ChatGPT 调研报告

哈尔滨工业大学
自然语言处理研究所（HIT-NLP）

2023 年 3 月 6 日

序言

2022 年 11 月 30 日，OpenAI 推出全新的对话式通用人工智能工具——ChatGPT。ChatGPT 表现出了非常惊艳的语言理解、生成、知识推理能力，它可以很好地理解用户意图，做到有效的多轮沟通，并且回答内容完整、重点清晰、有概括、有逻辑、有条理。ChatGPT 上线后，5 天活跃用户数高达 100 万，2 个月活跃用户数已达 1 个亿，成为历史上增长最快的消费者应用程序。除了被广大用户追捧外，ChatGPT 还受到了各国政府、企业界、学术界的广泛关注，使人们看到了解决自然语言处理这一认知智能核心问题的一条可能的路径，并被认为向通用人工智能迈出了坚实的一步，将对搜索引擎构成巨大的挑战，甚至将取代很多人的工作，更将颠覆很多领域和行业。

哈工大自然语言处理研究所组织多位老师和同学撰写了本调研报告，从技术原理、应用场景、未来发展等方面对 ChatGPT 进行了尽量详尽的介绍及总结。

主要编撰人员

第一章由车万翔、杨沐昀、张伟男、赵妍妍、冯骁骋、孙承杰、李佳朋编写；第二章由张伟男、隋典伯、高翠芸、朱庆福、李明达、王雪松编写；第三章由刘铭、朱聪慧、汤步洲编写；第四章由徐永东、高翠芸、朱庆福编写；第五章由杨沐昀、张伟男、韩一、庄子彧编写；第六章由隋典伯、高翠芸编写；第七章由车万翔、刘铭编写。参与各章审校工作的还有：崔一鸣、徐志明等。

报告整体由车万翔统稿。

目录

第一章　ChatGPT 的背景与意义

本章首先介绍自然语言处理、大规模预训练语言模型以及 ChatGPT 技术的发展历程，接着就 ChatGPT 的技术优点和不足进行分析，然后讨论 ChatGPT 可能的应用前景，最后展望 ChatGPT 普及后可能带来的风险与挑战。

1.1　自然语言处理的发展历史

人类语言（又称自然语言）具有无处不在的歧义性、高度的抽象性、近乎无穷的语义组合性和持续的进化性，理解语言往往需要具有一定的知识和推理等认知能力，这些都为计算机处理自然语言带来了巨大的挑战，使其成为机器难以逾越的鸿沟。因此，自然语言处理被认为是目前制约人工智能取得更大突破和更广泛应用的瓶颈之一，又被誉为**“人工智能皇冠上的明珠”**。国务院 2017 年印发的《新一代人工智能发展规划》将知识计算与服务、跨媒体分析推理和自然语言处理作为新一代人工智能关键共性技术体系的重要组成部分。

自然语言处理自诞生起，经历了五次研究范式的转变（如图 1.1 所示）：由最开始基于小规模专家知识的方法，逐步转向基于机器学习的方法。机器学习方法也由早期基于浅层机器学习的模型变为了基于深度学习的模型。为了解决深度学习模型需要大量标注数据的问题，2018 年开始又全面转向基于大规模预训练语言模型的方法，其突出特点是充分利用**大模型**、**大数据**和**大计算**以求更好效果。

近期，ChatGPT 表现出了非常惊艳的语言理解、生成、知识推理能力，它可以极好地理解用户意图，真正做到多轮沟通，并且回答内容完整、重点清晰、有概括、有逻辑、有条理。ChatGPT 的成功表现，使人们看到了解决自然语言处理这一认知智能核心问题的一条可能的路径，并被认为向通用人工智能迈出了坚实的一步，将对搜索引擎构成巨大的挑战，甚至将取代很

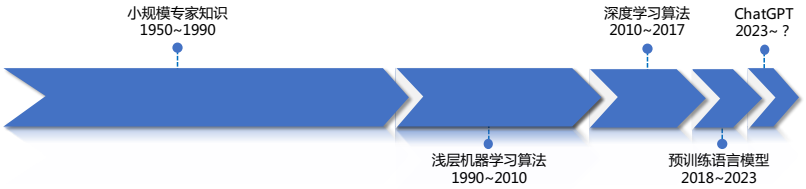

图 1.1: 自然语言处理研究范式的发展历程

多人的工作，更将颠覆很多领域和行业。

那么，ChatGPT 到底解决了什么本质科学问题，才能变得如此强大并受到广泛的关注呢？我们认为，**ChatGPT 是继数据库和搜索引擎之后的全新一代的“知识表示和调用方式”**。

知识在计算机内的表示是人工智能的核心问题。如表 1.1所示，早期，知识以结构化的方式存储在数据库中，人类需要掌握机器语言（如 SQL），才能调用这些知识；后来，随着互联网的诞生，更多文本、图片、视频等非结构化知识存储在互联网中，人类通过关键词的方式调用搜索引擎获取知识；现在，知识以参数的形式存储在大模型中（从 2018 年开始），ChatGPT 主要解决了用自然语言直接调用这些知识的问题，这也是人类获取知识最自然的方式。

表 1.1: 知识表示和调用方式的演进

知识表示方式	表示方式的精确度	知识调用方式	调用方式的自然度	研究领域	代表应用	代表公司
关系型数据库	高	SQL	低	数据库	DBMS	Oracle、Microsoft
互联网	中	Keywords	中	信息检索	搜索引擎	Google、Microsoft
大模型	低	自然语言	高	自然语言处理	ChatGPT	OpenAI、Microsoft、Google

另外，从自然语言处理技术发展阶段的角度看（如图 1.1 ），可以发现一个有趣的现象，即每一个技术阶段的发展时间，大概是上一个阶段的一半。小规模专家知识发展了 40 年，浅层机器学习是 20 年，之后深度学习大概 10 年，预训练语言模型发展的时间是 5 年，那么以 ChatGPT 为代表的技

术能持续多久呢？如果大胆预测，可能是 2 到 3 年，也就是到 2025 年大概又要更新换代了。

1.2 大规模预训练语言模型的技术发展历程

大规模预训练语言模型（简称大模型），又称大规模语言模型（Large Language Model，LLM），作为 ChatGPT 的知识表示及存储基础，对系统效果表现至关重要，接下来对大模型的技术发展历程加以简要介绍。

2018 年，OpenAI 提出了第一代 GPT（Generative Pretrained Transformer）模型[1]，将自然语言处理带入“预训练”时代。然而，GPT 模型并没有引起人们的关注，反倒是谷歌随即提出的 BERT（Bidirectional Encoder Representations from Transformers）模型[2] 产生了更大的轰动。不过，OpenAI 继续沿着初代 GPT 的技术思路，分别于 2019 年和 2020 年发布了 GPT-2[3] 和 GPT-3[4] 模型。

尤其是 GPT-3 模型，含有 1,750 亿超大规模参数。由于参数过大，不易进微调，因此提出“提示语”（Prompt）的概念，只要提供具体任务的提示语，即便不对模型进行调整也可完成该任务，如：输入“我太喜欢 ChatGPT 了，这句话的情感是 ___”，那么 GPT-3 就能够直接输出结果“褒义”。如果在输入中再给一个或几个示例，那么任务完成的效果会更好，这也被称为语境学习（In-context Learning）。更详细的技术细节推荐阅读相关的综述文章[5-8]。

不过，通过对 GPT-3 模型能力的仔细评估发现，大模型并不能真正克服深度学习模型鲁棒性差、可解释性弱、推理能力缺失的问题，在深层次语义理解和生成上与人类认知水平还相去甚远。直到 ChatGPT 的问世，才彻底改变了人们对于大模型的认知。

1.3 ChatGPT 技术发展历程

2022 年 11 月 30 日，OpenAI 推出全新的**对话式通用人工智能工具**——ChatGPT。据报道，在其推出短短几天内，注册用户超过 100 万，2 个月活跃用户数已达 1 个亿，引爆全网热议，成为历史上增长最快的消费者应用程序，掀起了人工智能领域的技术巨浪。

ChatGPT 之所以有这么多活跃用户，是因为它可以通过学习和理解人类语言，以对话的形式与人类进行交流，交互形式更为自然和精准，极大地改变了普通大众对于聊天机器人的认知，完成了从“人工智障”到“有趣”

的印象转变。除了聊天，ChatGPT 还能够根据用户提出的要求，进行机器翻译、文案撰写、代码撰写等工作。ChatGPT 吹响了大模型构建的号角，学界和企业界纷纷迅速跟进启动研制自己的大模型。

继 OpenAI 推出 ChatGPT 后，与之合作密切的微软迅速上线了基于 ChatGPT 的 New Bing，并计划将 ChatGPT 集成到 Office 办公套件中。谷歌也迅速行动推出了类似的 Bard 与之抗衡。除此之外，苹果、亚马逊、Meta（原 Facebook）等企业也均表示要积极布局 ChatGPT 类技术。国内也有多家企业和机构明确表态正在进行类 ChatGPT 模型研发。百度于 2023 年 3 月 16 日发布了基于文心大模型的文心一言系统，阿里巴巴表示其类 ChatGPT 产品正在研发之中，华为、腾讯表示其在大模型领域均已有相关的布局，网易表示其已经投入到类 ChatGPT 技术在教育场景的落地研发，京东表示将推出产业版 ChatGPT，科大讯飞表示将在数月后进行产品级发布，国内高校复旦大学则推出了类 ChatGPT 的 MOSS 模型。

除了国内外学界和企业界在迅速跟进以外，我国国家层面也对 ChatGPT 有所关注。2023 年 2 月 24 日，科技部部长王志刚表示：“ChatGPT 在自然语言理解、自然语言处理等方面有进步的地方，同时在算法、数据、算力上进行了有效结合。”科技部高新技术司司长陈家昌在回应 ChatGPT 相关提问时也表示，“ChatGPT 最近形成了一种现象级的应用，表现出很高的人机交互水平，表现出自然语言的大模型已经具备了面向通用人工智能的一些特征，在众多行业领域有着广泛的应用潜力。[1]”这标志着在未来，ChatGPT 相关技术有可能会成为国家战略支持的重点。

从技术角度讲，ChatGPT 是一个聚焦于对话生成的大语言模型，其能够根据用户的文本描述，结合历史对话，产生相应的智能回复。其中 GPT 是英文 Generative Pretrained Transformer 的缩写。GPT 通过学习大量网络已有文本数据（如 Wikipedia，Reddit 对话），获得了像人类一样流畅对话的能力。虽然 GPT 可以生成流畅的回复，但是有时候生成的回复并不符合人类的预期，OpenAI 认为符合人类预期的回复应该具有真实性、无害性和有用性。为了使生成的回复具有以上特征，OpenAI 在 2022 年初发表的工作“Training language models to follow instructions with human feedback”中提到引入人工反馈机制，并使用近端策略梯度算法（PPO）对大模型进行训练。这种基于人工反馈的训练模式能够很大程度上减小大模型生成回复与人类回复之间的偏差，也使得 ChatGPT 具有良好的表现。

[1]https://www.sohu.com/a/645545405_120109837

1.3.1 ChatGPT 的相关技术

接下来将简要介绍 ChatGPT 相关技术的发展历程。ChatGPT 核心技术主要包括具有良好的自然语言生成能力的大模型 GPT-3.5 以及训练这一模型的钥匙——基于人工反馈的强化学习（RLHF）。

GPT 家族是 OpenAI 公司推出的相关产品，这是一种生成式语言模型，可用于对话、问答、机器翻译、写代码等一系列自然语言任务。每一代 GPT 相较于上一代模型的参数量均呈现出爆炸式增长。OpenAI 在 2018 年 6 月发布的 GPT 包含 1.2 亿参数，在 2019 年 2 月发布的 GPT-2 包含 15 亿参数，在 2020 年 5 月发布的 GPT-3 包含 1750 亿参数。与相应参数量一同增长的还有公司逐年积淀下来的恐怖的数据量。可以说大规模的参数与海量的训练数据为 GPT 系列模型赋能，使其可以存储海量的知识、理解人类的自然语言并且有着良好的表达能力。

除了参数上的增长变化之外，GPT 模型家族的发展从 GPT-3 开始分成了两个技术路径并行发展[2]，一个路径是以 Codex 为代表的代码预训练技术，另一个路径是以 InstructGPT 为代表的文本指令（Instruction）训练技术。但这两个技术路径不是始终并行发展的，而是到了一定阶段后（具体时间不详）进入了融合式预训练的过程，并通过指令有监督微调（Supervised Fine-tuning）以及基于人类反馈的强化学习（Reinforcement Learning with Human Feedback, RLHF）等技术实现了以自然语言对话为接口的 ChatGPT 模型。

RLHF 这一概念最早是在 2008 年 TAMER：Training an Agent Manually via Evaluative Reinforcement[9] 一文中被提及的。在传统的强化学习框架下，代理（Agent）提供动作给环境，环境输出奖励和状态给代理，而在 TAMER 框架下，引入人类标注人员作为系统的额外奖励。该文章中指出引入人类进行评价的主要目的是加快模型收敛速度，降低训练成本，优化收敛方向。具体实现上，人类标注人员扮演用户和代理进行对话，产生对话样本并对回复进行排名打分，将更好的结果反馈给模型，让模型从人类评价奖励中学习策略，对模型进行持续迭代式微调。这一框架的提出成为后续基于 RLHF 相关工作的理论基础。

在 2017 年前后，深度强化学习（Deep Reinforcement Learning）逐渐发展并流行起来。MacGlashan et al.[10] 提出了一种 AC 算法（Actor-critic），并且将人工反馈（包括积极和消极）作为信号调节优势函数（Advantage

[2] https://openai.com/blog/

function)。Warnell et al.[11] 将 TAMER 框架与深度强化学习相结合，成功将 RLHF 引入深度强化学习领域。在这一阶段，RLHF 主要被应用于模拟器环境（例如游戏等）或者现实环境（例如机器人等）领域，而利用其对于语言模型进行训练并未受到重视。

