统计组合共现词和频率，使用频率代表共现权重，以此进行关键词抽取。

3. 针对频繁出现的句子

互联网语料中含有大量的网站说明、产品/公司广告介绍等，以及重复的广告、网站信息，会对语言模型产生误导，为广告数据。

我们使用句向量聚类的方法，对含义相近的句子进行聚类，如果超过指定阈值的句子数量达到一定比例，即认定为广告数据。其主要步骤有：

（1）获取句向量。将每条句子去重之后再进行编码，然后将句子输入到BERT中，使用第一个token（<cls>）作为句子向量，并将句子编码、句向量写入到磁盘中。

（2）根据句子向量，构建annoy高维向量索引；

（3）遍历所有句子，通过annoy索引，统计相似度超过给定阈值的句子数量。

（4）对每个句子的相似问题数量进行排序，相似问题数量越高，其为频繁出现低质量语料的概率更大。

注意，由于样本数量足够庞大，在得到句子向量表征之后，如果计算一个句子的语义相近句子，使用暴力检索的方式遍历所有语句计算相似度，是非常耗时的。因此，我们使用了Annoy算法进行索引构建以加快检索速度。Annoy是一种高维向量的近似索引算法，通过建立高维向量的二叉树表示，能够在较短的时间内找到任何查询点的最近点，在精度允许的条件下通过牺牲准确率来换取比暴力搜索快得多的搜索速度。

4. 其他

采用人工规则或人工标注方法。

2.3 训练质量分类模型

将经过PPL、关键词密度筛选得到的语料归为低质量语料，将经过语义聚类筛选得到的语料归为广告类语料，默认未被去除的为高质量语料，使用BERT进行有监督的三分类训练调优，并使用“验证集”进行评估。人工分析在“验证集”上的低质量文本中是否有被错分的，如果文本质量较高，则将其加入到高质量训练集中。并分析在“验证集”上的高质量文本中是否有被漏删的，将明显不合理的数据加入到低质量语料或广告语料中。重复几遍，不断扩充训练集、训练优化BERT模型，直至文本质量分类模型达到预期效果。

“达到预期”指的是大部分的高质量文本被保留，明显的低质量文本被去除。换句话说，我们并不期望模型能够十分准确地区分高质量语料和低质量语料，只期望模型能够将明显的低质量文本去除，对于剩余的高质量文本能有一个概率预估，文本质量越高的语句，得分越高，文本质量越低的语句，得分越低。

2.4 效果评估

与粗过滤相比，文本质量清洗后去除了约 2TB 的数据，其中50%被识别为广告。考虑到广告也可能包含完整的语义信息，手动评估它们以确定是否需要召回。由于数据集分散在36 台服务器上，为了避免人类偏见，在每台服务器上取样两组，每组由不同的评审员评估。部分统计结果见下表：

服务器 ID	样本 1	样本 2
cc01	1.25	0.60
cc02	2.17	1.85
cc03	2.88	3.43
cc04	6.59	6.48
cc05	1.02	1.49
cc06	1.08	1.13
cc07	0.64	0.72
cc08	0.70	0.86
cc09	0.65	0.60
cc10	0.94	1.04
cc11	0.70	0.81
cc12	0.40	0.39
平均值	**1.59**	**1.62**

表2 不同服务器上广告中高质量数据的百分比

从表2可以看出，样本1和样本2的高质量数据百分比是相似的，这说明评审人员在评估数据时具有高度一致性。广告中高质量数据的比例相当小，丢弃这部分数据是合理的。人工审查发现高质量语料的重复率约为2.4%，而广告语料重复率为 12.6%，由此可见，该分类模型对高质量数据进行了一定程度的重复数据消除和广告过滤。

3. 总结

为了给“源 1.0”提供高质量的预训练数据集，我们采用了一种海量互联网语料的质量清洗和分析方法。首先对海量文本数据进行采样，通过PPL、关键词密度筛选、聚类等方法自动筛选出低质量语料，然后使用分类模型在标注数据上进行训练，经过不断调优得到最终的文本质量分类模型。这一方法可以极大减少人工标注成本，从语义层面更加充分地去除各类低质量文本，有效提高预训练语言模型的理解能力。

