计算机研究与发展
Journal of Computer Research and Development
DOI: 10.7544/issn1000-1239.202330303
61(2): 351−361, 2024

大型语言模型:原理、实现与发展

舒文韬　李睿潇　孙天祥　黄萱菁　邱锡鹏
(复旦大学计算机科学技术学院　上海　200433)
(wtshu20@fudan.edu.cn)

Large Language Models: Principles, Implementation, and Progress

Shu Wentao, Li Ruixiao, Sun Tianxiang, Huang Xuanjing, and Qiu Xipeng
(*School of Computer Science*, *Fudan University*, *Shanghai* 200433)

Abstract　In recent years, the emergence and development of large language models (LLMs) have revolutionized the field of natural language processing and even artificial intelligence. With the increasing number of model parameters and training data, the perplexity of language models decreases in a predictable manner, which implies the improvement of performance on various natural language processing tasks. Therefore, scaling up language models has been a promising way to improve the system intelligence. In this survey, we first review the definition and scope of LLMs and provide a scale standard to distinguish "large" language models from the perspectives of performance and computing. Then, we review the development and representative work of LLMs in three dimensions: data, algorithm, and model architecture, showing how up-scaling in these dimensions drives the development of LLMs at different stages. Next, we discuss the emergent abilities of LLMs and possible interpretations behind them. We highlight three key emergent abilities, i.e., chain-of-thought prompting, in-context learning, and instruction-following, introducing their related advances and applications. Finally, we outline some potential directions and challenges of LLMs.

Key words　natural language processing; neural networks; large language models; pre-training; alignment

摘　要　近年来,大型语言模型的出现和发展对自然语言处理和人工智能领域产生了变革性影响.随着不断增大模型参数量和训练数据量,语言模型的文本建模困惑度以可预测的形式降低,在各类自然语言处理任务上的表现也持续提升.因此,增加语言模型的参数和数据规模成为提升系统智能水平富有前景的途径.首先回顾了大型语言模型的基本定义,从模型表现和算力需求的角度给出了"大型"语言模型的界定标准.其次,从数据、算法、模型3个维度梳理了大型语言模型的发展历程及规律,展示了不同阶段各个维度的规模化如何推动语言模型的发展.接着,考察了大型语言模型所表现出的涌现能力,介绍了思维链、情景学习和指令遵循等关键涌现能力的相关研究和应用现状.最后,展望了大型语言模型的未来发展和技术挑战.

关键词　自然语言处理;神经网络;大型语言模型;预训练;对齐

中图法分类号　TP391.1

语言模型(language model, LM),也称为统计语言模型(statistical language model),意在建模自然语言的概率分布,并估计任意语言序列的概率.语言模型可以充分利用互联网上大规模无标注语料作为训练数据,并广泛应用于机器翻译、语音识别等任务.随着深度学习算法和算力的迅速发展,研究人员发现,语言模型的表现可以随着模型参数量和训练数据的增长而持续提升[1],并对自然语言处理领域中的诸多

收稿日期:2023-04-07;**修回日期**:2023-09-27
通信作者:邱锡鹏(xpqiu@fudan.edu.cn)

任务,例如文本分类、命名实体识别、词性标注等有显著提升.因此,近年来语言模型,特别是大型语言模型(large language model, LLM)逐渐成为自然语言处理领域发展的主流,甚至展现出通向通用人工智能的潜能.

本文主要围绕大型语言模型的基本定义、发展路径、能力涌现和发展前景等4个方面展开讨论:

1) 基本定义.阐述了语言模型的基本定义和发展,从模型表现和算力需求的角度提供了"大型"语言模型的界定标准.

2) 发展路径.从数据、算法、模型3个维度回顾了语言模型的发展历程和重要工作,阐述了大型语言模型的规模定律,总结了近年来语言模型的发展规律.

3) 能力涌现.阐述了大型语言模型的能力涌现现象及可能的解释,重点介绍了情景学习、思维链和指令遵循3种关键涌现能力的有关研究和应用领域.

4) 发展前景.总结了大型语言模型在不同领域的技术发展方向和未来应用前景,阐述并分析了大型语言模型未来研究所面临的诸多技术挑战.

本文就大型语言模型的关键研究要素和主要技术问题进行了回顾和综述,以帮助读者深入了解这一领域的最新发展及未来展望.

1 大型语言模型的定义

1.1 语言模型

语言模型的目标在于建模自然语言的概率分布.具体地,语言模型可以通过多种方式实现,例如n-gram语言模型[2]将自然语言序列建模为马尔可夫过程(Markov process)从而简化自然语言的概率建模难度.目前被广泛使用的语言模型通常采用自左向右逐个预测单词的方式训练得到,即:

$$P(w_1,w_2,\cdots,w_T)=\prod_{t=1}^{T}P_\theta(w_t|w_0,w_1,\cdots,w_{t-1}),$$

其中w_0为起始符,w_T为结束符.在训练完成后,语言模型可以自回归(auto-regressive)地自左向右生成文本.

显然,由于自然语言的歧义性和句法的模糊性,通过上述方式建模自然语言的概率相当困难,需要参数化模型P_θ具有极大的容量.因此,目前的语言模型普遍采用Transformer模型架构[3],它通过注意力机制建模,输入文本中的长距离语义依赖,具有优秀的规模化能力和并行化计算能力[4].

1.2 大型语言模型的界定标准

虽然大型语言模型的概念已经深入人心,但目前尚无明确的界定标准来判断多大参数规模的语言模型才算作"大型"语言模型.一方面,"大型"语言模型应当具备某些"小型"语言模型不具备的能力;另一方面,大型语言模型的界定标准也随着算力的发展而变化,例如许多在今天看来规模不大的语言模型在五年前就可以算作大型语言模型.本节我们从模型表现和算力需求的角度讨论大型语言模型的界定标准.

1)模型表现.随着模型参数量的增长,研究人员发现许多过去性能处于随机水平的任务取得了显著提升.我们将这类随着模型参数规模增长而迅速习得的能力称为大型语言模型的涌现能力(emergent abilities)[5].在不同的任务上观测到涌现能力所需的参数量差异极大,目前仍然有大量困难任务未观测到模型性能的涌现.在目前受关注较多的大模型评测任务中,最小的涌现能力所需的参数量约为百亿左右,例如毒性分类能力的涌现所需的参数量约为71亿,3位数加减能力的涌现所需参数量约为130亿[5].因此,从模型表现的角度,把百亿参数规模作为大型语言模型的界定标准是较为合适的.