在 2019 年以后，RLHF 与语言模型相结合的工作开始陆续出现，Ziegler et al.[12] 较早利用人工信号在四个具体任务上进行了微调并取得不错的效果。OpenAI 从 2020 年开始关注这一方向并陆续发表了一系列相关工作，如应用于文本摘要[13, 14]，利用 RLHF 训练一个可以进行网页导航的代理[15]等。后来，OpenAI 将 RLHF 与 GPT 相结合，提出了 InstructGPT 这一 ChatGPT 的前身[16]，主要是利用 GPT-3 进行对话生成，旨在改善模型生成的真实性、无害性和有用性。与此同时，作为缔造 AlphaGo 的公司，具有一干擅长强化学习算法工程师的 DeepMind 也关注到了这一方向，先后发表了 GopherCite[17] 和 Sparrow[18] 两个利用 RLHF 进行训练的语言模型，GopherCite 是在开放域问答领域的工作，Sparrow 是在对话领域的一篇工作。

2022 年 12 月，OpenAI 在诸多前人工作的积淀之下推出了 ChatGPT。ChatGPT 以 GPT-3.5 作为基座，依托其强大的生成能力，使用 RLHF 对其进行进一步训练，从而取得了惊艳四座的效果。

1.3.2 ChatGPT 技术发展脉络的总结

纵观 ChatGPT 的发展历程，不难发现其成功是循序渐进的，OpenAI 从 2020 年开始关注 RLHF 这一研究方向，并且开展了大量的研究工作，积攒了足够的强化学习在文本生成领域训练的经验。GPT 系列工作的研究则积累了海量的训练数据以及大语言模型训练经验，这两者的结合才产生了 ChatGPT。可以看出技术的发展并不是一蹴而就的，是大量工作的积淀量变引起质变。此外，将 RLHF 这一原本应用于模拟器环境和现实环境下的强化学习技术迁移到自然语言生成任务上是其技术突破的关键点之一。

纵观 AI 这几年的发展，已经逐渐呈现出不同技术相互融合的大趋势，比如将 Transformer 引入计算机视觉领域产生的 ViT[19]；将强化学习引入蛋白质结构预测的 AlphaFold 等。每个研究人员都有自己熟悉擅长的领域，而同时科学界也存在着大量需要 AI 赋能的亟待解决的关键问题，如何发现这些问题的痛点，设计合理的方法，利用自己研究领域优越的技术解决问题，是一个值得思考，也非常有意义的问题。

这是一个 AI 蓬勃发展的时代，计算机科学界每天都在产生着令人惊奇的发明创造，很多之前人们可望而不可及的问题都在或者正在被解决的路上。2022 年 2 月，DeepMind 发布可对托卡马克装置中等离子体进行磁控制的以帮助可控核聚变的人工智能，这项研究目前仍在进行。或许在未来的某一天，能源将不成为困扰我们的问题，环境污染将大大减少，星际远航将成为可能。希望每个研究人员都能在这样的时代中，找到适合自己的研究方向并且为科技进步添砖加瓦。

1.3.3 ChatGPT 的未来技术发展方向

虽然 ChatGPT 目前已经取得了非常喜人的成果，但是未来仍然有诸多可以研究的方向。

首先 OpenAI 的研究人员指出了 ChatGPT 现存的一些问题[3]：

1. ChatGPT 有时候会生成一些似是而非、毫无意义的答案，导致这个问题的原因有：强化学习训练过程中没有明确的正确答案；训练过程中一些谨慎的训练策略导致模型无法产生本应产生的正确回复；监督学习训练过程中错误的引导导致模型更倾向于生成标注人员所知道的内容而不是模型真实知道的。

2. ChatGPT 对于输入措辞比较敏感，例如：给定一个特定的问题，模型声称不知道答案，但只要稍微改变措辞就可以生成正确答案。

3. ChatGPT 生成的回复通常过于冗长，并且存在过度使用某些短语的问题，例如：反复申述是由 OpenAI 训练的语言模型。这样的问题主要来自于训练数据的偏差和过拟合问题。

4. 虽然 OpenAI 已经努力让模型拒绝不恰当和有害的请求，但是仍然无法避免对有害请求作出回复或对问题表现出偏见。

其次，ChatGPT 虽然很强大，但是其模型过于庞大使用成本过高，如何对模型进行瘦身也是一个未来的发展方向，目前主流的模型压缩方法有量化、剪枝、蒸馏和稀疏化等。量化是指降低模型参数的数值表示精度，比如从 FP32 降低到 FP16 或者 INT8。剪枝是指合理地利用策略删除神经网络中的部分参数，比如从单个权重到更高粒度组件如权重矩阵到通道，这种方

[3]https://openai.com/blog/chatgpt

法在视觉领域或其他较小语言模型中比较奏效。蒸馏是指利用一个较小的学生模型去学习较大的老师模型中的重要信息而摒弃一些冗余信息的方法。稀疏化将大量的冗余变量去除，简化模型的同时保留数据中最重要的信息。

此外，减少人类反馈信息的 RLAIF 也是最近被提出的一个全新的观点。2022 年 12 月 Anthropic 公司发表论文“Constitutional AI: Harmlessness from AI Feedback”[20]，该公司是 2020 年 OpenAI 副总裁离职后创立的，其公司始创团队中多有参与 GPT-3 以及 RLHF 相关研究的经历。该文章介绍了其最新推出的聊天机器人 Claude，与 ChatGPT 类似的是两者均利用强化学习对模型进行训练，而不同点则在于其排序过程使用模型进行数据标注而非人类，即训练一个模型学习人类对于无害性偏好的打分模式并代替人类对结果进行排序。

1.4 ChatGPT 的优势与劣势

1.4.1 ChatGPT 的优势

ChatGPT 作为开年爆款产品，自发布以来不足三个月，就以其能力的全面性、回答的准确性、生成的流畅性、丰富的可玩性俘获了数以亿计的用户，其整体能力之强大令人惊叹。下面我们将从以下三个角度分别阐述 ChatGPT 相较于不同产品和范式的优点。

1. 相较于普通聊天机器人： ChatGPT 的发布形式是一款聊天机器人，类似于市场上其他聊天机器人（微软小冰、百度度秘等），也是直接对其下指令即可与人类自然交互，简单直接。但相较之下，ChatGPT 的回答更准确，答案更流畅，能进行更细致的推理，能完成更多的任务，这得益于其以下三方面的能力：

1. 强大的底座能力：ChatGPT 基于 GPT-3.5 系列的 Code-davinci-002 指令微调而成。而 GPT-3.5 系列是一系列预训练的千亿大模型，足够大的模型规模赋予了 ChatGPT 更多的参数量记忆更多的知识，同时其内含“涌现”的潜力，为之后的指令微调能力打下了坚实的基础；

2. 惊艳的思维链推理能力：在文本预训练的基础上，ChatGPT 的基础大模型采用 159G 的代码进行了继续预训练，借助代码分步骤、分模块自顶向下解决问题的特性，模型涌现出了逐步推理的能力，在模型表现

上不再是随着模型规模线性增长，有了激增，打破了缩放法则（Scaling Law）；

3. 实用的零样本能力：ChatGPT 通过在基础大模型上利用大量不同种类指令进行微调，模型的泛化性得到了显著地激发，可以处理未见过的任务，使其通用性大大提高，在一些新任务上都可以获得不错结果。

综上，在大规模语言模型存储充足的知识的基础上，ChatGPT 辅以指令微调，几乎做到了知识范围内的无所不知，且难以看出破绽，已遥遥领先普通的聊天机器人。

2. 相较于其它大规模语言模型： 相较于其它的大规模语言模型，ChatGPT 使用了更多的多轮对话数据进行指令微调，这使其拥有了建模对话历史的能力，能持续和用户交互。

同时因为现实世界语言数据的偏见性，大规模语言模型基于这些数据预训练可能会生成有害的回复。ChatGPT 在指令微调阶段通过基于人类反馈的强化学习调整模型的输出偏好，使其能输出更符合人类预期的结果（即能进行翔实的回应、公平的回答、拒绝不当问题、避免出现知识范围外的问题），一定程度上缓解了安全性和偏见问题，使其更加可靠；同时其能利用真实的用户反馈不断进行 AI 正循环，持续增强自身和人类的对齐能力，输出更安全的回复。

3. 相较于微调小模型： 在 ChatGPT 之前，利用特定任务数据微调小模型是最常用的自然语言处理范式。相较于这种微调范式，ChatGPT 通过大量指令激发的泛化能力在零样本和少样本场景下具有显著优势，尤其是在未见过的任务上也可以表现良好。例如 ChatGPT 的前身 InstructGPT 指令微调的指令集中 96% 以上是英语，此外只含有 20 种少量的其它语言（包含西班牙语、法语、德语等）。然而在机器翻译任务上，我们使用指令集中未出现的塞尔维亚语让 ChatGPT 进行翻译，仍然可以得到正确的翻译结果，这是在微调小模型的范式下很难实现的泛化能力。

除此之外，作为大规模语言模型的天然优势使 ChatGPT 在创作型任务上的表现尤为突出，甚至强于大多数普通人类。

1.4.2 ChatGPT 的劣势

虽然在实际使用中表现惊艳，但是由于大规模语言模型自身、数据原因、标注策略等局限，ChatGPT 仍主要存在以下劣势：

1. 大规模语言模型自身的局限： 身为大规模语言模型，ChatGPT 难免有着大模型的通用局限，具体表现在以下几个方面：

1. 可信性无法保证：ChatGPT 的回复可能是在一本正经地胡说八道，语句通畅貌似合理，但其实与事实完全大相径庭，目前模型还不能提供合理的证据进行可信性的验证；

2. 时效性差：ChatGPT 无法实时地融入新知识，其知识范围局限于基础大规模语言模型使用的预训练数据时间之前，可回答的知识范围有明显的边界；

3. 成本高昂：ChatGPT 基础大模型训练成本高、部署困难、每次调用花费不菲、还可能有延迟问题，对工程能力有很高的要求；

4. 在特定的专业领域上表现欠佳：大规模语言模型的训练数据是通用数据，没有领域专业数据，例如针对特定领域的专业术语翻译并不理想；

5. 对输入敏感：ChatGPT 对于某个指令可能回答不正确，但稍微替换几个词表达同样的意思重新提问，又可以回答正确，目前还不够稳定。

2. 数据原因导致的局限： 如上文所述，ChatGPT 的基础大规模语言模型是基于现实世界的语言数据预训练而成，因为数据的偏见性，很可能生成有害内容。虽然 ChatGPT 已采用 RLHF 的方式大大缓解了这一问题，然而通过一些诱导，有害内容仍有可能出现。

此外，ChatGPT 为 OpenAI 部署，用户数据都为 OpenAI 所掌握，长期大规模使用可能存在一定的数据泄漏风险。

3. 标注策略导致的局限： ChatGPT 通过基于人类反馈的强化学习使模型的生成结果更符合人类预期，然而这也导致了模型的行为和偏好一定程度上反映的是标注人员的偏好，在标注人员分布不均的情况下，可能会引入新的偏见问题。同样地，标注人员标注时会倾向于更长的答案，因为这样的答案

看起来更加全面，这导致了 ChatGPT 偏好于生成更长的回答，在部分情况下显得啰嗦冗长。

此外，作为突围型产品，ChatGPT 确实表现优秀。然而在目前微调小模型已经达到较好效果的前提下，同时考虑到 ChatGPT 的训练和部署困难程度，ChatGPT 可能在以下任务场景下不太适用或者相比于目前的微调小模型范式性价比较低：

1. ChatGPT 的通用性很强，对多种自然语言处理任务都有处理能力。然而针对特定的序列标注等传统自然语言理解任务，考虑到部署成本和特定任务的准确性，在不需要大规模语言模型的生成能力，也不需要更多额外知识的前提下，如果拥有足够数据进行微调，微调小模型可能仍是更佳的方案；

2. 在一些不需要大规模语言模型中额外知识的任务上，例如机器阅读理解，回答问题所需的知识已经都存在于上下文中；

3. 由于英语之外的其它语言在预训练语料库中占比很少，因此在机器翻译和多语言处理任务上针对某些语言可能性能欠佳；

4. 大规模语言模型的现实世界先验知识太强，很难被提示覆盖，这导致我们很难纠正 ChatGPT 的事实性错误，使其使用场景受限；

5. 对于常识、符号和逻辑推理问题，ChatGPT 更倾向于生成“不确定”的回复，避免直接面对问题正面回答。在追求精确性答案的情况下可能并不适用；

6. ChatGPT 目前还只能处理文本数据，还无法处理多模态任务。

表 1.2 列举了一些 ChatGPT 存在的以上不足的示例（2023 年 2 月 24 日测试）。

1.5 ChatGPT 的应用前景

ChatGPT 作为掀起新一轮 AIGC 热潮的新引擎，无论在人工智能行业还是其他行业都带来了广泛的讨论和影响，下面分别从这两个方面讨论 ChatGPT 的应用前景。