关于“源1.0”的更多信息，大家可以参阅浪潮发布在arxiv上的论文：https://arxiv.org/abs/2110.04725。

浪潮“源”AI大模型如何求解数学应用题

“源1.0”大模型是浪潮信息发布的中文巨量模型，参数量高达2457亿，在中文语言能力理解和生成评测基准CUGE总榜中取得榜首，并获得语言理解（篇章级）、语言生成、对话交互、多语言、数学推理等5项评测最佳成绩。其中在数学推理评测中，源1.0大模型完成1000道小学数学应用题，以76.9的高分大幅领先。
数学对逻辑和推理能力有极强的要求，以往大模型在数学领域表现欠佳。源1.0为何能取得这么好的成绩？本文将介绍数学推理任务的背景、研究现状，以及源1.0在数学推理任务方面的解决方案和表现。

1. 数学单词问题的研究背景及意义

数学单词问题，即Math Word Problem（MWP），其主要目标是根据自然语言文字描述的内容解决相应的数学问题。也就是说，对于给定的数学问题，模型需要理解相关文字的数学含义，并推理出正确的表达式。

一个典型的MWP示例如下。

> 问题：“快车和慢车同时从相距450千米的两城相对开出，4.5小时后两车还相距90千米，快车和慢车的速度比为9:7，慢车每小时行多少千米？”
>
> 表达式：(450 - 90)/4.5*7/(9+7)
>
> 结果：35

不难发现，该题目除了要求模型能够理解基本的加减乘除法之外，还需要理解什么是比例问题。此外，若将问题中的“相对开出”改为“相反方向开出”，将会导致问题的数学逻辑大相径庭。如何让模型分辨出语言表达上的差异，并正确地推理出对应的表达式是MWP任务的基本要求。

需要注意的是，在上面的MWP中，表达式中所需的数字量均可以在问题中找到，但在某些情况下，表达式中所需要的数字量并不会全部包含在问题中。例如，在含有分数的MWP示例中（如下灰框中所示），需要根据题目中的数学逻辑，在表达式中额外添加相应的数字量“1”。同样的问题还常见于计算圆的周长或面积时，需要额外添加数字量“3.14”。

> 问题：“一根电线长80米，第一次截去的全长的2/5，第二次截去了余下的1/4，这根电线还剩多少米？”
>
> 表达式：80*(1-2/5-(1-2/5)*1/4)
>
> 结果：36

毫无疑问，MWP任务给模型的语言理解能力和数学推理能力都带来了极大的挑战，如何解决MWP任务也是NLP领域的研究热点之一。

2. 数字单词问题的研究现状

实际上，直到2016年MWP的任务精度仍然比较有限。关于MWP任务在2016年之前的研究在此不作细述，相关综述可参考论文：How well do Computers Solve Math Word Problems? Large-Scale Dataset Construction and Evaluation (Huang et al., ACL 2016)

近几年，借助DNN解决MWP任务的方法显著提升了MWP任务精度，这些方法大致可以分为以下三类：基于seq2seq模型、基于seq2tree模型和基于预训练模型。

2.1 基于seq2seq模型

该方法是由Wang Yan等学者[1]首次应用在MWP任务上，并在大规模多题型的数据集（Math23K）上取得了显著的效果（对于Math23K数据集将在后续内容中进行说明）。该方法本质上是采用Encoder-Decoder（enc-dec）结构直接完成了从“问题”到“表达式”的映射。值得一提的是，前述的Math23K数据集规模较大题型较多（约22000道），是目前MWP任务评测的benchmark。

此外，通过设计不同的Encoder和Decoder结构可以得到改进后的seq2seq方法。不过令人惊讶的是，Transformer结构的enc-dec并未在Math23K数据集上表现出明显的优势；而采用LSTM结构作为enc-dec的LSTMVAE方法表现最佳。