2)算力需求.训练大型语言模型的算力需求应当略微超过当前广泛可得的硬件条件.以当前较流行的单台配备了8张消费级显卡NVIDIA 3090 GPU的服务器测算,使用ZeRO模型并行计算方案[6]和Adam优化器[7],能够启动训练的模型规模约为百亿参数.因此,从算力需求的角度,超过百亿参数的语言模型可以被认为是常规计算资源难以完成训练的大型语言模型.

综上,不管从模型表现还是算力需求的角度,百亿参数量都是一个较为合适的大型语言模型的界定标准.值得注意的是,参数量并不是界定大型语言模型的唯一标准,模型架构、训练数据量、训练所需FLOPs等也是衡量大型语言模型的重要因素[8].例如,一个包含千亿参数但训练严重不充分的语言模型也难以被认为是一般意义上的大型语言模型.考虑到大规模语言模型训练成本高昂以及人们对语言模型规模定律(scaling law)[1]的认识,目前绝大多数大型语言模型都具备与其参数量相匹配的模型配置和训练数据,因而以参数量作为大型语言模型的界定标准是一种较为方便且合理的做法.

1.3 大型语言模型介绍

自GPT-3[9]问世以来,国内外多家机构加大对大

Table 1 Comparison of Existing Large Language Models

表 1 已有大型语言模型对比

模型	发布机构	所在国家	模型参数量	模态	最大序列长度	使用方式
GPT-3	OpenAI	美国	1 750 亿	语言	2 048	API
GPT-4	OpenAI	美国		语言、图像	32 000	API
Codex	OpenAI	美国	120 亿	代码		API
J1-Jumbo	AI21 Labs	美国	1 780 亿	语言	2 048	受限访问
J1-Grande	AI21 Labs	美国	170 亿	语言	2 048	受限访问
BLOOM	BigScience	法国	1 760 亿	语言	2 048	开源
GPT-NeoX	EleutherAI		200 亿	语言	2 048	开源
Anthropic-LM	Anthropic	美国	520 亿	语言	8 192	
Claude	Anthropic	美国		语言	100 000	受限访问
CodeGen	Salesforce	美国	160 亿	代码	2 048	开源
Turing-NLG	Microsoft	美国	170 亿	语言		
MT-NLG	Microsoft	美国	5 300 亿	语言	2 048	
OPT	Meta	美国	1 750 亿	语言	2 048	开源
LLaMA	Meta	美国	650 亿	语言	2 048	开源
T5	Google	美国	110 亿	语言	512	开源
UL2	Google	美国	200 亿	语言	512	开源
AlphaCode	Google	美国	410 亿	代码	768	
PaLM	Google	美国	5 400 亿	语言	2 048	API
LaMDA	Google	美国	1 370 亿	语言		
Chinchilla	Google	美国	700 亿	语言		
Gopher	Google	美国	2 800 亿	语言	2 048	
CPM-2	清华大学、智源	中国	1 980 亿	语言		开源
GLM-130B	清华大学、智谱	中国	1 300 亿	语言	2 048	开源
MOSS	复旦大学	中国	160 亿	语言	2 048	开源
InternLM	上海 AI LAB	中国	1 040 亿	语言	2 048	
ERNIE 3.0 Titan	百度	中国	2 600 亿	语言	512	受限访问
源 1.0	浪潮	中国	2 450 亿	语言	2 048	受限访问
盘古-α	华为	中国	2 000 亿	语言	1 024	
盘古-Σ	华为	中国	10 000 亿	语言	1 024	
WeLM	腾讯	中国	100 亿	语言		受限访问
M6	阿里巴巴	中国	1 000 亿	语言、图像		
M6-10T	阿里巴巴	中国	100 000 亿	语言、图像	512	
PLUG	阿里巴巴	中国	270 亿	语言		
Baichuan	百川智能	中国	70 亿	语言	4 096	开源
YaLM	Yandex	俄罗斯	1 000 亿	语言	2 048	开源

型语言模型的研发投入，近 3 年来涌现了一批具有竞争力的大型语言模型. 目前已有的大型语言模型总体呈现出以工业界投入为主，以英文为主，以及以闭源为主等特点. 表 1 中列举了当前常见大型语言模型的基本信息.

2 大型语言模型的发展路径

语言模型本是自然语言处理领域中的一个分支任务，近年来研究人员发现训练一个好的语言模型

对提升诸多自然语言处理任务, 例如情感分析、文本分类、序列标注等的性能具有显著帮助, 因而其重要性逐渐得到重视, 成为如今自然语言处理领域的发展主流.

历史上, 语言模型有许多变种, 例如将自然语言序列预测假设为马尔可夫过程(Markov process)的n-gram语言模型、最大熵(maximum entropy)语言模型等. 在本文中, 我们仅考虑当下流行的通过预测下一个单词训练得到的语言模型及其简单变体, 例如word2vec模型[10], 这类模型的训练任务可以概括为$P_\theta(w_t|context)$, 其中P_θ通常通过神经网络来建模, $context$可以是单词w_t之前的文本$w_0, w_1, \cdots, w_{t-1}$(如GPT模型[11]), 也可以是单词$w_t$的上下文$w_0, w_1, \cdots, w_{t-1}, w_{t+1}, \cdots, w_T$(如BERT模型[12]), 还可以是单词$w_t$的周围一定窗口范围的词$w_{t-k}, w_{t-k+1}, \cdots, w_{t-1}, w_{t+1}, \cdots, w_{t+k}$(如word2vec CBOW模型[10]).