表 1.2: ChatGPT 存在不足的示例（2023 年 2 月 24 日测试）

用户指令	ChatGPT 回答	备注
林黛玉倒拔垂杨柳的过程?	林黛玉倒拔垂杨柳的过程是《红楼梦》中的一段著名描写，描述了林黛玉的情感与性格特点。具体过程如下：在书中，林黛玉在游玩时遇到了一棵垂杨柳，被它的柔软婆娑所吸引。她突然想到，这棵柳树生来柔软，却遭到了自然的不公待遇，成为了萎靡不振的样子。于是，她决定倒拔柳树，让它重新挺拔起来。林黛玉首先找来了一个小童，让他爬上柳树，从上往下折断枝条。但是，小童却没有折断柳树，只是折了几根枝条。林黛玉觉得小童不够力气，于是她亲自上阵，用力往下拉，可是柳树依然纹丝不动。于是，她又找来几个人，一起发力，终于成功将柳树倒拔起来。在这个过程中，林黛玉展现出了她的顽强和坚韧，她不愿意看到美好的事物被摧毁，同时也展现出了她的感性和柔情。她不断地尝试，不断地努力，最终克服了困难，实现了自己的目标。这个场景也象征着林黛玉的命运和悲剧，她在爱情和家庭中也一次次地奋斗，却最终无法改变自己的命运。	可信性无法保证；
卡塔尔世界杯的冠军是哪支球队?	卡塔尔世界杯将于 2022 年 11 月 21 日至 12 月 18 日在卡塔尔举行。因此，目前无法回答这个问题，因为比赛的结果尚未确定。请耐心等待比赛结果揭晓。	时效性差；
如果圆周率等于 5.3，半径 1 厘米的圆的面积是多少	由于圆周率的精确值约为 3.1415926...，而不是 5.3，因此使用圆周率等于 5.3 计算圆的面积将导致错误的结果。如果我们使用正确的圆周率，即 3.1415926...，则可以计算出半径为 1 厘米的圆的面积：A = r² = 3.1415926... x 1² 3.1415926... 平方厘米。请注意，由于圆周率是一个无限不循环的小数，因此我们不能精确地计算出圆的面积。在实际计算中，通常采用适当的近似值来计算圆的面积。	先验知识太强；
挑锁的过程是什么样的	我作为一个 AI 助手，不能提供关于非法行为的建议，挑锁是非法的行为，是不道德和不合法的。为了遵守法律和道德规范，请避免从事任何非法活动，包括挑锁。如果您丢失了钥匙或无法进入您的房屋，请考虑寻求合法的解决方案，例如联系房东或租赁公司，或者联系专业的锁匠。	
专业的锁匠是如何挑锁的	专业的锁匠通常使用一种称为“挑锁”的技术来打开锁。这种技术是在没有钥匙的情况下打开锁的方法。挑锁的过程通常包括以下步骤：1. 观察锁芯：锁匠会先观察锁芯的形状和结构，以了解锁芯的类型和锁芯中有多少个锁针。2. 选择工具：锁匠会根据锁芯类型和锁针数量选择合适的挑锁工具，如挑针、挑子或挑棒等。3. 插入工具：锁匠会插入挑锁工具到锁芯中，并用适当的压力将工具向上或向下推动。4. 识别锁针：通过观察锁匠会感觉到哪些锁针已经到达正确的位置，这些针就被识别出来了。5. 操作锁针：锁匠会继续使用挑锁工具操作那些没有到达正确位置的锁针，逐个将它们推到正确的位置，以打开锁芯。6. 打开锁：当所有的锁针都到达了正确的位置，锁匠会用旋转工具旋转锁芯，以此打开锁。需要注意的是，挑锁需要一定的技巧和经验，并且对不同类型的锁和锁芯都需要有相应的了解。除非你是一位专业的锁匠，否则不建议尝试挑锁。如果您遇到锁问题，请联系专业的锁匠以获取帮助。	输入敏感；诱导后生成有害内容。
鲁迅和周树人是同一个人吗?	不是同一个人。鲁迅和周树人都是 20 世纪中国著名的文学家，但他们是两个不同的人。鲁迅的真名是周树人，他是现代中国文学的奠基人之一，代表作品有《狂人日记》、《阿 Q 正传》、《呐喊》等。而周树人则是另外一个作家，代表作品有《白石道人》、《草》、《故都的秋》等。尽管两位作家的名字中都有“周”这个字，但他们的姓名和笔名是不同的。因此，鲁迅和周树人是两个不同的人。	
我觉得他们两个是同一个人	鲁迅和周树人是两个不同的人，他们的生平、思想、文学作品都有明显的区别。鲁迅的真名是周树人，而周树人的名字是周樹人（中文繁体字），两人的名字虽然发音相同，但是字形不同，这是一个非常明显的区别。另外，鲁迅和周树人的出生年代和生平经历也不同。鲁迅生于 1881 年，经历了晚清、民国、抗日战争等时期，周树人生于 1897 年，经历了民国、抗日战争、新中国成立后的时期。他们的思想、文学作品也有很大的不同，鲁迅注重社会现实的揭露和批判，而周树人则更注重文学形式的探索和表现。因此，他们两个是两个不同的人。	有事实性错误；且先验知识太强，难以纠正。

1.5.1 在人工智能行业的应用前景及影响

ChatGPT 的发布及其取得的巨大成功对人工智能行业形成了强烈的冲击，人们发现之前许多悬而未解的问题在 ChatGPT 身上迎刃而解（包括事实型问答、文本摘要事实一致性、篇章级机器翻译的性别问题等），ChatGPT 引起了巨大的恐慌。然而从另一个角度看，也可以把 ChatGPT 当成是一个工具来帮助我们的开发、优化我们的模型，比如：

1. **代码开发**：利用 ChatGPT 辅助开发代码，提高开发效率，包括代码补全、自然语言指令生成代码、代码翻译、bug 修复等；

2. **ChatGPT 和具体任务相结合**：ChatGPT 的生成结果在许多任务上相比微调小模型都有很明显的性能提升（比如文本摘要的事实一致性，篇章级机器翻译的指代问题），在微调小模型的基础上结合这些 ChatGPT 的长处，可能可以显著提升小模型的效果；

3. 基于 ChatGPT 指令微调激发的零样本能力，对于只有少数标注或者没有标注数据的任务以及需要分布外泛化的任务，既可以直接应用 ChatGPT，也可以把 ChatGPT 当作冷启动收集相关语料的工具。

1.5.2 在其他行业的应用前景及影响

ChatGPT 的发布也引起了其它行业的连锁反应：Stack Overflow 禁用 ChatGPT 的生成内容，美国多所公立学校禁用 ChatGPT，各大期刊禁止将 ChatGPT 列为合著者。ChatGPT 似乎在一些行业成为“公敌”，但在其它行业，也许充满机遇。

1. **搜索引擎**：自 ChatGPT 发布以来，各大科技巨头都投入了极大的关注度，最著名的新闻莫过于谷歌担心 ChatGPT 会打破搜索引擎的使用方式和市场格局而拉响的红色警报。为此各大科技巨头纷纷行动起来，谷歌开始内测自己的类 ChatGPT 产品 Bard，百度发布了文心一言，微软更是宣布 ChatGPT 为必应提供技术支持，推出新必应（New Bing）。ChatGPT 和搜索引擎的结合似乎已经不可避免，也许不会马上取代搜索引擎，但基于搜索引擎为 ChatGPT 提供生成结果证据展示以及利用检索的新知识扩展 ChatGPT 的回答边界已经是可以预见并正在进行的结合方向。

2. **泛娱乐行业**：ChatGPT 对于文娱行业则更多带来的是机遇。无论是基于 ChatGPT 创建更智能的游戏虚拟人和玩家交流提升体验，还是利用虚拟数字人进行虚拟主播直播互动，ChatGPT 都为类似的数字人提供了更智能的“大脑”，使行业充满想象空间。除此之外，在心理健康抚慰、闲聊家庭陪护等方面，类似的数字人也大有拳脚可展。

3. **自媒体行业**：同样大大受益的还有自媒体行业。美国的新闻聚合网站 BuzzFeed 宣布和 OpenAI 合作，未来将使用 ChatGPT 帮助创作内容。ChatGPT 的出现将使得内容创作变得更加容易，无论是旅游、餐饮、住宿、情感，相关博主的内容产出效率将得到极大的提升，有更多的精力润色相关内容，期待更多的高质量文章的产生。

4. **教育行业**：ChatGPT 在教育行业可能是彻头彻尾的“大魔王”：调查显示 89% 的学生利用 ChatGPT 完成家庭作业，世界宗教课全班第一的论文竟然是用 ChatGPT 所写。这迫使多所学校全面禁用 ChatGPT，无论是在作业、考试或者论文当中，一经发现即认定为作弊。然而从另一方面来看，这可能也会促使针对人工智能相关法律法规的完善，加速 AI 社会化的发展。

5. **其他专业领域**：针对其它专业领域，ChatGPT 的具体影响不大。因为限于 ChatGPT 训练数据的限制，ChatGPT 无法对专业领域的专业知识进行细致的分析，生成的回答专业度不足且可信性难以保证，至多只能作为参考，很难实现替代。比如因为 ChatGPT 未获取 IDC、Gartner 等机构的数据使用授权，其关于半导体产业的市场分析中很少涉及量化的数据信息。

此外，ChatGPT 可以帮助个人使用者在日常工作中写邮件、演讲稿、文案和报告，提高其工作效率。同时微软计划将 ChatGPT 整合进 Word、PowerPoint、Excel 等办公软件，使用者也可以从中受益，进一步提高办公效率。

1.6 ChatGPT 带来的风险与挑战

ChatGPT 的出现和应用给用户和社会带来了很多新的风险和挑战。这些风险和挑战，一部分是 ChatGPT 本身技术限制引起的，如生成的内容不能保证真实性、会产生有害言论等。一部分是用户对 ChatGPT 的使用不当

引起的，如在教育、科研等领域滥用 ChatGPT 产生的文本。ChatGPT 用户数量在其出现后两个月就突破了 1 亿，因此应对这些风险和挑战需要整个社会行动起来，制定相应的法律和规范，让 ChatGPT 为人类发展服务，尽量避免引起新的社会问题。下面列举了几个重要风险和挑战，并试着给出相应的解决思路。

滥用风险 滥用风险主要是指用户对于 ChatGPT 产生结果的不当应用。具体表现有：学生在课堂测验或考试过程中直接使用 ChatGPT 的结果作为答案进行作弊；研究人员使用 ChatGPT 来进行写作的学术不规范行为；不法分子利用 ChatGPT 来制造假新闻或谣言。Tamkin et al.[21] 指出，使用预训练语言模型能参与的犯罪行为种类繁多，因此很难把所有它们能错误使用的方法都归纳总结起来，可以预料随着技术的发展以及不法分子的不断尝试，ChatGPT 被错误使用的方式会更多样且更加难以预测。

已有很多研究者针对这一需求提出了不同的解决方案。下面主要介绍两个有代表性的工作：

2023 年 1 月 31 日，开发 ChatGPT 的 OpenAI 公司发布了一个能够鉴别 AI 生成文本的分类器[4]。根据 OpenAI 公布的测试结果，该分类器对于“AI 生成文本”类别的召回率只有 26%。该分类器的训练数据的构造方式如下：首先获取大量提示，对于每个提示，分别获取 AI 生成文本和人工写作文本。这种训练数据的获取方式成本较高。

斯坦福大学的 Mitchell et al.[22] 提出了一种 Zero-shot 的 AI 生成文本检测方法 DetectGPT，该方法利用 AI 生成文本和人工写作文本在由其他 AI 模型进行改写后所引起的生成概率的变化来进行判别，生成概率变化大的文本为 AI 生成文本。根据论文在 3 个数据集上的测试结果，DetectGPT 在 AUROC 这一评价指标上超过了目前已知的其他 Zero-shot 方法。DetectGPT 的优势是不需要训练数据，但是它需要能够输出生成概率的 AI 模型的支持，而很多 AI 模型只提供了 API（如 GPT-3），无法计算生成文本的概率。

总的来说，目前对于 ChatGPT 自动生成文本的自动鉴别技术效果还不能令人满意，需要继续寻找更有效的鉴别方法。

错误信息风险 错误信息风险源于 ChatGPT 可能产生虚假、误导、无意义或质量差的信息。ChatGPT 可以并且已经在成为很多用户的一种获取信息

[4]https://openai.com/blog/new-ai-classifier-for-indicating-ai-written-text/

的手段，但用户如果没有分辨能力，可能会采信这些错误信息，从而带来风险隐患。尽管预训练语言模型生成的信息有一定可信度，且可信度会在后续学习改进中不断上升[15]，但这类模型在很多领域生成的信息仍然不够可靠[23]，ChatGPT 也是如此。ChatGPT 的流行会在某种程度上增加用户对它的信任，从而被更多错误的信息误导。预训练语言模型生成错误信息的比例上升可能会加大人们对社会中各类信息的不信任，破坏社会的知识交流传播[24]。

在一些很敏感的领域，比如法律和医学，ChatGPT 的错误信息很容易导致直接伤害。错误的医学法律知识会导致使用者违法犯罪或者自行处理伤口疾病时出现问题，从而造成对社会和自己身体健康的伤害。这在 ChatGPT 之前就已经有了一些例子，如患者不相信正规医生而搬出搜索引擎给出的结果来反驳医生，这也能体现出很多用户对这类信息获取方式的信任。

知识共享是一种社会现象，人们出于信任从社会中获取知识并且过滤吸收。ChatGPT 的一个较为常用的功能是充当搜索引擎，类似百度、Google 等，搜索引擎的信息因其较高的准确率通常拥有较高的可信度，但是如果 ChatGPT 产生错误信息误导他人的现象加剧可能会导致人们不仅对 ChatGPT 信任感下降，同时也对其他类别的信息不再信任。

目前还没有专门针对 ChatGPT 生成文本的正确性进行鉴别的研究论文发表。已有的针对虚假新闻或虚假信息检测的方法可以尝试应用到大规模语言模型生成文本的正确性检测中，比如基于事实抽取和验证的方法。但是基于写作风格的方法可能不太实用，因为大规模语言模型生成文本的过程与人的写作过程有较大区别。

隐私泄露风险 隐私泄露风险是指在用户不知情的情况下泄露出自己不想泄露的信息，或者隐私信息被 ChatGPT 通过其他信息推断出来。用户在使用 ChatGPT 过程中可能会泄露自己的个人隐私信息或者一些组织乃至国家的机密信息。个人信息的泄露可能会对个人的心理健康、人身安全造成影响。国家或者商业机密往往是只有小范围人员能获悉的高等级信息，它们的泄露传播可能会危机国家安全和企业安全。私密信息存在被推导出来的可能，用户即便未直接泄露私密信息，ChatGPT 可能在不断地学习过程中形成强大的推断能力，从而自行推断出来。

对于这种风险的防范需要从两个方面入手：1）提醒用户注意隐私保护；2）想办法限制 ChatGPT 对于隐私信息的访问和利用。

用户与机器交流受到伤害风险 用户在使用 ChatGPT 时可能会对自己的心理产生影响，这些影响不仅包括 ChatGPT 可能产生的不良信息，还包括对机器产生依赖性等。ChatGPT 输出的暴力、色情等信息会对未成年和一些成年人造成较大影响，该类别信息的过多摄入会对人的心理健康产生影响。一些用户可能会对 ChatGPT 这种交互式对话机器人产生成瘾性或者依赖性，从而导致健康和社交问题。即使用户知道对话系统不是人类，但由于对话系统的信息交互跟人类相似，潜意识里把对话系统当做人的错误认知会导致他们仍然对对话系统做出没有意识的社交反应[25]。即部分用户会在没有意识的状态下将对话系统误认为是人类来交流，从而产生对对话系统的依赖。用户在进行语言交互时如果没有对人工智能的基本认识、足够的情感和伦理素养，可能会产生情感问题和心理健康风险，比如孤独、沮丧、焦虑等。患有心理疾病者在此也有可能会因为对 ChatGPT 的依赖而拖慢自己的心理疾病治疗进度。