2.2 基于seq2tree模型

基于Seq2tree模型实际上是基于seq2seq模型的变种，简单来说，就是将number-mapping后的表达式转化为树结构作为模型训练的输出（如图1所示），由于父节点与子节点处的数学符号以及连接方式是固定的，这种方式能够有效地限制表达式的多样性。这里，表达式的多样性可以理解为针对同一个问题可以列出不同的表达式，例如$n_1+n_2-n_3$还可以写成$n_2+n_1-n_3$或者$n_1+(n_2-n_3)$。

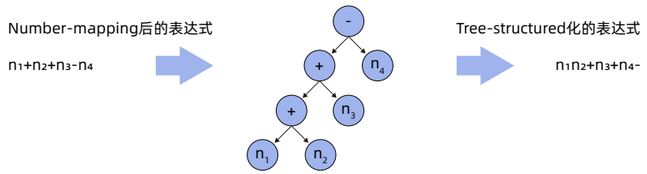


图1 树结构化的表达式生成示意[2]

在前述基础下，基于seq2tree模型的MWP任务解决方法应运而生，其核心思想是将原先的decoder被替换成了tree-based decoder。至此，MWP任务解决思路似乎主要集中在如何替换encoder和decoder问题上。例如，Wang Lei等学者又调整了encoder结构，提出了Graph2tree的方法并且在Math23K任务上精度高达75%。

2.3 基于预训练模型

Wang Lei等学者[3]发现BERTGen和RoBERTGen（Dec：BERT、RoBERT；Enc：Transformer）在Math23K数据集上表现较为优秀（76.9%）。此外，他们还验证了GPT-2模型在Math23K数据集上的表现（74.3%），结果稍逊于基于BERT模型的方法，这可能是GPT-2模型结构的原因（Decoder结构）。

2.4 其他MWP任务解决方法

根据前述方法，可以看到的是encoder采用BERT模型较好，decoder采用tree-based方式较好，若将两者结合形成BERT encoder + tree-based decoder[4]，其在Math23K数据集上的精度达到了惊人的84.4%，是目前Math23K任务的baseline。

此外，在众多MWP任务解决方法中Recall and learn方法[5]是十分值得一提的。该方法跳出了经典的enc-dec结构，通过模拟人脑在解决问题时的类比能力，推理出数学问题的表达式，最终该方法在Math23K任务上的精度能够达到82.3%。

3. “源1.0”大模型的MWP任务解决方案

需要指出的是，尽管构建单个技能模型在一定程度上能够较好地完成MWP任务，但现有技能模型绝大多数仍采用的是encoder-decoder结构，针对类似decoder结构下（如GPT-2）的模型数值推理能力的研究仍然较少。此外，从实现通用人工智能的目标来看，提升通用大模型的数值推理能力是十分必要的。

接下来，笔者将详细介绍浪潮信息的“源1.0”大模型（decoder结构）在Math23K任务上的相关工作，希望能够对提升通用大模型的数值推理能力有所启发。“源1.0”大模型在数学推理能力方面目前位列中文语言能力评测基准CUGE榜首。

3.1 目标导向的问答式Prompt设计

Math23K的标准数据样例为：

```
{
"text":"某班学生参加数学兴趣小组，其中，参加的男生是全班人数的20%，参加的女生是全班人数的(2/7)多2人，不参加的人数比全班人数的(3/5)少5人，全班有多少人？",
"segmented_text":"某班 学生 参加 数学 兴趣小组，其中，参加的男生是全班人数的20%，参加的女生是全班人数的(2/7)多2人，不参加的人数比全班人数的(3/5)少5人，全班有多少人？",
"equation":"x=(5-2)/(20%+(2/7)+(3/5)-1)",
"label":"35"
}
```

其中"text"和"equation"分别对应了任务的问题和表达式信息。在尝试过各种prompt后，最终确定的prompt设计如下。这种prompt设计将原本的问题拆分成了题干和待求解问题（“问：全班有多少人”）两个部分，这是由于“问：”后面的内容对表达式的生成十分关键。例如，“全班有多少人”和“全班女生有多少人”所对应的表达式是完全不同的。