图1展示了语言模型的主要发展路径: 2008年, Collobert等人[13]发现将语言模型作为辅助任务预先训练, 可以显著提升各个下游任务上的性能, 初步展示了语言模型的通用性; 2013年, Mikolov等人[10]在更大语料上进行语言模型预训练得到一组词向量, 接着通过迁移学习的手段, 以预训练得到的词向量作为初始化, 使用下游任务来训练任务特定模型; 2018年, Google公司的Devlin等人[12]将预训练参数从词向量扩增到整个模型, 同时采用Transformer架构作为骨干模型, 显著增大了模型容量, 在诸多自然语言处理任务上仅需少量微调即可取得很好的效果; 随后, 研究人员继续扩增模型参数规模和训练数据量, 同时采取一系列对齐算法使得语言模型具备更高的易用性、忠诚性、无害性, 在许多场景下展现出极强的通用能力, OpenAI于2022年底发布的ChatGPT以及2023年发布的GPT-4[14]是其中的代表. 纵观近十余年来语言模型的发展历程, 不难发现2个规律:

1)以语言模型及其变体为训练任务, 从多个维度实现规模化. 从2008年至今, 语言模型的训练任务变化很小, 而其训练数据逐渐从6亿单词增长到如今的超万亿单词, 算法从传统的多任务学习范式发展到更适合大规模预训练的迁移学习范式, 模型从容量较小的CNN/RNN模型发展为包含超过千亿参数的Transformer模型.

2)将更多模型参数和训练任务从下游转移到上游. 从模型参数的角度, 2013年以前的大多数模型要从头训练(training from scratch)所有参数; 2013~2018年主要基于预训练的词向量训练参数随机初始化的任务特定模型; 2018~2020年逐渐转向“预训练+微调”范式, 即使用预训练模型作为下游任务初始化, 仅需添加少量任务特定参数, 例如在预训练模型上添加一个随机初始化的线性分类器; 2020年前后, 基于提示(prompt)的方法得到了很大发展, 通常直接使用包括语言模型分类头(language modeling head)在内的整个预训练语言模型, 通过调整其输入内容来得到任务特定输出. 从训练任务的角度, 语言模型从与其他下游任务联合多任务训练逐渐发展成为独立的上游任务, 通过数据、模型、算法等多个维度的规模化逐渐降低对下游任务训练的需求, 近年来的大型语言模型通常在已有的上千个指令化自然语言处理任务(例如FLAN[15])上训练, 从而可以在未经下游任务训练的情况下很好地泛化到未见任务上.

下面我们分别从数据、算法、模型3个维度阐述语言模型的发展路径.

2.1 数 据

由于语言模型直接对文本的数据分布进行建模,

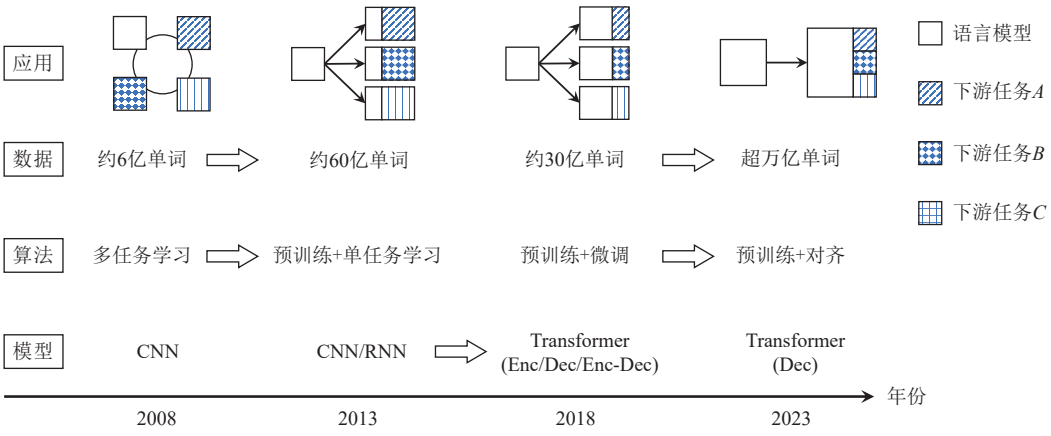


Fig. 1 Development path of language models

图1 语言模型发展路径

无需人工标注，因此可以充分利用互联网上海量的文本数据. 2008 年 Collobert 等人[13] 构造的语言模型训练在来自维基百科的约 6.3 亿单词上进行训练；2013 年 Mikolov 等人[10] 提出的 word2vec 在包含约 60 亿单词的 Google News 语料上进行词向量预训练；2018 年发布的 BERT 在约 8 亿个单词的 BooksCorpus 和约 25 亿个单词的英文维基百科，共约 33 亿个单词上进行预训练，虽然训练数据量较更早的 word2vec 有所下降，但由于其所采用的 Transformer 模型参数量大幅度增加，训练成本和效果均显著提升[12]；2023 年的最新语言模型，例如 GPT-4 和 LLaMA[16]，通常在超过万亿个语言单词上进行预训练.

随着预训练模型的规模化，维基百科、BooksCorpus 等高质量语料的规模和多样性逐渐无法满足训练需求，因而研究人员开始寻找更加广泛的数据来源，例如 CommonCrawl,Github,ArXiv 等，而这些数据质量和格式参差不齐，通常需要细粒度去重、低质量文本过滤、格式处理等繁杂的数据清洗步骤才能用于模型训练. 此外，互联网语料中还存在大量包含歧视性、刻板印象、事实性错误的文本，若用于训练将显著影响模型性能，导致模型产生带有毒性或幻觉的输出.

除预训练数据外，带标签的特定任务数据仍然具有极高的利用价值. 研究人员发现，为已有的大量自然语言处理任务编写描述指令并在大量此类指令化数据集上训练后，语言模型可以很好地根据输入的任务描述指令完成训练阶段未见过的任务. 为了增强语言模型的易用性、诚实性、安全性，通常还需要少量对齐数据进行训练，该部分数据通常包括人工编写的指令及其回复和对模型回复的偏好数据，前者与指令化任务数据类似，但通常具有更高的多样性，用于语言模型的监督微调；后者通常体现为多条模型回复的排序或两两比较结果，用于训练偏好模型（也称为反馈模型）. 此外，模型部署后收集的真实用户数据也常常作为对齐数据的一部分，用于训练偏好模型和调优语言模型. 通过对齐数据，语言模型可以与人类世界价值观进行对齐，显著降低模型毒性和幻觉问题. 最近一段时间，使用 ChatGPT 等能力较强的语言模型生成的合成数据因其获取成本低、数据质量高等优势得到了广泛应用，基于合成数据训练得到的语言模型取得了不俗的性能. 相较于人工标注的数据，合成数据的质量评估、潜在风险，以及更加多样的生成方法仍然需要大量研究工作.

2.2 算　法

在学习算法上，语言模型的发展大致经历了 4 个阶段：

1）多任务学习. 这一阶段的语言模型通常作为学习过程中一个可选的辅助任务，通过在少量无标签数据上训练语言模型任务来增益其他下游任务性能.