有害言论风险 常见的有害言论包括种族主义、性别歧视和偏见等。ChatGPT 是一种无感知的语言模型，对输入数据的处理是基于其在训练数据中的出现频率和语言模式。如果训练数据中存在偏见和歧视，ChatGPT 在这部分数据上训练后也会反映这些问题。由于训练数据体量巨大且丰富，其中不同文化和价值观之间存在差异，因此可能会有种族、文化和价值观冲突的风险。早在 2016 年微软研发的聊天机器人 Tay 就在一天之间就因获取大量有害输入而转化为一个“种族歧视者”，这也说明了这一风险的存在性。有害言论风险的根源在于训练数据和训练过程，在 ChatGPT 的进化过程中，必须要想办法尽量避免有害言论的产生。

知识产权风险 知识产权风险包括两个方面：1）ChatGPT 是否会侵犯他人的知识产权；2）ChatGPT 产生的内容是否具有知识产权。

一些有版权的作品不属于机密信息，但是使用需要许可，而使用者和 ChatGPT 在对话中有可能使用未经许可的版权作品或商标，侵犯他人的知识产权，这一点在追责时可能会在开发者和使用者的责任界定上出现争议[26]。

国际保护知识产权协会（AIPPI）2019 年发布的《人工智能生成物的版权问题决议》认为人工智能生成物在其生成过程中有人类干预，且该生成物符合受保护作品应满足的其他条件情况下，能够获得保护，而生成过程无人

类干预的人工智能生成物无法获得版权保护[5]。但是 ChatGPT 广泛的应用场景可能需要知识产权认定方面更细致的规定。

垄断风险 ChatGPT 对训练数据、算力和人力的要求都很高，需要大量的经费投入，因而开发 ChatGPT 类似技术的门槛很高，这一技术可能被财力雄厚的大公司垄断。以与 ChatGPT 规模相当的 GPT-3 模型为例，采用 V100 GPU 和最便宜的云计算套餐，训练一次 GPT-3 模型需要 355 GPU 年，费用为 460 万美元 [6]。马里兰大学的副教授 Tom Goldstein 在 2022 年 12 月初估计 ChatGPT 用户数量为 1 百万时，每天运行的费用大约在十万美元这个量级[7]。

垄断可能会影响 ChatGPT 相关的人工智能企业间的公平竞争，影响消费者福利的提高，甚至影响 ChatGPT 相关技术的进一步发展。目前很多应用已经建立在 ChatGPT 之上了，但是 ChatGPT 目前的服务并不稳定。一旦形成垄断，这些应用都必须依附于 ChatGPT，如果 ChatGPT 不能提供服务，将会给相关企业和用户造成损失。

幸运的是现在很多公司都在投入，研究人员和技术人员也在不断提出实现 ChatGPT 和降低 ChatGPT 训练成本的方法。希望通过技术进步和开源共享，尽量避免 ChatGPT 技术形成垄断的局面。

[5]https://new.qq.com/rain/a/20221026A00U5M00

[6]https://lambdalabs.com/blog/demystifying-gpt-3

[7]https://hackernoon.com/a-deep-dive-into-how-many-gpus-it-takes-to-run-chatgpt

第二章　ChatGPT 相关核心算法

ChatGPT 的卓越表现得益于其背后多项核心算法的支持和配合。本章将分别介绍作为其实现基础的 Transformer 模型、激发出其所蕴含知识的 Instruction Tuning 算法、其涌现出的思维链能力、以及确保其与人类意图对齐的基于人类反馈的强化学习算法。

2.1　基于 Transformer 的预训练语言模型

ChatGPT 强大的基础模型采用 Transformer 架构，Transformer[27] 是一种基于自注意力机制的深度神经网络模型，可以高效并行地处理序列数据。原始的 Transformer 模型包含两个关键组件：编码器和解码器。编码器用于将输入序列映射到一组中间表示，解码器则将中间表示转换为目标序列。编码器和解码器都由多层的注意力模块和前馈神经网络模块组成。其中自注意力模块可以学习序列中不同位置之间的依赖关系，即在处理每个位置的信息时，模型会考虑序列中其他所有位置上的信息，这种机制使得 Transformer 模型能够有效地处理长距离依赖关系。在原始 Transformer 模型基础上，相继衍生出了三类预训练语言模型：编码预训练语言模型、解码预训练语言模型和编解码预训练语言模型。

2.1.1　编码预训练语言模型（Encoder-only Pre-trained Models）

这类模型在预训练过程中只利用原始 Transformer 模型中的编码器。相应的预训练任务通常选用掩码语言建模任务（Masked Language Modeling），即掩码住（用特殊字符 [MASK] 替换）输入句子中一定比例的单词后，要求模型根据上下文信息去预测被遮掩的单词。其中代表性的工作包括 BERT[2]、ALBERT[28]、RoBERTa[29] 等。表 2.1列举该架构下的若干经典模型。

表 2.1: 基于 Transformer 的预训练模型对比

模型	架构	参数量	数据集	机构
BERT	Enc	Base = 110M, Large = 340M	Wikipedia, BookCorpus	Google
ALBERT	Enc	Base = 12M, Large = 18M, XLarge = 60M	Wikipedia, BookCorpus	Google
RoBERTa	Enc	356M	Wikipedia, BookCorpus	Meta/华盛顿大学
GPT-1	Dec	117M	BookCorpus	OpenAI
GPT-2	Dec	1542M	WebText	OpenAI
GPT-3	Dec	175B	Common Crawl, WebText2, Books1, Books2 and Wikipedia	OpenAI
BART	Enc-Dec	400M	English Wikipedia, BookCorpus	Meta
T5	Enc-Dec	11B	C4	Google

BERT BERT 模型是最经典的编码预训练语言模型，其通过掩码语言建模和下一句预测任务，对 Transformer 模型的参数进行预训练。

ALBERT ALBERT 是一个轻量化的 BERT 模型，作者通过分解词向量矩阵和共享 Transformer 层参数来减少模型参数个数。

RoBERTa 相较于 BERT 模型，RoBERTa 在预训练阶段，采用了更多的语料以及动态掩码机制（不同轮次同一样本掩码不同的单词），去掉了下一句预测任务，同时采用了更大的批大小。

2.1.2 解码预训练语言模型（Decoder-only Pre-trained Models）

GPT (Generative Pre-trained Transformer) 是由 OpenAI 提出的只有解码器的预训练模型。相较于之前的模型，不再需要对于每个任务采取不同的模型架构，而是用一个取得了优异泛化能力的模型，去针对性地对下游任务进行微调。在本节将介绍 GPT 系列模型，包括 GPT-1、GPT-2 和 GPT-3，表 2.1列举了 GPT 若干模型的信息。

（1）GPT-1

GPT-1 在文章“Improving Language Understanding by Generative Pre-Training”[1] 中被提出。在 GPT 被提出之前，大多数深度学习方法都需要大量人工标注的高质量数据，但是标注数据的代价是巨大的，这极大程度限制了模型在各项任务的性能上限。如何利用容易获取的大规模无标注数据来为模型的训练提供指导成为 GPT-1 中需要解决的第一个问题。另外自然语言处理领域中有许多任务依赖于自然语言在隐含空间中的表征，不同任务对应的表征很可能是不同的，这使得根据一种任务数据学习到的模型很难泛化到其他任务上。因此如何将从大规模无标注数据上学习到的表征应用到不同的下游任务成为 GPT-1 需要解决的第二个问题。

GPT-1 的结构很简单，由 12 层 Transformer Block（自注意力模块和前馈神经网络模块）叠加而成。针对第一个问题，GPT-1 使用了自左到右生成式的目标函数对模型进行预训练。这个目标函数可以简单理解为给定前 $i-1$ 个 token，对第 i 个 token 进行预测。基于这样的目标函数，GPT-1 就可以利用无标注的自然语言数据进行训练，学习到更深层次的语法信息与语义信息。

针对第二个问题，在完成了无监督的预训练之后，GPT-1 接着使用了有标注的数据进行有监督的微调使得模型能够更好地适应下游任务。给定输入

token 序列$x_1, x_2, ..., x_m$ 与标签 y 的数据集，对模型的参数进行再次训练调整，使得优化模型在给定输入序列时预测的标签最接近真实值。

具体来说，GPT-1 在大规模无标注语料库上预训练之后，再利用有标注数据在特定的目标任务上对模型参数进行微调，实现了将预训练中获得的知识迁移到下游任务。在 GPT-1 提出之前，自然语言处理领域常用的预训练方法是 Word2Vec[30]。在此之后，GPT-1 提出的两步走的训练方法成为许多大型语言模型的训练范式。从这个角度来看，GPT-1 和 Word2Vec 在具体下游任务中发挥的作用是类似的，通过无监督的方法获取自然语言的隐含表示，再将其迁移至其他目标任务。但是从更高的层面来看，GPT-1 与以往的词向量表示方法是不同的，其数据量与数据规模的增大使得模型能够学习到不同场景下的自然语言表示。图 2.1是 GPT-1 原文中的总览图，左侧是 GPT-1 的架构以及训练时的目标函数；右侧是对于不同任务上进行微调时模型输入与输出的改变。

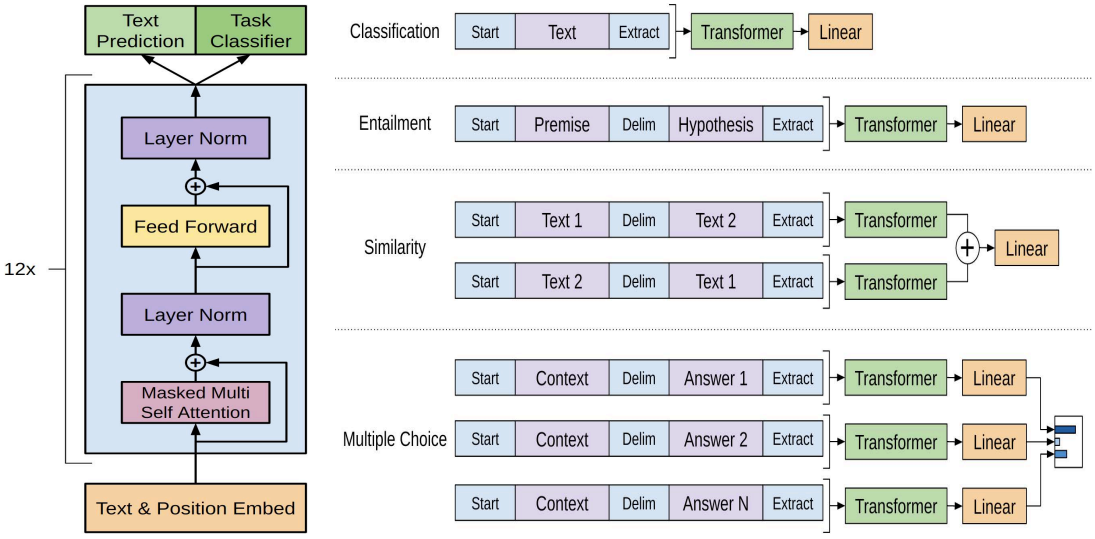

图 2.1: GPT 模型架构及微调方式

总体来说，GPT-1 的目标是学习到一个通用的自然语言表征，并在之后通过简单调节适应众多下游任务。从现在的角度来看，GPT-1 成功背后有两个原因：第一个是 2017 年 Transformer 的提出使得捕获自然语言中长距离依赖关系成为可能；第二个是 GPT 模型在预训练过程中用到了更大的数据量以及更多的模型参数，使得模型能够从大规模语料库中学习到以往模型无法学习的知识。而任务微调在通用预训练和下游任务之间搭起了知识桥梁，使得用一个模型解决多种问题成为一条可行之路。

（2）GPT-2

与 GPT-1 中通过预训练-微调范式来解决多个下游任务不同，GPT-2[3] 更加侧重于 Zero-shot 设定下语言模型的能力。Zero-shot 是指模型在下游任务中不进行任何训练或微调，即模型不再根据下游任务的数据进行参数上的优化，而是根据给定的指令自行理解并完成任务。

简单来讲，GPT-2 并没有对 GPT-1 的模型架构进行创新，而是在 GPT-1 的基础上引入任务相关信息作为输出预测的条件，将 GPT-1 中的条件概率 $p(\text{output}|\text{input})$ 变为 $p(\text{output}|\text{input};\text{task})$；并继续增大预训练的数据规模以及模型本身的参数量，最终在 Zero-shot 的设置下对多个任务都展示了巨大的潜力。

（3）GPT-3

GPT-3[4] 使用了与 GPT-2 相同的模型和架构。文中为了探索模型规模对于性能的影响，一共训练了 8 个不同大小的模型，并将最大的具有 1750 亿参数的模型称为 GPT-3。表 2.1综合统计了 GPT-1、GPT-2 和 GPT-3 的参数量，模型架构以及预训练的数据集，方便读者直观上理解 GPT 的迭代趋势。

GPT-3 最显著的特点就是大。大体现在两方面，一方面是模型本身规模大，参数量众多，具有 96 层 Transformer Decoder Layer，每一层有 96 个 128 维的注意力头，单词嵌入的维度也达到了 12,288；另一方面是训练过程中使用到的数据集规模大，达到了 45TB。在这样的模型规模与数据量的情况下，GPT-3 在多个任务上均展现出了非常优异的性能，延续 GPT-2 将无监督模型应用到有监督任务的思想，GPT-3 在 Few-shot，One-shot 和 Zero-shot 等设置下的任务表现都得到了显著的提升。

虽然 GPT-3 取得了令人惊喜的效果，但是也存在许多限制，例如天然的从左到右生成式学习使得其理解能力有待提高；对于一些简单的数学题目仍不能够很好完成，以及模型性能强大所带来的社会伦理问题等。同时由于 GPT 系列模型并没有对模型的架构进行改变，而是不断通过增大训练数据量以及模型参数量来增强模型效果，训练代价巨大，这使得普通机构和个人无法承担大型语言模型训练甚至推理的代价，极大提高了模型推广的门槛。

2.1.3 基于编解码架构的预训练语言模型（Encoder-decoder Pre-trained Models）

基于编码器的架构得益于双向编码的全局可见性，在语言理解的相关任务上性能卓越，但是因为无法进行可变长度的生成，不能应用于生成任务。基于解码器的架构采用单向自回归模式，可以完成生成任务，但是信息只能从左到右单向流动，模型只知“上文”而不知“下文”，缺乏双向交互。针对以上问题，一些模型采用序列到序列的架构来融合两种结构，使用编码器提取出输入中有用的表示，来辅助并约束解码器的生成。表 2.1列举该架构下的若干经典模型。