{

某班学生参加数学兴趣小组，其中，参加的男生是全班人数的20%，参加的女生是全班人数的(2/7)多2人，不参加的人数比全班人数的(3/5)少5人，问：全班有多少人？答：

x=(5-2)/(20%+(2/7)+(3/5)-1)

}

3.2 相似启发式数据增强方法

Math23K数据集的题型虽然较为丰富，但题型分布并不均匀。例如，涉及图形周长、面积和体积类的问题显然比其他题目类型要少，为保证模型在各类数学题型上均有较好的表现，有必要将该类型的题目扩充。

本文采用了Ape210K数据集[6]对Math23K训练集进行扩充，Ape210K数据集是另一种较为常用的中文应用数学题集，其题型更为丰富且题量更大（训练集约20万道题）。然而，为保证模型在Math23K测试集上有良好的表现，并不能简单地将Math23K和Ape210K数据集混合在一起。为保证数据增强的有效性，本文提出了一种相似启发式数据增强方法（如图2所示）。

该方法针对Math23K训练集中的每一道题，首先判断是否属于图形周长、面积和体积类题目。若属于，则top-K取值为2，同时通过相似题检索从Ape210K中召回对应的相似题；若不属于，则top-K取值为1，同样进行相似题检索。最后，将找到的相似题添加至Math23K训练集中，数据增强后的训练集约包含42000道题。

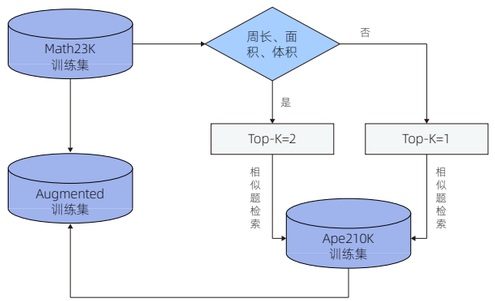


图2 相似启发式数据增强方法

3.3 Reset-position-id与reset-attention-mask设计

输入到模型的一个batch中通常包含多道应用题，且会出现截断等问题。为避免不同题目和表达式之间相互影响，对模型进行reset-position-id和reset-attention-mask处理。图3示意了reset前后的对比，采用了[eod]对不同题目之间做切割，在reset-pos-id之前，其位置编码按照从左到右的顺序排列；reset-pos-id之后，位置编码按照单个题目进行顺序排列。类似的，在reset-attn-mask之前，掩码矩阵对应的是batch尺寸的下三角矩阵；reset-attn-mask后，原先的掩码矩阵被拆分成若干小的掩码矩阵，每个小掩码矩阵对应单个题目尺寸的下三角矩阵。