2）预训练+单任务学习. 随着语言模型任务的重要性受到越来越多的关注，研究人员开始在大规模无标注语料上预先训练一组词向量[10]，以此作为下游任务中模型词向量的初始化，使用任务特定数据训练模型参数. 其中词向量可以继续使用任务数据微调也可以保持不变而仅训练模型其余部分参数. 该阶段中单任务学习仍然是一个需要精心设计的环节，研究人员需要针对任务特性选择合适的模型结构和训练方法.

3）预训练+微调. 虽然通过语言模型任务预训练词向量的方式取得了巨大成功，但预训练词向量存在固有的缺陷：难以处理一词多义问题，例如“苹果”一词既可以指苹果这一水果，也可以指苹果公司. 一种卓有成效的解决方案就是将模型与词向量一同进行预训练，由此可以得到某个单词在特定语境下的表示，例如，通过预训练模型编码后，苹果一词在“苹果很好吃”和“苹果手机很好用”2 种不同语境下得到完全不同的表示. Peters 等人[17] 首先使用 LSTM 模型证明了这一做法的有效性，BERT，GPT 等模型则采用容量更大、更适合并行计算的 Transformer 模型. 经过大规模参数预训练之后，人们发现在下游任务上只需要对参数进行微调即可取得很好的效果.

4）预训练+对齐. 随着训练数据规模和模型参数规模的增长，研究人员发现保持模型参数不变而仅需调整模型输入的提示就可以得到不错的效果. 通过与人类对齐，包括使用自然语言指令化的任务数据训练和基于人类反馈学习，大型语言模型可以显著提高其易用性和安全性，用户通过简单的提示语即可得到期望的回复，实用性显著增强. 此外，相比过去主要基于监督学习方式，在对齐阶段还普遍引入了强化学习：首先训练反馈模型建模人类反馈数据，接着使用该反馈模型通过强化学习手段提升语言模型性能，使其更加符合人类偏好.

2.3 模　型

过去的语言模型训练常常基于卷积神经网络（convolutional neural network，CNN）、循环神经网络（recurrent neural network，RNN）及其变体，例如 LSTM、GRU[18] 等. 其中，CNN 具有优秀的并行计算能力，能够处理较长的输入序列，但其受限于感受野的大小，

难以处理自然语言中广泛存在的长距离依赖问题; RNN 及其变体将历史序列信息选择性地压缩进隐状态, 据此预测下一个单词, 这一结构上的先验非常符合自然语言序列的特点, 因而在诸多自然语言处理任务上具有广泛的应用. 然而, 由于 RNN 在训练过程中对输入序列中每个单词的处理都依赖其前序计算结果, 因而无法充分利用 GPU 的并行计算能力[19]. 2017 年, Vaswani 等人[3]提出了 Transformer 模型, 使用注意力机制对输入序列进行全局建模, 能够充分利用 GPU 的并行计算能力, 在机器翻译任务上取得了成功. 随后, Radford 等人[11]和 Devlin 等人[12]使用 Transformer 作为语言模型训练的骨干模型, 取得了突破性进展, 从此 Transformer 模型及其变体逐渐成为语言模型的主流.

2.4 规模定律

大型语言模型训练难度大、训练成本高, 如果能够根据已有小规模试验来提前预测为达到某种性能水平需要多少参数量、数据量、计算量, 则可以显著降低大模型训练试错成本. 这种模型性能与参数量、数据量、计算量等变量的经验关系就被称为“规模定律”.

OpenAI 的 Kaplan 等人[1]通过大量实验表明这样的规模定律是存在的, 即语言模型的性能(通过损失函数值衡量)是可以被参数量、数据量、计算量等变量预测的. 具体地, 他们发现语言模型的性能与 3 个因素均呈现幂律关系:

$$L(X)=\left(\frac{X_c}{X}\right)^{\alpha_X},$$

其中 L 为损失函数值, X 为参数量、数据量或计算量(FLOPs), X_c 和 α_X 为与参数量、数据量或计算量相关的常量. 当参数量和数据量按比例增长时, 语言模型的损失函数值是可以被预测的, 具体地, 在给定计算量情况下为达到语言模型最优性能, 模型参数量每增长 8 倍, 训练数据量应当增长 5 倍. 此外, 还发现: 相比训练数据和参数规模, 模型的宽度和深度等超参数对性能影响相对较小; 模型训练曲线同样遵循幂律变化, 可以通过早期训练曲线预测训练时间较长时模型的损失函数值, 且该幂律函数的参数与模型大小无关; 相较于小模型, 大模型需要更少的训练步数和更少的训练数据即可达到相同的性能水平. 这些经验规律大大降低了大型语言模型的试错成本, 对其后几年大型语言模型的发展起到了重要指导作用.

然而, DeepMind 的 Hoffmann 等人[20]在 2022 年通过训练参数量从 7 千万到 160 亿的超过 400 个语言模型, 给出了不同的规模定律: 给定计算量情况下为达到语言模型最优性能, 应当等比例增长训练数据量和模型参数量. 按照这一规模定律训练出的 Chinchilla 模型包含 700 亿个参数, 在包含约 1.4 万亿单词的语料上进行训练, 其在多任务理解评测基准 MMLU 上的性能超越了 2 800 亿个参数的 Gopher 和 5 300 亿个参数的 MT-NLG, 验证了其规模定律的有效性. 2023 年 Meta 推出的开源语言模型 LLaMA 采用了类似的训练配比, 使用 1.4 万亿个单词训练了 650 亿个参数, 取得了与 Chinchilla 可比的性能.

图 2 给出了当前常见的大型语言模型的参数量和训练计算量, 不难发现, 较近的语言模型, 如 Chinchilla 和 LLaMA 通常采用相对较大的训练数据和相对较小的参数规模, 这在下游微调和推理部署时具有显著的效率优势.