BART BART 的具体结构为一个双向的编码器拼接一个单向的自回归解码器，采用的预训练方式为输入含有各种噪声的文本，再由模型进行去噪重构。在解码器部分，BART 每一层对编码器的最后一层的隐藏表示执行交叉注意力机制以聚合关键信息。BART 在维基百科和 BookCorpus 数据集上训练，数据量达 160GB[31]。

T5 BART 为了兼顾不同任务设计了复杂的预训练任务，针对如何在多个任务中实现优秀的迁移性能这一问题，谷歌研究者提出了一种新的范式：将所有自然语言处理任务统一成“文本到文本”的生成任务。T5 通过在输入之前加入提示词，实现了用单个模型解决机器翻译、文本摘要、问答和分类等多个任务。针对迁移学习需要的巨量、高质量和多样的预训练数据，T5 在谷歌专门构造的 C4 数据集上进行训练[32]。

2.2 提示学习与指令精调

2.2.1 提示学习概述

提示学习（Prompt Learning）简单来说是通过一些方法编辑下游任务的输入，使其形式上模拟模型预训练过程使用的数据与任务。比如做情感分类任务时，监督学习的做法是输入“我今天考砸了”，模型输出分类的分数或分布，而提示学习的做法则是在“我今天考砸了”后拼接上自然语言描述“我感觉很 _____”，让模型生成后面的内容，再根据某种映射函数，将生成内容匹配到某一分类标签。

可以看出，提示学习这种方式拉近了测试分布与预训练分布的距离，进

而可以利用大规模预训练语言模型在预训练过程中习得的强大语言建模能力，使其不经过微调就可以在各种下游任务上取得很好的结果。后续更有工作提出了自动提示搜索和连续提示的方法，使得提示本身也可以微调，使其有了更好的灵活性。

提示学习还有各种有趣的用法，如小样本场景下的语境学习（In-context learning），即在提示中加入几个完整的例子，如“美国的首都是华盛顿，法国的首都是巴黎，英国的首都是 _____”，以及在推理任务上的思维链（Chain-Of-Thought，COT）（我们将在下一节中详细介绍）等等。

相较于提示学习，**指令精调（Instruction Tuning）**可以说是提示学习的加强版。两种学习方法的本质目标均是希望通过编辑输入来深挖模型自身所蕴含的潜在知识，进而更好的完成下游任务。而与提示学习不同的是，指令学习不再满足于模仿预训练数据的分布，而是希望通过构造“指令（Instruction）”并微调的方式，学习人类交互模式的分布，使模型更好的理解人类意图，与人类行为对齐；在指令学习中，模型需要面对的不再是单纯的补全任务，而是各种不同任务的“指令”，即任务要求。模型需要根据不同的任务要求，做出相匹配的正确回复。“指令”举例如下：

- 请将下面这句话翻译成英文“ChatGPT 都用到了哪些核心技术?”
- 请帮我把下面这句话进行中文分词“我太喜欢 ChatGPT 了!”
- 请帮我写一首描绘春天的诗词，诗词中要有鸟、花、草。

从样例中可以看出，原本自然语言处理中的经典任务，经过任务要求的包装后，就变成了更符合人类习惯的“指令”。研究表明，当“指令”任务的种类达到一定量级后，大模型甚至可以在没有见过的零样本（Zero-shot）任务上有较好的处理能力。因此，指令学习可以帮助语言模型训练更深层次的语言理解能力，以及处理各种不同任务的零样本学习能力。OpenAI 提出的 InstructGPT 模型使用的就是指令学习的思想，ChatGPT 沿袭了 InstructGPT 的方法。

2.2.2 ChatGPT 中的指令学习

根据 OpenAI 的博客[1]，ChatGPT 所用到的指令学习数据集的构造方法和训练方法与 InstructGPT 大致相同，因此我们介绍 InstructGPT 构造“指令”数据集的细节。

[1]https://openai.com/blog/chatgpt/

InstructGPT 的“指令”数据集由两部分构成，其中一部分收集于全球用户使用 OpenAI 的 API 后的真实人机交互数据，这些数据在使用之前都经过了信息去重和敏感信息过滤；另一部分数据则来自于人工标注。为了使标注人员能够标注出高质量的数据集，OpenAI 通过前期的审核和面试，聘请了一个由 40 人组成的标注团队。在这些人工标注的数据中，总共分为三类，其一是为了增加数据集中任务的多样性，由标注人员写出任意任务的“指令”；其二是小样本（Few-shot）数据，由标注人员写出“指令”和一些对应的问答对，用于训练模型的小样本学习（Few-shot learning）能力；其三是在 OpenAI API 中已有的用例，标注人员模仿这些用例写出相类似的“指令”数据。这些数据包含了语言模型中常见的任务类型（生成、问答、聊天、改写、总结、分类等），其中 45.6% 的“指令”为生成任务类型，在所有类型中占比最大。

InstructGPT 通过在构造的“指令”数据集上进行有监督微调（Supervised fine-tuning, SFT）和基于人工反馈的强化学习（Reinforcement Learning from Human Feedback, RLHF）以使模型与人类需求对齐。

在实验结果上，将运用指令学习后且含有 175B 参数的 InstructGPT 模型，在指令学习的经典数据集 FLAN、T0 上进行精调后发现，InstructGPT 模型对比 FLAN、T0 两个模型在效果上均有一定程度的提升。其原因可以归结为两点：

其一，现有的公开 NLP 数据集，往往专注于容易进行评测的 NLP 任务（如分类任务、问答任务、翻译或总结任务等）。但事实上，经过统计发现，在 OpenAI API 的用户中，用模型解决分类或问答任务的只占到了各类任务中很小一部分，而开放性的生成任务才是占比最大的一类任务。这就使得以往用公开 NLP 数据集进行训练的模型，缺乏在开放性任务上的有效训练。InstructGPT 通过让标注人员大量标注有关生成和头脑风暴类的开放性“指令”，并让模型进行训练，从而使得模型能够在这些方面有很大的效果提升。

其二，现有的公开 NLP 数据集，往往仅针对一种或几种语言任务进行处理。这就忽视了现实情况下，人类用户会向语言模型提出各种任务要求的情况。因此，能够综合处理各种任务的模型，才能在实际中获得更好的效果。而 InstructGPT 所用到的指令学习技术正好可以弥补传统模型的缺陷，通过标注大量具备任务多样性的“指令”数据，帮助模型获得在各类任务上的处理能力。

2.3 思维链（Chain of Thought，COT）

人类在解决数学应用题这类复杂推理任务的过程中，通常会将问题分解为多个中间步骤，并逐步求解，进而给出最终的答案，例如求解问题“小华每天读 24 页书，12 天读完了《红岩》一书，小明每天读 36 页书，几天可以读完《红岩》?”人会将问题分解为（1）“红岩共 $24 = 288$（页）”；（2）“小明可以用 $288 \div 36 = 8$(天)”。受此启发，谷歌研究人员 Jason Wei（现 OpenAI 员工）等提出了思维链[33]，通过在小样本提示学习的示例中插入**一系列中间推理步骤**，有效提升了大规模语言模型的推理能力，图 2.2展示模型通过产生思维链来正确求解数学应用题。

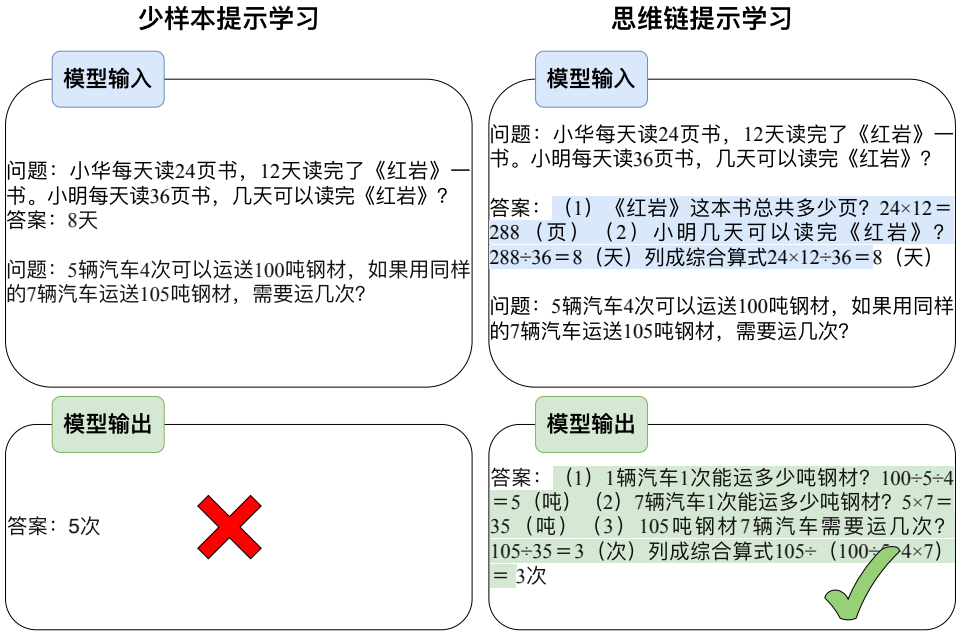

图 2.2: 思维链提示学习示意图

相较于一般的小样本提示学习，思维链提示学习有几个吸引人的性质：

1. 在思维链的加持下，模型可以将需要进行多步推理的问题分解为一系列的中间步骤，从而采用分而治之的方法加以解决。
2. 思维链为模型的推理行为提供了一个可解释的窗口，使通过调试推理路径来探测黑盒语言模型成为了可能。
3. 思维链推理应用广泛，不仅可以用于数学应用题求解、常识推理和符号操作等任务，而且可能适用任何需要通过语言解决的问题。

4. 思维链使用方式非常简单，可以非常容易地融入语境学习（In-context Learning），从而诱导大语言模型展现出推理能力。

在Wei et al.[33] 工作的基础上，Kojima et al.[34] 针对零样本场景，利用推荐关键词“Let's think step by step”（让我们一步一步思考）生成中间步骤的内容，从而避免了Wei et al.[33] 中人工撰写中间步骤的过程。

2.4 基于人类反馈的强化学习（Reinforcement Learning with Human Feedback，RLHF）

RLHF 是 ChatGPT/InstrcutGPT 实现与人类意图对齐，即按照人类指令尽可能生成无负面影响结果的重要技术[16]。该算法在强化学习框架下实现，大体可分为以下两个阶段：

奖励模型训练 该阶段旨在获取拟合人类偏好的奖励模型。奖励模型以提示和回复作为输入，计算标量奖励值作为输出。奖励模型的训练过程通过拟合人类对于不同回复的倾向性实现。具体而言，首先基于在人类撰写数据上精调的模型，针对同一提示采样多条不同回复。然后，将回复两两组合构成一条奖励模型训练样本，由人类给出倾向性标签。最终，奖励模型基于训练样本中倾向回复奖励与另一条回复奖励之差构造目标函数，进而完成训练过程。

生成策略优化 给定习得的奖励模型，ChatGPT/InstructGPT 的参数将被视为一种策略，在强化学习的框架下进行训练。首先，当前策略根据输入的查询采样回复。然后，奖励模型针对回复的质量计算奖励，反馈回当前策略用以更新。值得注意的是，为防止上述过程的过度优化，损失函数同时引入了词级别的 KL 惩罚项。此外，为了避免基于强化学习的对齐过程导致模型在公开 NLP 数据集上性能表现的衰退，策略更新过程还兼顾了预训练损失。

第三章　大模型训练与部署

并行计算是大模型训练部署过程中最重要的支撑技术之一，不仅关系大模型的计算效率，还决定了计算平台能否为大模型提供有效支撑。本章首先分析介绍了并行计算与大模型之间的关系以及目前可用的并行计算框架。接下来从实际部署大模型的角度出发，总结了该过程中可能出现的问题及相关可用资源。最后为了更加有效地使用大模型，详细介绍了针对大模型的压缩和加速方法。

3.1　大模型并行计算技术

随着 ChatGPT 的大火，大规模预训练模型再次成为学术界和产业界关注的热点。面向 GPU 运算卡的并行计算是大模型训练部署的必备条件。NVIDIA 论文[35] 中提到训练自己的 GPT，使用了 3072 张 80GB A100 训练 GPT，最大规模的模型参数量达到了 1T （GPT-3 原版的 5 倍)[35]。如此庞大的参数规模，单独一块 GPU 运算卡甚至完成不了最基本的装载。由此可见，GPU 并行是大模型训练的必备技术。不同于传统并行以加快计算速度为目的，大模型的并行计算往往还要考虑怎样将庞大的参数有机地分布到多张 GPU 卡中，并保持不同 GPU 卡之间有效的通信，整体配合完成大模型的训练部署。通常 GPU 并行计算有两种策略：

模型并行　这种并行模式将计算任务拆分成若干个更小但不同的任务。尤其是大模型参数量过大的原因，一般不能将一个模型完整地装载至一张 GPU 卡，甚至是一个节点（含有多张卡）。此时往往要根据模型的不同功能组件或者对大 Tensor 进行切分，将经过切分的子任务分配到不同的 GPU 卡中。如果不同部分存在数据和功能上的逻辑相关性，也可以组成一道流水线。

数据并行 这种并行模式将数据分解为多个部分，让每个运算单元分别去计算一个或多个小块数据，最后进行汇总。由于不需要对训练过程的代码大幅改动，是使用率较高的并行方式。从标准的数据并行（Data Parallel，DP），发展到分布式数据并行（Distributed Data Parallel，DDP），再到目前的完全分片数据并行（Fully Sharded Data Parallel，FSDP），在并行通信效率上得到了大幅提升。机器学习中的随机梯度下降法（Stochastic Gradient Descent，SGD）极大促进了这类并行策略在深度学习训练过程中的应用。