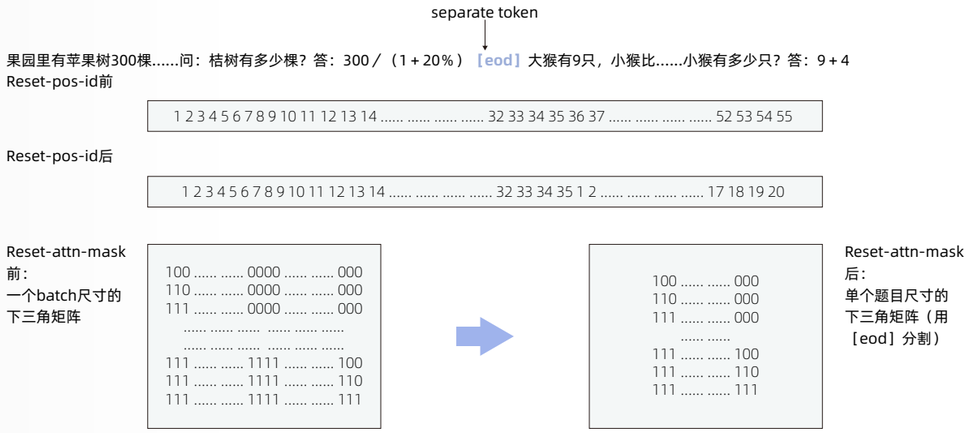


图3 reset-pos-id和reset-attn-mask前后对比（示意）

4. 训练参数及结果

训练过程的主要参数设置如下。

参数	数值
Seq-length	2048
Batch-size	256
Learning-rate	5e-6
Train-iters	400

表1 模型训练部分参数

在训练了400个iteration后，模型的loss收敛至0.39（图4）。

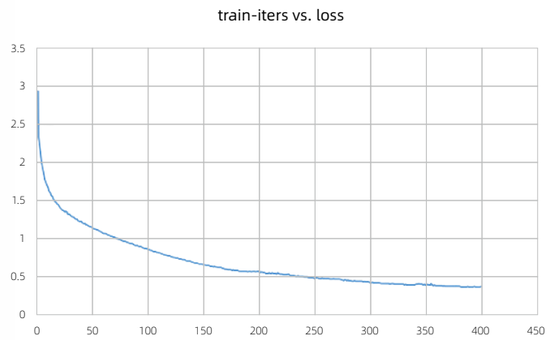


图4 模型loss曲线

之后，在Math23K测试集上对所提方法的精度进行了测试，并与现有相关方法的结果进行对比（表2）。不难看出，与BERT、GPT-2以及CPM-2模型相比，所提方法下的“源1.0”大模型在Math23K任务上的精度最高。

模型名称	Encoder-Decoder	Math23K精度（%）
BERTGen	是	76.6
RoBERTGen	是	76.9
CPM-2	是	69.4
GPT-2	Decoder结构	74.3
源1.0	Decoder结构	76.9

表2 源1.0模型与BERT、GPT等在Math23K测试集上的对比（相关结果见参考文献[4]）

5. 总结与展望

为提升decoder结构下的通用大模型在MWP任务上的精度，本文提出了一种目标导向的问答式prompt设计方法，该方法有利于引导模型建立问题与表达式之间的准确对应关系；同时提出了一种相似启发式数据增强方法，通过相似句召回的方式对数据集进行扩充，克服了原有数据集中题型分布不均匀的问题；此外，采用了重置位置编码和掩码矩阵的方法，解决了单个batch中的题目之间相互影响的问题。最后，在Math23K数据集上验证了所提方法，结果证明了“源1.0”模型有很强的数学推理能力。

针对MWP任务，“源1.0”模型后续将开展的工作包括：

1. 合理利用Number-mapping和tree结构的数据前处理，以及类似于recall and learn方法中的掩码矩阵设计，进一步提高“源1.0”在MWP任务上生成答案精度。

2. 虽然“源1.0”仅在Math23K任务上取得了较好的成绩，且目前还不能解决全部的MWP题型，但已经证明了“源1.0”模型具备了较强的数学推理能力。如何进一步挖掘“源1.0”在MWP任务上的潜力，以解决更为复杂的多元方程以及几何题型的问题，是我们后续准备继续深入研究的重要方向。

“源1.0”大模型技术白皮书

参考文献

[1] Yan Wang, Xiaojiang Liu, Shuming Shi (2017). Deep Neural Solver for Math Word Problems.

[2] Lei Wang, Yan Wang, Deng Cai, et al (2018). Translating a Math Word Problem to an Expression Tree.

[3] Yihuai Lan, Lei Wang, Qiyuan Zhang, et al (2021). MWPToolkit: An Open-Source Framework for Deep Learning-Based Math Word Problem Solvers

[4] Zhenwen Liang, Jipeng Zhang, Lei Wang, et al (2021). MWP-BERT: Numeracy-Augmented Pre-training for Math Word Problem Solving

[5] Shifeng Huang, Jiawei Wang, Jiao Xu, Da Cao, and Ming Yang. (2021). Recall and Learn: A Memory-augmented Solver for Math Word Problems.

[6] Wei Zhao, Mingyue Shang, Yang Liu, et al (2020). Ape210K: A Large-Scale and Template-Rich Dataset of Math Word Problems.