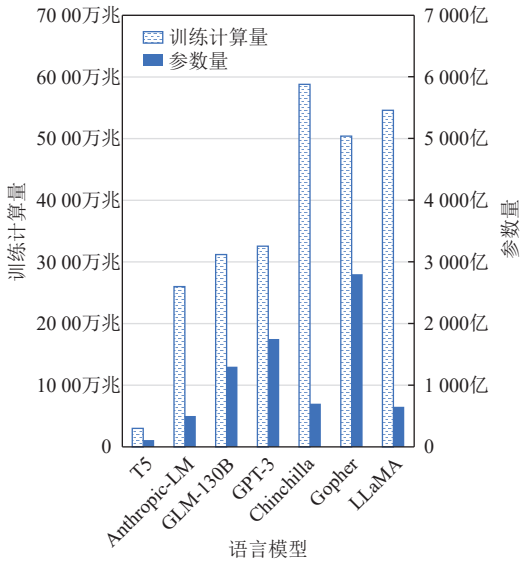


Fig. 2 Number of parameters and training FLOPs of common LLMs

图 2 常见大型语言模型的参数量和训练计算量

到目前为止, 规模定律仍然是一个非常重要且值得探索的方向, 特别是中文语言模型的规模定律尚未有公开研究. 此外, 已有的对规模定律的研究主要为通过大量实验得出的经验性规律, 而缺乏对其理论机理的解释.

3 大型语言模型的涌现能力

规模定律展示了语言模型的性能可以随着模型和数据规模可预测地增长, 然而, 当对应到具体任务时, 研究人员发现并非所有任务上的性能都是随着模型和数据规模平滑地、可预测地增长, 其中很多

任务上的表现是当模型和数据规模到达某个阈值后突然提升的. 这种较小规模模型不具备而大型语言模型具备的完成某些任务的能力就被称为“涌现能力”. 例如, 在少样本提示设定下进行三位数加减任务时, 当 GPT-3 达到 130 亿个参数、2×10^{22} 计算量时准确率出现迅速提升, 而在此之前模型准确率一直接近零. 值得注意的是, 即使同一任务的涌现阈值也不是放之四海皆准的, 而是与模型架构、训练方法等因素有关联, 例如三位数加减任务对于 LaMDA 则需要 680 亿个参数、10^{23} 计算量才能取得显著提升[5].

目前, 关于大型语言模型涌现能力的研究主要为实证研究, 其背后的理论机理仍然有待探索. 不过, 我们仍然可以从一些不同的视角来更好地理解大型语言模型的涌现能力. 例如, Wei 等人[5] 发现当把一些表现出涌现现象的任务的性能衡量指标从粗粒度指标(如准确率)替换为细粒度指标(如模型预测与真实标签的交叉熵)后, 这些任务上的表现曲线不再呈现出相变性, 而是可预测的平滑曲线. 然而, 值得注意的是, 并不是所有任务都能够找到使其性能曲线变得平滑的衡量指标. 此外, Michaud 等人[21] 提出了量子化模型(quantization model)来解释语言模型的规模定律和涌现现象, 他们假设模型的整体能力由许多量子化的能力组成, 由于数据分布常常呈现 Zipf 分布, 因此这些量子化能力的习得曲线自然地符合幂律分布. 在实验中他们观测到单个量子化能力的习得是涌现的, 即当模型参数规模达到某个阈值后在该能力相关单词的预测上损失值迅速下降; 而大多数单词的预测需要多个不同的量子化能力, 这些能力在不同的模型规模下涌现, 因此宏观表现为模型损失值随着规模增加而平滑地下降. 这也为理解某些任务性能的涌现提供了一个视角, 即解决某些较复杂任务所需的能力可以分解为多个子能力, 只有当所有子能力均被习得才能解决原任务, 因而在所有子能力均被习得后才能观测到任务性能的迅速提升.

相比于较小规模语言模型, 大型语言模型具备一些较为关键的涌现能力, 大大加强了其在真实场景下的可用性, 包括情景学习、思维链和指令学习.

3.1 情景学习

情景学习(in-context learning)[9] 是指将一部分样本及其标签作为示例拼接在待预测样本之前, 大型语言模型能够根据这小部分示例样本习得如何执行该任务. 具体地, 语言模型接受 $x_1, y_1, \cdots, x_k, y_k, x_{\text{query}}$ 为输入, 输出 x_{query} 对应的标签 y_{query}. 相较于传统的基于梯度更新的学习方式, 情景学习无需更新模型参数即可学习输入样本中的模式, 显著降低了学习成本, 使得“语言模型即服务(language-model-as-a-service, LMaaS)”[22] 变得可行.

尽管情景学习与一般的机器学习过程差别甚大, 例如情景学习中不存在显式的学习算法和参数更新, 但其输入输出形式又与机器学习相仿, 即可以认为输入中的 $\{x_1, y_1, \cdots, x_k, y_k\}$ 为训练集, 待预测的 x' 为测试样本. 目前已有一些工作试图建立情景学习与机器学习的联系. Akyürek 等人[23] 通过在线性回归任务上的实验发现, 基于 Transformer 的语言模型在进行情景学习时能够隐式地实现梯度下降, 即示例样本在输入到语言模型后在前馈传播过程中已经执行了与传统机器学习类似的学习过程, 从而能够习得训练集中的模式并给出测试样本的预测结果. 同时, Dai 等人[24] 通过分析 Transformer 中的注意力计算与梯度下降计算的对偶关系, 将语言模型解释为元优化器(meta optimizer), 并从多个角度展示了情景学习与传统语言模型微调的相似性. 基于该观察, 他们还设计了一种带有动量的注意力机制, 提升了情景学习能力, 这表明针对情景学习能力优化的模型架构研究仍有较大的探索空间. 值得注意的是, 尽管已有不少研究从理论和实证的层面展示了情景学习与梯度下降的联系, 但情景学习的工作机理仍不完全明确, 从优化的角度如何有效地提升语言模型情景学习的能力也是亟待探索的方向.

从应用的角度, 已有不少研究探索了情景学习的特性以及提升语言模型情景学习能力的方法. 例如, Min 等人[25] 发现情景学习的表现对特定上下文设置很敏感, 包括提示模板、上下文示例的选择与分布, 以及示例的顺序. 他们的实验表明, 示例样本对性能的影响主要来自 4 个方面: 输入-标签的配对格式、标签的分布、输入的分布以及输入-标签的映射关系. Wei 等人[26] 在 PaLM-540B 上得出了相反的结论, 即错误的映射关系会显著降低模型在二分类任务上的准确率, 这表明大型语言模型以一种异于小模型的方式进行情景学习. Zhao 等人[27] 发现, 多数标签和近因偏差也是导致情景学习结果出现偏见的重要因素: 语言模型更加倾向于与示例中占据多数的答案保持一致, 并且顺序越靠后的示例样本对预测结果的影响越大. 对此, 他们设计了一种校准方法用于消除示例标签及其位置分布可能导致的偏差.