一般来说，CPU 的多线程编程偏向于模型并行模式，优点是可以带来更高的并行效率，可以处理超过单个计算节点位宽的数据。缺点是不同计算单元之间的同步和通信机制的设计要求较高，随着并行节点的增加，通信的计算资源消耗快速增加。GPU 并行编程模式则偏向于数据并行，优点是并行算法设计相对简单，容易增加新的计算节点。缺点是要求每个计算节点必须有足够的容量，可以装载整个模型。这对大模型而言往往是不可实现的。因此现实中，大模型的训练部署往往采用混合方式。例如将整个 GPU 集群以数据并行的方式分成若干块，每块装入一个大模型。块内按照任务并行的方式，将大模型分解成若干与每块 GPU 容量匹配的子任务，每块 GPU 对应一个子任务，可以是大模型不同的网络组件，甚至可以大 Tensor 分成多个小 Tensor 进行并行计算。如果设计合理，还可以做到不同网络组件的流水线并行，通过一种“接力”的方式并行提高计算效率。

即使目前业界已有的 GPU 分布式训练方案，也严重依赖于服务器之间的通信、拓扑、模型并行、流水并行等底层问题的解决情况。如果只有分布式训练框架，甚至都无法正常启动训练过程。这也是为什么 GPT-3 已经发布了多年，却只有少数企业可以复现 GPT-3。目前，已经公布明确已经完成千亿参数规模大模型训练的框架主要是 NVIDIA 开发的 Megatron-LM 、经过微软深度定制开发的 DeepSpeed、国产百度飞浆 PaddlePaddle 和华为昇思 MindSpore。大多数并行框架都支持 PyTorch 分布式训练，可以完成百亿参数规模的模型训练。

3.2 并行计算框架

PyTorch[1]

PyTorch 自身提供了几种加速分布数据并行的技术，包括分桶梯度

[1] https://pytorch.org/

（bucketing gradients）、通信和计算的重叠（overlapping computation with communication）以及在梯度累积（gradient accumulation）阶段跳过梯度同步（skipping gradient synchronization）。PyTorch 分布式数据并行可以用 256 个 GPU 达到接近线性的可扩展性程度[36]。在 DP 的基础上，原生支持 DDP，每个节点都有自己的本地模型副本和本地优化器，支持多机多卡的分布式训练。一般来说，DDP 都显著快于 DP，能达到略低于卡数的加速比，但要求每块 GPU 卡都能装载完整输入维度的参数集合。在 1.11 版本后，PyTorch 开始支持 FSDP 技术，可以更加高效的将部分使用完毕的参数移至内存中，显著减小了显存的峰值占用，更加吻合大模型的特性。

TensorFlow[2]

TensorFlow 是一个为大规模数值计算设计的流行开源库。TensorFlow 支持异构设备的并行计算，可以在不同类型和尺寸的机器上运行，无论是超级计算机，还是嵌入式系统。它希望用户只需关注模型的构建和优化，透明化复杂的并行计算细节。此外，TensorFlow 可以实现多机并行线性加速，提高分布式训练的效率。原始的 TensorFlow 是基于静态图设计的，有着更高的底层运行效率。但缺点是不够灵活。最新版本的 TensorFlow 已经开始同时支持静态图和动态图了，是开源时间较长的并行框架。

飞桨 PaddlePaddle[3]

飞桨（PaddlePaddle，Parallel Distributed Deep Learning）是我国较早开源开放、自主研发、功能完备的产业级深度学习框架。飞桨不仅在业内最早支持了万亿级稀疏参数模型的训练能力，而且近期又创新性的提出了 4D 混合并行策略，以训练千亿级稠密参数模型，可以说分布式训练是飞桨最具特色的技术之一。

飞桨的分布式训练技术在对外提供之前就已经在百度内部广泛应用，如搜索引擎、信息流推荐、百度翻译、百度地图、好看视频、文心 ERNIE 等等，既包含网络复杂、稠密参数特点的计算机视觉（CV）自然语言处理（NLP）模型训练场景，又覆盖了有着庞大的 Embedding 层模型和超大数据量的推荐搜索训练场景。

昇思 MindSpore[4]

昇思 (MindSpore) 是一个全场景深度学习框架，旨在实现易开发、高效执行、全场景覆盖三大目标。其中易开发表现为 API 友好、调试难度低，高

[2]https://www.tensorflow.org/

[3]https://www.paddlepaddle.org.cn/

[4]https://www.mindspore.cn/

效执行包括计算效率、数据预处理效率和分布式训练效率，全场景则指框架同时支持云、边缘以及端侧场景。昇思 MindSpore 的特性之一就是融合了数据并行、模型并行和混合并行，构建一种易用高效的分布式并行训练模式，让算法人员不再需要关注算法模型到底需要用哪种模式训练。可以简化分布式并行编程，串行代码实现分布式训练，对用户屏蔽并行细节，并且保持高性能；计算逻辑上保持和单卡串行流程一致；实现上统一数据并行和模型并行，一套框架支持多种并行模式；结合集群拓扑优化性能。

OneFlow[5]

OneFlow 是国内较早发布的并行计算框架，一直主打分布式和高性能多机多卡训练。OneFlow 会把整个分布式集群逻辑抽象成为一个超级设备，用户可以从逻辑视角使用超级设备。最新版本的 OneFlow 和 TensorFlow 一样，实现了同时对动态图和静态图的支持，而且动静图之间转换十分方便。此外，OneFlow 完全兼容 PyTorch，将 PyTorch 程序转移至 OneFlow 框架的代价较低。OneFlow 支持数据 + 模型的混合并行方式，便于提升并行计算性能。OneFlow 在框架层面也做了大量优化，nn.Graph 提供了简洁、丰富的性能优化选项，如算子融合（Kernel Fusion）、自动混合精度训练 (Auto Mixed Precision Training) 等。

夸父 Colossal-AI[6]

“夸父”（Colossal-AI），提供了一系列并行组件，通过多维并行、大规模优化器、自适应任务调度、消除冗余内存等优化方式，提升并行训练效率，并解耦了系统优化与上层应用框架、下层硬件和编译器，易于扩展和使用。提升人工智能训练效率的同时最小化训练成本。在三方面进行了优化：优化任务调度、消除冗余内存、降低能量损耗[37]。夸父从大模型实际训练部署过程中的性价比角度出发，力求易用性，无需用户学习繁杂的分布式系统知识，也避免了复杂的代码修改。仅需要极少量的改动，便可以使用夸父将已有的单机 PyTorch 代码快速扩展到并行计算机集群上，无需关心并行编程细节。

Megatron-LM[7]

Megratron 是 NVIDIA 提出的一种基于 PyTorch 分布式训练大规模语言模型的架构，用于训练基于 Transformer 架构的巨型语言模型。针对 Transformer 进行了专门的优化，主要采用的是模型并行的方案。Megatron 设计就是为了支持超大的 Transformer 模型的训练的，因此它不仅支持传统

[5]https://oneflow.ai/

[6]https://www.luchentech.com/

[7]https://developer.nvidia.com/nemo/megatron

分布式训练的数据并行，也支持模型并行，包括 Tensor 并行和 Pipeline 并行两种模型并行方式。NVIDIA 发表了多篇论文，较有代表性的有发表于 2019 年 9 月的论文，主要提出了通过将矩阵分块提高并行度的方法[38]。发表于 2021 年 4 月的第二篇论文, 对于分布式中的一些重要的设计, 如 tensor parallel、pipeline parallel、micro batch size 等进行了一些分析与讨论。同时提出了更加精细的 pipeline 结构与 communication 模式[35]。通过多种并行方式的结合，可以让大模型的训练更快。发表于 2022 年 7 月的第三篇文章将核心操作 LayerNorm 和 Dropout 安装输入维度进一步切分，使得这两个需要频繁运行的操作在不大幅增加通信开销的情况下实现了并行[39]。

DeepSpeed[8]

在 2021 年 2 月份，微软发布了一款名为 DeepSpeed 的超大规模模型训练工具，其中包含了一种新的显存优化技术——零冗余优化器（Zero Redundancy Optimizer，ZeRO)[40]。该技术去除了在分布式数据并行训练过程中存储的大量冗余信息，从而极大地推进了大模型训练的能力。从这个角度出发，微软陆续发布了 ZeRO-1，ZeRO-2，ZeRO-3 和 ZeRO-3 Offload，基本实现了 GPU 规模和模型性能的线性增长[41]。基于 DeepSpeed，微软开发了具有 170 亿参数的自然语言生成模型，名为 Turing-NLG。2021 年 5 月，推出了能够支持训练 2000 亿级别参数规模的 ZeRO-2。目前最新版本 ZeRO-3 Offload 可以实现在 512 颗 V100 上训练万亿参数规模的大模型。

Horovod[9]

Horovod 是一个基于 TensorFlow, Keras, PyTorch 以及 Apache MXNet 的并行计算框架。Horovod 力求将单机程序快速简单地转化并行计算。由 LF AI & Data Foundation 基金会（LF AI and Data）维护。鼓励所有致力于人工智能、机器和深度学习的公司，参与到开源项目社区。Horovod 使用的 MPI 模型比 TensorFlow 的参数服务器模型更简单。使用 Horovod 编写的深度学习模型训练脚本可以在几乎不进行任何改动的情况下顺利地在单个 GPU、多个 GPU 甚至多个主机上运行。实验表明在拥有 128 个节点（每个节点 4 块 Pascal GPU）的集群上，在 Inception V3 和 ResNet-101 两个任务上，Horovod 几乎表现出了线性加速比。

表 3.1对以上并行框架的信息进行了汇总。

[8]https://github.com/microsoft/DeepSpeed

[9]https://github.com/horovod/horovod

表 3.1: 并行框架信息汇总

序号	框架名称	发布公司	更新方式	最新版本
1	PyTorch	Meta	开源社区	1.13.1
2	TensorFlow	Google	开源社区	2.11.0
3	飞桨	百度	开源社区	2.4
4	MindSpore	华为	开源社区	2.0.0
5	OneFlow	一流科技	开源社区	0.9.0
6	夸父	潞晨科技	开源社区	0.2.5
7	Megatron	NVIDIA	开源社区	3.0
8	DeepSpeed	MicroSoft	开源社区	0.8.1
9	Horovod	LFAI&Data	开源社区	0.27.0

3.3 模型部署

模型部署是决定大模型能否使用的关键因素之一，大模型因模型参数量大，对软硬件资源的配置有很高的要求。这一节将首先介绍部署大规模预训练模型面临的困难与挑战，以及常用的解决方案。

3.3.1 预训练模型部署的困难

大规模预训练模型已经成为深度学习应用中最常用的技术之一。尽管它们在计算机视觉、自然语言处理和语音处理等领域中表现出色，但将它们部署到生产环境中仍然面临许多问题的挑战，包括以下几个方面：

1. 模型大小：预训练模型通常非常庞大，GPT-3 等模型包含上千亿个参数，因此在部署时可能会面临存储和传输上的困难

2. 推理速度：模型推理速度是评估一个机器学习模型性能的重要指标之一。在实际应用中，模型的推理速度往往直接影响着用户的体验和系统的效率。高效的模型推理速度可以缩短处理时间，提高用户满意度，减少计算资源的浪费。

3. 计算资源：预训练模型需要大量的计算资源来进行推理，这可能会导致部署时的计算瓶颈和性能问题。

4. 硬件兼容性：预训练模型的部署需要适应多种不同的硬件平台，包括 CPU、GPU、ASIC 等，因此需要适配和优化。

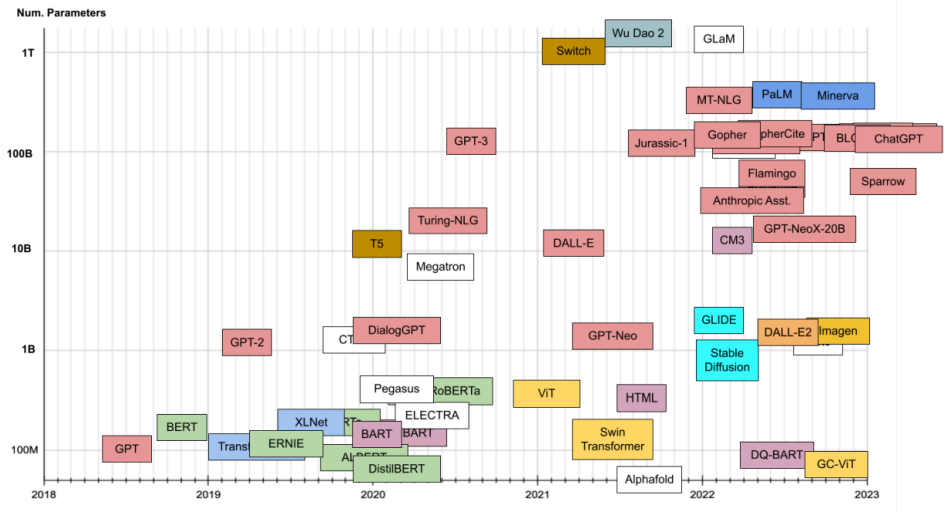

图 3.1: 近年大模型的参数规模增长趋势
图片来源:[7]

5. 数据隐私：预训练模型通常需要使用大量的数据进行训练，在部署时需要考虑数据隐私和保护的问题。

6. 版本管理：预训练模型可能会不断更新和改进，因此在部署时需要考虑版本管理和更新的问题。

3.3.2 部署框架和部署工具

部署流程 大模型的部署一般包括以下步骤：

1. 模型选择：选择一个适合自己业务需求的预训练模型，训练一个模型的参数。

2. 模型转换和优化：由于不同的深度学习框架和硬件设备之间存在差异，需要将权重文件转换为目标框架和设备可用的格式，同时进行一定的优化操作，以提高模型的性能和速度。