目前, 情景学习已经成为大型语言模型能力的重要评测方法. 例如在被广泛用于大型语言模型评

测的基准数据集 MMLU 上，研究人员通常通过小样本情景学习的方式评测语言模型的表现. 因此，情景学习作为大型语言模型的基础能力之一，其理论机理和标准化应用方式是极为重要的研究方向.

3.2 思维链

思维链(chain-of-thought)[28]是提升大型语言模型推理能力的常见提示策略，它通过提示语言模型生成一系列中间推理步骤来显著提升模型在复杂推理任务上的表现. 其中，最直接的提示语言模型生成思维链的方法就是通过情景学习，即对少量样本$\{x_1, y_1, \cdots, x_k, y_k\}$手工编写其中间推理过程，形成$\{x_1, t_1, y_1, \cdots, x_k, t_k, y_k, x_{\text{query}}\}$作为语言模型的输入，使语言模型生成$x_{\text{query}}$对应的推理步骤和答案$\{t_{\text{query}}, y_{\text{query}}\}$. Kojima 等人[29]发现无需手工编写示例样本的推理步骤，仅需简单的提示词，例如“Let’s think step by step”即可使得语言模型生成中间推理过程及最终答案，这一提示策略称为“零样本思维链提示”. 通过思维链方法可以显著提升语言模型在常识问答、数学推理等任务上的性能. 随后，研究人员提出了一些基于思维链提示方法的改进策略，例如 Least-to-Most[30]、Self-consistency[31]、Diverse[32]等策略，通过这些策略可以进一步提升语言模型推理能力.

值得注意的是，在较小规模，如小于百亿参数语言模型上应用思维链提示策略反而会降低其在推理任务上的准确率，这是由于较小的语言模型通常会生成通顺但不合逻辑的思维链. 为了增强较小语言模型的思维链能力，一种被证明有效的做法是使用大型语言模型生成的思维链作为较小模型的训练信号[33]. 然而，这种方式通常会降低较小语言模型的通用能力.

CoT 为何能提示激发 LLM 的推理能力尚未得到解释. 有一种观点认为在预训练数据中加入代码可以帮助 LLM 具备 CoT 推理能力，但不少实验现象表明代码预训练和 CoT 推理能力并非完全挂钩. 事实上，BLOOM-176B [34]在预训练过程中加入了大量 GitHub 代码，但并未展现出 CoT 推理能力；与之对应的是没有经过大量代码预训练的 UnifiedQA [33,35]和微软 KOSMOS[36-37]，表现出了较好的 CoT 乃至多模态 CoT 推理能力.

3.3 指令遵循

指令遵循(instruction-following)能力是指语言模型根据用户输入的自然语言指令执行特定任务的能力. 相较情景学习需要通过少量示例样本提示语言模型执行特定任务，指令遵循的方式更为直接高效. 然而，指令遵循能力通常需要语言模型在指令数据集上进行训练而获得. 一种直接的构造指令数据集的手段是为已有的大量自然语言处理任务数据集编写自然语言指令，这种指令可以是对任务的描述，还可以包含少量示例样本. 研究人员发现，在大量指令化的自然语言处理任务数据集上训练后，语言模型可以根据用户输入的指令较好地完成未见任务.

然而，虽然已有的自然语言处理任务数据质量较高，但其多样性难以覆盖真实场景下用户的需求. 为此，InstructGPT[38]和 ChatGPT 采用人工标注的指令数据，具有更高的多样性且更加符合真实用户需求. 随着大型语言模型能力越来越强，研究人员发现可以通过编写少量种子指令(seed instruction)来提示语言模型生成大量高质量、多样化的指令数据集[39]. 近年来，使用较强的大型语言模型的输出来训练较小规模语言模型已经成为一种被广泛使用的方法，通过这种方式可以较容易地使得较小语言模型具备基本的指令遵循能力[40-41]. 然而，这种通过蒸馏获得的较小语言模型仍难以具备复杂指令遵循能力，且仍然存在严重的幻觉问题.

4 未来发展与挑战

以 ChatGPT、GPT-4 为代表的大型语言模型已经在社会各界引起了很大反响，其中 GPT-4 已经具备通用人工智能的雏形. 一方面，大型语言模型的强大能力向人们展现了其广阔的研究和应用空间；而另一方面，这类模型的快速发展也带来了许多挑战和应用风险.

虽然通过简单的规模化，大型语言模型已经取得了令人印象深刻的效果，但其仍有巨大的改进和扩展空间.

1) 高效大型语言模型. 当前大型语言模型主要采用 Transformer 架构，能够充分利用 GPU 的并行计算能力并取得不俗的性能表现. 但由于其计算和存储复杂度与输入文本长度呈平方关系，因此存在推理效率慢、难以处理长文本输入等缺陷. 对此，研究人员从稀疏注意力机制[42]、高效记忆模块[43]、新型架构[44]等角度探索计算高效的大型语言模型. 然而，已有高效模型架构的工作尚未在大规模参数量下进行验证，高效架构在大规模语言模型预训练下的表现及其改进是未来大型语言模型的重要发展方向.

2) 插件增强的语言模型. 集成功能插件已经成为大型语言模型快速获得新能力的重要手段[45]. 例如，

通过集成搜索引擎可以允许模型访问互联网实时信息，通过集成计算器可以帮助模型更精确地执行数学推理，通过集成专业数据库可以使得模型具备专业知识问答能力. 因此，如何通过训练或者提示的手段增强大型语言模型使用第三方插件甚至发明新插件的能力，如何使得模型能够根据插件反馈改进自身行为，最终解决较复杂推理问题成为饱受关注的研究方向. 此外，插件开发与模型能力的协同演化和生态建设也是值得重视、多方共建的重要议题.

3）实时交互学习. 目前语言模型仍以静态方式提供服务，即仅根据用户指令生成对应回复而无法实时动态更新自身知识，使得语言模型能够在与用户交互过程中完成实时学习，特别是能够根据用户输入的自然语言指令更新自身知识，是迈向通用人工智能的重要步骤. 目前元学习、记忆网络、模型编辑等领域的进展初步揭示了该方向的可行性，但面向大规模输入和参数的高效实时学习仍然是极重要与具有挑战性的研究方向.

4）语言模型驱动的具身智能. 具身智能与物理世界交互并在环境中完成任务的智能，意味着智能从被动观察学习到探索真实环境、影响真实环境的转变. 语言模型拥有相当的世界知识储备和一定的逻辑推理、因果建模和长期规划等高级认知功能，因而被广泛用于具身任务，并参与环境理解、任务理解、任务序列生成与分发等诸多环节. 通过多模态深度融合、强化逻辑推理与计划能力等手段，打造具备强大认知智能的具身系统正在成为大型语言模型和机器人领域的研究热点.

大型语言模型能力的迅速增长也对其落地应用带来了许多风险与挑战.

1）检测. 大型语言模型生成的文本高度复杂甚至相当精致，在很多场景下难以与人类创作的文本区分开. 这引发了对语言模型生成文本滥用的担忧，例如虚假文本生成在医学、法律、教育等领域的滥用可能导致巨大的隐患. 因而，语言模型生成文本的检测和监管成为亟待解决的问题，而现有的文本检测技术或模型水印等技术尚不能完全可靠地判断一段文本是否为模型生成. 从数据、训练、推理、产品等全链路进行设计和监管以提高模型生成文本的检测准确率，是确保大型语言模型不被滥用的重要条件.

2）安全性. 大型语言模型的训练数据大量来自互联网上未经标注的文本，因而不可避免地引入了有害、不实或歧视性内容. 此外，蓄意攻击者也可利用提示词注入等手段欺骗模型产生错误的输出，从而干扰系统运行、传播虚假信息或进行其他非法活动[46]. 尽管当前已经可以通过清洗训练数据、强化学习与社会价值观进行对齐等途径显著提升语言模型应用的安全性，但实际使用时安全性隐患仍层出不穷. 如何构造适合中文环境的安全性评估标准及其相应的训练数据仍然是中文语言模型大规模落地应用的重要挑战.

3）幻觉. 目前 ChatGPT 和 GPT-4 等高性能语言模型仍然存在较严重的幻觉问题，即经常生成包含事实性错误、似是而非的文本，这严重影响了其在部分专业领域应用的可靠性. 尽管通过接入搜索引擎、使用基于人类反馈的强化学习等手段可以显著降低模型生成的幻觉，但由于语言模型的黑箱性，有效识别模型的内部知识和能力边界仍旧是极具挑战性的未解难题.

总之，大型语言模型给自然语言处理乃至人工智能领域带来了巨大的范式变革，将原来按不同任务进行横向划分的领域设定转变为按流程阶段进行纵向划分的新型研究分工，并构建了以大型语言模型为中心的人工智能新生态.

作者贡献声明：舒文韬和李睿潇完成论文的撰写；孙天祥列举提纲，并校改论文；黄萱菁和邱锡鹏提出指导意见.

参 考 文 献

[1] Kaplan J, McCandlish S, Henighan T, et al. Scaling laws for neural language models[J]. arXiv preprint, arXiv: 2001. 08361, 2020

[2] Brown P F, Della P V J, Desouza P V, et al. Class-based n-gram models of natural language[J]. Computational Linguistics, 1992. 18(4): 467−480

[3] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C] // Proc of the 30th Annual Conf on Neural Information Processing Systems. New York: Curran Associates, 2017: 5990−6008

[4] Lin Tianyang, Wang Yuxin, Liu Xiangyang et al. A survey of Transformers[J]. AI Open, 2021 (3): 111−132

[5] Wei J, Tay Y, Bommasani R, et al. Emergent abilities of large language models[J]. arXiv preprint, arXiv: 2206. 07682, 2022

[6] Rajbhandari S, Rasley J, Ruwase O, et al. ZeRo: Memory optimizations toward training trillion parameter models[C]//Proc of: Int Conf for High Performance Computing, Networking, Storage and Analysis (SC20). Piscataway, NJ: IEEE, 2020: 1−16

[7] Kingma D P, Ba J. Adam: A method for stochastic optimization[J]. arXiv preprint, arXiv: 1412. 6980, 2014

[8] Hoffmann J, Borgeaud S, Mensch A, et al. Training compute-optimal large language models[J]. arXiv preprint, arXiv: 2203. 15556, 2022

[9] Brown T, Mann B, Ryder N, et al. Language models are few-shot learners[J]. Advances in Neural Information Processing Systems, 2020. 33, 1877−1901

[10] Mikolov T, Chen Kai, Corrado G, et al. Efficient estimation of word representations in vector space[J]. arXiv preprint, arXiv: 1301. 3781, 2013

[11] Radford A, Narasimhan K, Salimans T, et al. Improving language understanding by generative pre-training[DB/OL].https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf, 2018

[12] Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint, arXiv: 1810. 04805, 2018

[13] Collobert R, Weston J. A unified architecture for natural language processing: Deep neural networks with multitask learning[C]//Proc of the 25th Int Conf on Machine Learning. New York: ACM, 2008: 160−167

[14] OpenAI. GPT-4 technical report[J]. arXiv preprint, arXiv: 2303. 08774, 2023

[15] Chung H W, Hou Le, Longpre S, et al. Scaling instruction-finetuned language models[J]. arXiv preprint, arXiv: 2210. 11416, 2022

[16] Touvron H, Lavril T, Izacard G, et al. LLaMa: Open and efficient foundation language models[J]. arXiv preprint, arXiv: 2302. 13971, 2023

[17] Peters M, Neumann M, Iyyer M. Deep contextualized word representations. [C]// Proc of the 2018 Conf of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Stroudsburg, PA: ACL, 2018: 2227–2237

[18] Cho K, Van Merriënboer B, Gulcehre C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation[J]. arXiv preprint, arXiv: 1406. 1078, 2014

[19] Tang G, Müller M, Rios A, et al. Why self-attention? a targeted evaluation of neural machine translation architectures[J]. arXiv preprint, arXiv: 1808. 08946, 2018

[20] Hoffmann J, Borgeaud S, Mensch A, et al. Training compute-optimal large language models[J]. arXiv preprint, arXiv: 2203. 15556, 2022

[21] Michaud E J, Liu Ziming, Girit U, et al. The quantization model of neural scaling[J]. arXiv preprint, arXiv: 2303. 13506, 2023.