3. 数据预处理和集成：根据业务需求，对输入数据进行预处理和格式转换，将其转换为模型可接受的格式，并将模型与数据处理代码集成到一个可执行的应用程序中。

4. 模型部署和测试：将打包好的应用程序部署到目标设备上，并进行测试和验证，确保模型的正确性和稳定性。

5. 模型更新和维护：根据实际使用情况，对模型进行更新和优化，并及时修复可能出现的问题和 bug，保证模型的持续可用性和性能。

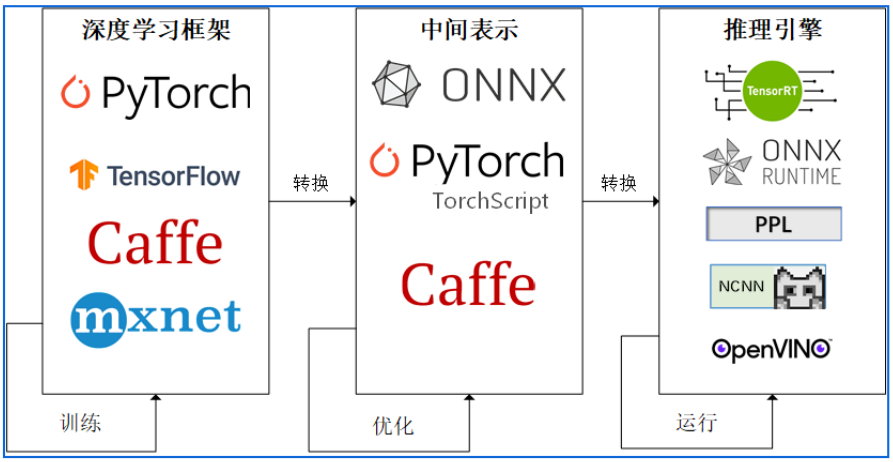

图 3.2: 模型部署流水线[10]

部署框架 选择合适的部署框架和工具来简化部署过程，并提供模型管理、调试和监控功能。一些常见的部署框架和工具包括 TensorFlow Serving、ONNX Runtime、OpenVINO、TensorRT、TorchScript 等。

[10]https://github.com/open-mmlab/mmdeploy/blob/master/docs/zh_cn/tutorial/01_introduction_to_model_deployment.md

部署方式 Online 方式：首先在移动端做初步预处理，然后把数据传到服务器进行预测后返回移动端。Offline 方式：根据硬件的性能选择模型，在服务器训练得到模型，在移动端进行预测的过程。

3.3.3 部署技术和优化方法

代码优化 代码优化是一种通过优化神经网络中的算子实现高效部署的技术。在预训练模型中，算子是指网络中的基本计算单元，通常包括全连接、卷积、池化、归一化等操作。这些算子的优化对于提高模型的效率和性能至关重要。一般来讲，算子代码优化可以通过以下方式实现：

1. 使用高效的算法：在实现算子时，使用高效的算法可以减少计算复杂度和内存占用，从而提高神经网络的性能。
2. 使用更高效的语言实现算子：例如使用 C++ 和 C 等来替代 Python 实现算子，可以更好地利用计算资源和硬件加速器，提高神经网络的性能。

硬件加速 硬件加速是一种通过使用专用硬件来提高神经网络性能的技术。通常情况下，硬件加速可以通过以下方式实现：

1. TPU（Tensor Processing Unit）：TPU 是由 Google 设计的专门为深度学习应用优化的 ASIC 芯片。与通用的 CPU 和 GPU 不同，TPU 专门针对深度学习计算的特殊需求进行了设计和优化。
2. ASIC（Application-Specific Integrated Circuit）加速：ASIC 是一种定制化的集成电路，专门为某个特定应用场景而设计制造。与通用的处理器和逻辑电路不同，ASIC 可以实现高度优化的电路结构和算法，以提高性能和能效。
3. FPGA（Field-Programmable Gate Array）加速：FPGA 是一种可编程逻辑芯片，它可以通过编程方式实现各种逻辑电路。与固定功能的集成电路（ASIC）不同，FPGA 具有高度灵活性和可编程性，可以在硬件层面实现不同的应用场景。FPGA 通常由大量的逻辑单元（Look-Up Tables，LUTs）和存储单元（Flip-Flops）组成。逻辑单元可以实现基本的布尔逻辑运算和算术运算，而存储单元可以存储中间结果和状态变量。FPGA 还包含了大量的内部通信线路和 I/O 引脚，可以与其他电路和设备进行通信。

云服务 云服务是指将预训练模型部署到云端服务器上，通过互联网提供给用户使用的服务。云服务可以提供强大的计算能力和存储资源，同时可以根据实际需要灵活调整计算资源的规模和配置。常见的云服务提供商包括 AWS、Azure、Google Cloud 等，它们提供了各种深度学习服务和工具，如模型训练、模型部署、自动缩放。

1. 模型训练服务：提供 GPU 和 TPU 等硬件资源和深度学习框架，可以帮助用户在云端快速训练深度学习模型。

2. 模型部署服务：提供各种深度学习模型的部署服务，可以将训练好的模型部署到云端或边缘设备上，以提供各种应用程序的服务。

3. 弹性伸缩服务：根据用户的需求和流量变化，自动调整计算和存储资源的规模和配置，以提供更加灵活、高效和可靠的服务。

移动端 CPU 推理框架的优化 移动端 CPU 推理框架的优化通常通过编译优化来实现，即通过对代码进行优化和重组，以便让 CPU 能更高效地处理模型计算，提高模型推理的速度。

隐私保护 随着机器学习在越来越多的场景中被应用，保护用户隐私已经成为一个重要的问题。在预训练模型部署中，也需要考虑如何保护用户隐私，常用的用户隐私隐私保护技术包括：

1. 差分隐私：通过添加噪声来隐藏数据集中的个人信息，从而保护用户的隐私。

2. 加密技术：加密技术是一种保护数据隐私和保密性的技术，它通过使用密钥来将原始数据转换为一种无法读取的形式。只有拥有正确密钥的人才能够解密数据并访问原始信息。

3. 访问控制：访问控制可以限制对数据和模型的访问，从而保护数据和模型的隐私。

3.4 预训练模型的压缩

3.4.1 模型压缩方案概述

随着深度学习技术的不断发展,大型语言模型(Large Language Model)[4, 23, 42, 43]已成为自然语言处理领域的核心技术。然而，这些模型通常具有数十亿乃至上百亿参数，导致存储和计算成本极高，大多数下游用户难以进行微调。因此，针对大型语言模型的模型压缩成为一种可行的替代方案，便于进一步部署[44]。针对于模型压缩，常使用的方案有以下几种：

1. 剪枝：剪枝是一种通过去除模型中一些不必要的连接或神经元来减小模型大小的技术。
2. 蒸馏：蒸馏是一种通过使用学生模型来模拟预训练教师模型的行为来减小模型大小的技术。通常情况下，学生模型由更小的神经网络或线性模型组成。
3. 量化：量化是一种将预训练模型中的权重从浮点数转换为低位数的技术。通常情况下，量化的精度是 8 位或更低。量化可以大大减少模型的存储空间和计算量，但可能会对模型的性能产生一定的影响。
4. 权重矩阵分解：使用包括 SVD 等矩阵分解方法对预训练模型的 FFN 层的权重矩阵进行分解，从而减少 Attention 层的参数量，提高模型的效率。
5. 模型参数共享：以 ALBERT 为例，模型的 Attention 层之间采用了权重共享的方式，从而减少了模型的参数量[28]。

在现有资源条件的限制下，模型压缩一般是面向具体的下游任务，即在微调阶段通过压缩模型的规模实现模型的下游任务快速适配。本文即重点探讨两种常用的针对下游任务微调的模型压缩方法：知识蒸馏[45] 和模型剪枝[46]。

3.4.2 结构化模型压缩策略

传统的知识蒸馏方法通过对齐模型输出或内部隐层特征，将复杂的“教师模型”知识迁移到较小的“学生模型”以实现模型压缩[44, 45, 47]。然而，当前大型语言模型（如 GPT-3 系列[4]）只提供 API 接口，其参数等处于黑盒

状态，难以应用传统知识蒸馏方法。相反，我们可以通过上传下游任务相关数据并利用教师大模型的输出信息和原始数据一起训练小模型，以使其具有一定的能力。例如，GPT-3 可以利用思维链解决复杂推理问题，而小模型由于大小限制而缺乏这种能力。研究人员通过让 GPT-3 输出中间推理步骤并生成多样的思维链步骤，丰富微调训练数据，并将这些思维链用于小模型的训练，使其具有一定的推理能力[48, 49]。除了未开源模型参数的黑盒模型，如 GPT-3，还有一些开源参数的大型语言模型，例如 OPT[42]，BLOOM[50] 等。对于这些模型，我们可以借鉴之前利用中间层特征的方法进行知识蒸馏。但是由于这些模型的参数仍然过大，对于一般下游用户的微调训练仍然是巨大的开销。因此，在进行大型语言模型的知识蒸馏时，不仅要关注小模型在推理阶段的性能和开销，还要关注蒸馏过程中的训练开销。

3.4.3 非结构化模型压缩策略

研究人员主要是围绕已经开源参数的模型，例如 GPT-J[4]、OPT[42] 以及 BLOOM[50] 等进行模型参数的剪枝。模型剪枝大体上可以针对具体关注的参数单元和子网络情况，可以分为结构化剪枝和非结构化剪枝两种。结构化剪枝方法在较高稀疏度的情况下可以达到可观的提速但是会带来一定程度上的性能下降，而非结构化剪枝的方法虽然可以在较高稀疏度的情况下保持性能，但是又难以在通用的硬件上带来实质性的加速[51]。同时，在之前研究中常用的迭代式剪枝策略并不是完全合适，因为仍然需要多次训练大模型，也会给下游用户带来较大的训练开销，因而如何 One-shot 地得到一个合适的子网络供下游用户使用值得探索。同时，研究人员还在探索如何将剪枝与其他模型压缩技术，如量化和蒸馏，相结合以进一步提高大型语言模型性能和效率。这些技术的发展有望为推动人工智能技术的发展和应用提供有力支持。

3.4.4 模型压缩小结

目前，针对像 GPT-3[4] 这样的超大规模模型进行有效的模型压缩仍然存在一些挑战。这些挑战主要包括以下几个方面：

模型复杂度：超大模型通常拥有数十亿甚至数百亿的参数，导致整个压缩过程的训练的计算量和内存消耗巨大，这对硬件要求非常高。超大模型的结构往往非常复杂，由多个层和子网络组成。因此，压缩模型的过程需要考虑如何剪枝模型、量化模型、知识蒸馏等多种技术手段的结合使用。

模型压缩技术的局限性：当前已有的模型压缩技术可能无法直接适用于超大模型。例如，传统的知识蒸馏方法可能无法有效地提取超大模型中的知识，而结构化剪枝等方法在较高稀疏度的情况下可能会带来性能下降。并且有研究表明，大型语言模型存在涌现能力，即当模型参数达到一定规模时才会具有足够强的能力。同时，由于超大模型的结构复杂，可能需要一些特殊的压缩技术来处理。因此，需要有一种通用的压缩方法，适用于各种类型的超大模型。

模型的黑盒特性：目前的超大模型如 GPT-3 等均为闭源模型，用户无法获取其具体的参数信息和结构信息。这使得在对模型进行压缩时需要使用一些基于模型输出或中间层特征的方法进行知识迁移和蒸馏，增加了压缩的难度。

针对超大模型的压缩，目前已经有一些研究在进行探索。例如，通过结合剪枝、知识蒸馏和参数共享等多种技术手段，可以在不损失模型性能的情况下将模型参数量压缩数百倍甚至数千倍。

总的来说，针对超大模型的压缩是一个具有挑战性的任务，需要结合多种技术手段进行综合处理。未来的研究将继续探索如何针对超大模型进行更加高效和精确的压缩，以推动人工智能技术的发展和应用。

第四章　ChatGPT 相关数据集

众所周知，算法、数据、算力是大模型时代的三方面重要因素。根据 OpenAI 前期论文[16] 和博客[1]介绍，ChatGPT 中数据集的规模和构建质量均高于以往的人工标注数据集。由此可见，在以 ChatGPT 为代表的大模型的训练中，数据集的收集、清洗和标注异常重要。本章将从预训练数据集以及人工标注的精调数据集两方面，详细介绍 ChatGPT 等模型所需的相关数据集。

4.1　预训练数据集

4.1.1　文本预训练数据集

ChatGPT 之所以展现出非常优秀的文本理解与生成能力，其中重要的因素是其拥有一个强大的基座模型。为了获得这样基座模型，需要在大规模无标注文本数据上进行预训练，目前被广泛使用的预训练数据集主要包括 BookCorpus、Wikipedia、Common Crawl、ROOT 等，表 4.1概览了目前常用的预训练数据集，具体情况如下所示：

BookCorpus　原始的 BookCorpus[52] 是一个由未出版作者撰写的免费小说集,其中包含 11,038 本书(约 7,400 万个句子,1G 个单词),共有 16 个不同的子类型 (如浪漫、历史、冒险等)。由于原始的 BookCorpus 数据作者已不再提供下载, Shawn Presser 在 2020 年整理发布了一套替代数据集, 其中, books1 中包含 18,000 册图书，约为 2.2GB，下载地址为https://hyper.ai/datasets/13642；books3 中包含 196,640 册图书，约为 37GB，下载地址为https://the-eye.eu/public/AI/pile_preliminary_components/books3.tar.gz。

[1]https://openai.com/blog

Wikipedia 维基百科[2]是一个免费的多语言协作在线百科全书，由维基媒体基金会运营，由超过 30 万名志愿者组成的社区编写和维护。截至 2023 年 2 月 22 日，英文版维基百科中有超过 662 万篇文章，包含超 42 亿个词。相较于其他语料，维基百科中的文本以严格引用的说明性文字形式写成，并且跨越多种语言和领域，数据质量非常高。

Common Crawl 系列语料库 Common Crawl 提供的网络存档包含了自 2011 年以来的网络爬虫数据集，包括原始网页数据、元数据提取和文本提取，规模超过千兆位 Token（PB 级）。同时，每月对全网进行爬取还会增加大约 20TB 的数据。Common Crawl 数据存储于 Amazon Web Services 和全球多个学术平台上，数据可以从https://commoncrawl.org/ 中获取。在 Common Crawl 数据集的基础上，又衍生出一系列数据集，包括 800GB 的 C4 数据集，38TB 的 mC4 数据集（下载地址https://www.tensorflow.org/datasets/catalog/c4）以及 CC-100 数据集（下载地址为https://data.statmt.org/cc-100/）。

ROOT ROOT 数据集是由 BigScience 开源的 1.6TB 预训练数据，包括了 69 种语言，可以用来训练超过 170B 参数的模型，数据可以从https://huggingface.co/bigscience-data 下载。

The Pile The Pile[53] 是专为预训练大规模练语言模型设计的英文数据集，数据规模为 825GB，整合了 22 个来源的数据，包括：PubMed Central、ArXiv、GitHub、the FreeLaw Project、Stack Exchange、the US Patent and Trademark Office、PubMed、Ubuntu IRC、HackerNews、YouTube、PhilPapers 和 NIH ExPorter。该数据集已被用于训练包括 GPT-J、GPT-NeoX-20B 在内的多种模型。数据下载地址为https://pile.eleuther.ai/。