[22] Sun Tianxiang, Shao Yunfan, Qian Hong, et al. Black-box tuning for language-model-as-a-service[C]//Proc of Int Conf on Machine Learning. New York: PMLR, 2022: 20841−20855

[23] Akyürek E, Schuurmans D, Andreas J, et al. What learning algorithm is in-context learning? investigations with linear models[J]. arXiv preprint, arXiv: 2211. 15661, 2022

[24] Dai Damai, Sun Yutao, Dong Li, et al. Why can GPT learn in-context? language models secretly perform gradient descent as meta optimizers[J]. arXiv preprint, arXiv: 2212. 10559, 2022

[25] Min S, Lyu X, Holtzman A, et al. Rethinking the role of demonstrations: What makes in-context learning work?[J]. arXiv preprint, arXiv: 2202. 12837, 2022

[26] Wei J, Wei J, Tay Y, et al. Larger language models do in-context learning differently[J]. arXiv preprint, arXiv: 2303. 03846, 2023

[27] Zhao Z, Wallace E, Feng Si, et al. Calibrate before use: Improving few-shot performance of language models[C]// Proc of Int Conf on Machine Learning. New York: PMLR 2021: 12697−12706

[28] Wei J, Wang Xuezhi, Schuurmans D, et al. Chain-of-thought prompting elicits reasoning in large language models[J]. Advances in Neural Information Processing Systems, 2022. 35, 24824−24837

[29] Kojima T, Gu S S, Reid M, et al. Large language models are zero-shot reasoners[J]. Advances in neural information processing systems, 2022. 35, 22199−22213

[30] Zhou D, Schärli N, Hou L, et al. Least-to-Most prompting enables complex reasoning in large language models[J]. arXiv preprint, arXiv: 2205. 10625, 2022

[31] Wang Xuezhi, Wei J, Schuurmans D, et al. Self-consistency improves chain of thought reasoning in language models[J]. arXiv preprint, arXiv: 2203. 11171, 2022

[32] Zhang Zhuosheng, Zhang A, Li Mu, et al. Automatic chain of thought prompting in large language models[J]. arXiv preprint, arXiv: 2210. 03493, 2022

[33] Khashabi D, Kordi Y, Hajishirzi H. UnifiedQA-v2: Stronger generalization via broader cross-format training[J]. arXiv preprint, arXiv: 2202. 12359, 2022

[34] Scao T L, Fan A, Akiki C, et al. BLOOM: A 176B-parameter open-access multilingual language model[J]. arXiv preprint, arXiv: 2211. 05100, 2022

[35] Khashabi D, Min S, Khot T, et al. UnifiedQA: Crossing format boundaries with a single QA system[J]. arXiv preprint, arXiv: 2005. 00700, 2020

[36] Huang Shaohan, Dong Li, Wang Wenhui, et al. Language is not all you need: Aligning perception with language models[J]. arXiv preprint, arXiv: 2302. 14045, 2023

[37] Peng Zhiliang, Wang Wenhui, Dong Li, et al. Kosmos-2: Grounding multimodal large language models to the world[J]. arXiv preprint, arXiv: 2306. 14824, 2023

[38] Ouyang Long, Wu J, Jiang Xu, et al. Training language models to follow instructions with human feedback[J]. Advances in Neural Information Processing Systems, 2022. 35, 27730−27744

[39] Wang Yizhong, Kordi Y, Mishra S, et al. Self-instruct: Aligning language model with self generated instructions[J]. arXiv preprint, arXiv: 2212. 10560, 2022

[40] Bai Yuntao, Kadavath S, Kundu S, et al. Constitutional AI: Harmlessness from AI feedback[J]. arXiv preprint, arXiv: 2212. 08073, 2022

[41] Zheng Lianmin, Chiang W L, Sheng Ying, et al. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena[J]. arXiv preprint, arXiv: 2306. 05685, 2023

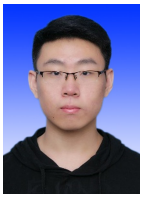
[42] Wang Sinong, Li B Z, Khabsa M, et al. Linformer: Self-attention with linear complexity[J]. arXiv preprint, arXiv: 2006. 04768, 2020

[43] Dao T, Fu D, Ermon S, et al. Flashattention: Fast and memory-efficient exact attention with io-awareness[J]. Advances in Neural

Information Processing Systems, 2022. 35, 16344−16359
[44] Peng Bo, Alcaide E, Anthony Q, et al. RWKV: Reinventing RNNs for the Transformer Era[J]. arXiv preprint, arXiv: 2305. 13048, 2023
[45] Schick T, Dwivedi-Yu J, Dessì R, et al. Toolformer: Language models can teach themselves to use tools[J]. arXiv preprint, arXiv: 2302. 04761, 2023
[46] Chen Yufei, Shen Chao, Wang Qian, et al. Security and privacy risks in artificial intelligence system[J]. Journal of Computer Research and Development, 2019. 56(10): 2135−2150(in Chinese)
(陈宇飞, 沈超, 王骞, 等. 人工智能系统安全与隐私风险 [J]. 计算机研究与发展, 2019, 56(10): 2135−2150)

Shu Wentao, born in 2002. Undergraduate. His main research interests include deep learning, natural language processing, and large language models.
舒文韬, 2002 年生. 本科生. 主要研究方向为深度学习、自然语言处理、大型语言模型.

Li Ruixiao, born in 2001. Undergraduate. His main research interests include deep learning and natural language processing.
李睿潇, 2001 年生. 本科生. 主要研究方向为深度学习、自然语言处理.



Sun Tianxiang, born in 1997. PhD candidate. His main research interests include deep learning and natural language processing.
孙天祥, 1997 年生. 博士研究生. 主要研究方向为深度学习、自然语言处理.

Huang Xuanjing, born in 1972. PhD, professor, PhD supervisor. Distinguished member of CCF. Her main research interests include natural language processing and information retrieval.
黄萱菁, 1972 年生. 博士, 教授, 博士生导师. CCF 杰出会员. 主要研究方向为自然语言处理、信息检索.

Qiu Xipeng, born in 1983. PhD, professor, PhD supervisor. Senior member of CCF. His main research interests include artifitial intelligence, natural language processing, and large language models.
邱锡鹏, 1983 年生. 博士, 教授, 博士生导师. CCF 高级会员. 主要研究方向为人工智能、自然语言处理、大型语言模型.