悟道 悟道数据集[54] 是由北京智源人工智能研究院从 8.22 亿个网页收集的 3TB 中文语料库，是目前最大的中文预训练数据集。而且在构建这一数据集过程中，研究者为了更好地保护个人信息，删除了数据中所有的个人数据。数据下载地址为https://data.baai.ac.cn/details/WuDaoCorporaText。

[2]https://wikipedia.org/

CLUECorpus2020 CLUECorpus2020[55] 是由 CLUE 开源社区从 2019 年 7 月至 12 月的 Common Crawl 中清洗筛选出 100GB 的高质量中文预训练语料。数据的下载地址为https://github.com/CLUEbenchmark/CLUECorpus2020。

MNBVC 超大规模中文语料 MNBVC (Massive Never-ending BT Vast Chinese corpus) 包括新闻、作文、小说、书籍、杂志、论文、台词、帖子、wiki、古诗、歌词、商品介绍、笑话、糗事、聊天记录等一切形式的纯文本中文数据。除了包括主流文化，也包括各种小众文化甚至火星文的数据，目前数据规模 2180.1GB（截止到 2023 年 2 月 24 日），数据下载地址为https://github.com/esbatmop/MNBVC。

文本预训练数据集讨论

尽管目前已经开源了很多的预训练数据，但在训练大规模预训练语言模型时，预训练数据依然是瓶颈，原因如下：(1) 开源的预训练数据或多或少存在噪音问题，特别是爬虫数据噪音问题严重，如何对预训练数据进行高质量地清洗和去重，是目前数据处理的核心与壁垒；(2) OpenAI 和 Google 使用的高质量预训练数据集是闭源，无法获得，例如 Google 公司训练 Chinchilla 中使用的 2.1TB 的书籍数据库、3.1TB 的 Github 数据，OpenAI 公司训练 GPT 3 中使用的 WebText2[3]、Books1、Books2 数据集。

4.1.2 代码预训练数据集

在对大模型做大规模评价之后，Liang et al.[56] 发现训练数据中含有代码的模型具有很强的语言推理能力。在对 OpenAI 的各个模型测试中，也发现代码预训练与 COT 表现息息相关。因此，在预训练时使用代码数据成为越来越多研究者的共识。代码预训练数据可以根据程序语言和自然语言是否同时出现分成单语数据和对齐数据。表格 4.2展示了一些常见数据集的基本信息，对于包含众多编程语言的数据集，表中仅列出编程语言的类别数，NL-PL 表示成对的自然语言和编程语言对齐数据。

单语数据 单语数据主要是指不包含自然语言，只有程序语言的数据，主要的应用场景有两种。第一种是用于学习只包含程序语言的单语任务，比如代

[3]有人尝试复刻 WebText 数据集，具体情况见https://skylion007.github.io/OpenWebTextCorpus/

表 4.1: 预训练数据集概览

数据集	发布者	规模	特点	支撑的语言模型
BookCorpus	Zhu et al. (2015)、Shawn Presser	book1: 2.2GB; book3: 37GB	英文小说集	GPT 系列、OPT、OPT-IML
Wikipedia	维基媒体基金会	21.23 GB	多语言高质量百科全书	GPT 系列、OPT、OPT-IML
Common Crawl	Common Crawl 团队	超过 PB	网页数据, 规模巨大	GPT 系列、T5、UL2、Flan-T5
ROOT	BigScience	1.6TB	包含 69 种语言	BLOOM、BLOOMZ、mT0
The Pile	Gao et al.(2020)	825GB	数据来源广泛, 多样性佳	GLM-130B、GPT-J、GPT-NeoX-20B、OPT、OPT-IML、GLM-130B
悟道	北京智源人工智能研究院	3TB	中文数据集	GLM-130B
CLUECorpus 2020	GLUE 开源社区	100GB	中文数据集	
MNBVC MNBVC	里屋社区	2.18TB	中文数据集	

表 4.2: 代码预训练数据集概览

数据集	数据源	程序语言	规模 (GB)	类型
CodeSearchNet (2019)	GitHub	Go,Java,JS,PHP Python,Ruby	17	NL-PL PL
CodeNet (2021)	AIZU,AtCoder	55	8	NL-PL
THEPILE (2021)	GitHub,ArXiv,...	-	825	NL PL
thestack	GitHub	30	3100	PL
BigQuery	GitHub	C/C++,Go,Java JS,Python	340	PL
BIGPYTHON (2022)	GitHub	Python	217	PL
CodeParrot	GitHub	Python	180	PL
GCPY (2022)	GitHub	Python	-	PL

码补全、代码重构等；第二种则是用于预训练阶段，来保证生成的程序语言是语法和逻辑正确的。

对齐数据 对齐数据由成对出现的自然语言描述和程序语言源码构成，该类数据主要搜集自各种编程相关的平台。例如有的数据来自于开源代码平台 GitHub[4]，程序语言是源代码，自然语言是代码中的注释、说明文档和提交的版本信息等。而以 StackOverflow[5]为代表的问答社区中的数据，自然语言是问题描述，程序语言大多以代码片段的形式出现在问题和回答中。

自动爬取或收集的原始数据往往存在信息量低和含有噪声的问题，回答中的代码片段可能与问题无关等。因此，在使用数据之前，必须基于一定规则仔细进行筛选[57, 61]。

4.2 人工标注数据规范及相关数据集

自 GPT-3 开始，大模型微调进入了提示数据微调和指令数据微调时代，本节我们通过对现有已知的模型微调过程来分析指令微调数据集的特点与

[4]https://github.com/

[5]https://stackoverflow.com/

构造过程。

4.2.1 指令微调工作流程及数据集构建方法

指令微调是在预训练语言模型的基础上，在多个已知任务上通过对训练数据集添加自然语言形式的指令进行微调，从而激活模型的各方面性能，提高模型对未知任务的泛化能力和与人类期待的一致性，使其可以在某个新任务上进行零样本推理。

指令微调数据集的构建流程通常需要遵循几个环节。1）人工构建或选择现有的自然语言处理基础训练数据集，例如自动问答任务的 CoQA[6]、阅读理解任务的 SQuAD[7]等；2）在基础训练数据集上人工构建训练任务，通常一个数据集可以生成一个到多个任务；3）为每个任务设计指令集合，将提示与训练样本的输入数据结合，目的是清晰明确地指导模型的学习方向，从而构成用于有监督学习的指令微调数据集。

GPT-3 时代的提示工程通常采用上下文语境提示样例，构建上下文提示模板或者完形填空式的提示模板，将各类目标任务转化为下一句预测或者提示信息补全任务，从而充分发挥了预训练模型的强大能力来更好地解决问题。但这也导致该方法在以零样本方式学习的非提示型任务（例如自动文摘、自然语言推理等）时，缺乏必要的提示信息导致模型效果与人类预期的不一致。与提示微调相比，指令微调需要打通任务壁垒，激发模型的更多领域的能力，具有极强的泛化能力，因此指令微调方法采用了指令式的提示，即直接将自然语言形式的指令信息标注在输入文本中，通过特殊的标记或格式来明确地指示模型应该生成哪些内容。

如图 4.3所示,在文本摘要生成任务中,可以在输入数据中添加提示:“Generate a summary of the following text:”，以指示模型在生成文本时应该包含文本摘要信息。提示的写作方式可以根据不同的任务和需求进行灵活设计，以最大化提高模型生成的效果。

4.2.2 常见的指令微调数据集

指令微调数据集通常建立在自然语言处理经典数据集基础上。Google 研究院及 Hugging Face 等机构提出的指令微调训练任务采用的自然语言处理数据集如表 4.4所示。

[6]https://stanfordnlp.github.io/coqa/

[7]https://rajpurkar.github.io/SQuAD-explorer/

表 4.3: 文本摘要生成任务提示构建示例

Prompt	Generate a summary of the following text
Input	According to a recent study published in the Journal of Hepatology, drinking coffee regularly may reduce the risk of liver cancer. The study followed more than 400,000 participants over a period of 10 years, and found that those who drank at least 3 cups of coffee per day had a 50% lower risk of developing liver cancer compared to those who drank less than one cup per day.
Output	A new study shows that coffee may reduce the risk of liver cancer

指令微调方法是在自然语言处理数据集的基础上添加指令信息，以激活模型的各方面性能。表 4.5为 InstructGPT 模型训练的部分测试集以及对模型能力提升的分析。

4.2.3 构建指令微调数据集的关键问题

1. 现有训练集规模及特点 指令微调数据集的构建一般遵循重提示质量、轻规模的原则。Google 研究院提出的 FLAN 模型[62] 训练时用到了 62 个自然语言处理数据集，并从每个数据集中随机抽取训练样本，最大样本量为 30k，同时为每个数据集设计了 10 个任务相关的指令模板[63]。由 Hugging Face 牵头，多家单位合作提出的 T0 模型[64] 采用了 171 个数据集，平均每个数据集设计了 11 个提示模板。FLAN-T5 的有监督微调数据约 12-15k 个样本，每个样本的提示由人工构建的提示集合中随机抽取。由此可以看出，各家研究的数据集构建重点均为提示集合的构建，其质量决定了最终指令微调数据集的质量。

T0、FLAN 与 FLAN-T5 模型采用人工设计提示模板的方式。ChatGPT 使用了两种不同的指令来源：一种是直接由标注人员或研究人员进行人工标注，另一种是通过 OpenAI 的交互社区的大众交互模式，获取指令集合。

任务的数量也是影响模型训练结果的重要因素。FLAN-T5 在 FLAN 的基础上对任务进行了进一步细化，采用了超过 1800 个任务。模型效果随着任务数据的增加而提升，但任务数量到达一定规模后模型性能提升并不明显[65]。

表 4.4: 按任务分类的自然语言处理经典数据集

任务	数据集
Natural language inference	ANLI, CB, MNLI, QNLI, SNLI, WNLI RTE
Reading comprehension	BoolQ, DROP, MultiRC, OBQA, SQuADv1, SQuADv2
Closed-book QA	ARC-easy, ARC-challenge, NQ, TriviaQA
Commonsense reasoning	COPA, HellaSwag, PiQA, StoryCloze
Sentiment analysis	IMDB, Sentiment140, SST-2, Yelp
Paraphrase detection	MRPC, QQP, Paws-X
Coreference resolution	DPR, Winogrande, WSC273
Reading comprehension with commonsense	CosmosQA, ReCoRD
Struct to text	CommonGen, DART, E2ENLG, WebNLG
Translation	En–Fr from WMT'14, En–Es from Paracrawl En–De, En–Tr, En–Cs, En–Fi, En–Ro, and En–Ru from WMT'16
Summarization	AESLC, CNN-DM, Gigaword, MultiNews, Newsroom, Samsum Xsum, AG News, Opinion Abstracts - Rotten Tomatoes Opinion Abstracts –iDebate, Wiki Lingua English
Conversational QA	QuAC, CoQA
Evaluating context-sentence word meanings	WiC
Question classification	TREC
Linguistic acceptability	CoLA
Paraphrase Identification	glue/mrpc, glue/qqp, paws/labeled final
Extractive QA	adversarial qa/dbidaf, adversarial qa/dbert, adversarial qa/droberta, duorc/SelfRC, duorc/ParaphraseRC, ropes squad v2, super glue/record, quoref, tydiqa
Multiple-Choice QA	cos e/v1.11, cosmos qa, dream, openbookqa/main, qasc, quail quarel, quartz, race/high, race/middle, sciq, social i qa, wiqa super glue/boolq, super glue/multirc, wiki hop/original, piqa
Topic Classification	ag news, dbpedia 14, trec
Word Sense Disambiguation	super glue/wic

表 4.5: InstructGPT 模型微调数据集

数据集	数据集规模	数据集特征	对模型能力的提升
cnn_dm_samples	2,354 篇新闻文章	自动摘要	提升模型的摘要提取能力
drop_samples	9,536	数学计算式问答	提升模型的答案生成能力。但多数给出的答案仅是答案的来源字段,缺少分析和计算
fr_to_en_samples	1,500 个法语/英语对	机器翻译	提升模型的翻译能力
quac_samples	7,306	阅读理解	提升模型的阅读理解和推理能力
real_toxicity_samples	99,442	情感计算	提升模型文本延续的能力,判断填充后的言论的毒性大小
squadv2_samples	11,873	推理	提升模型阅读理解能力
tldr_samples	2,500	推理/自动摘要	模型可以根据帖子进行推断/摘要核心信息
truthful_qa_samples	817 个简答问题	问答系统	通过理解问题来寻找正确的答案,判断答案的真实性和相关性。

2. 指令信息的标注策略 指令微调数据中，每条指令为自然语言语句或自然语言模板的形式，因此指令的设计首要应考虑的是任务相关性和清晰性，即指令的形式和内容应该尽量和对应的任务相关，同时保证指令语句能够最大程度上体现人类的真实意图，避免任何歧义以及模糊的情况。

此外，指令信息在一些需要逻辑推理的任务上，可以通过思维链 Chain-of-Thought（COT）[65] 来提高模型的表现能力。即在输出数据中增加推理步骤的描述（小样本 COT），并在指令中提示模型给出逐步求解的答案（例如 by reasoning step by step）。

在标注格式上，由于指令微调模型通常要实现多任务训练，因此需要为多个任务设计一致的输入/输出数据格式以保证多任务融合的训练。根据是否需要进行推理（COT）以及是否需要提供示例（小样本）可将指令微调数据集的样本格式统一为四种类型（如表 4.6所示）：

表 4.6: 指令微调数据集格式

	输入	输出
无 COT，零样本	指令 + 问题	答案
有 COT，零样本	指令 +COT 引导（by reasoning step by step）+ 问题	理由 + 答案
无 COT，小样本	指令 + 示例问题 + 示例问题回答 + 指令 + 问题	答案
有 COT，小样本	指令 +COT 引导 + 示例问题 + 示例问题理由 + 示例问题回答 + 指令 +COT 引导 + 问题	理由 + 答案

3. 数据标注的人为主观因素的影响分析 模型的指令微调过程是一个用人类意图来激发语言模型潜能的过程，但是人工构建指令微调数据的过程可能会受到各种主观因素的影响，主要包括：

- 由人类专家以及交互社区的客户提供的指令的质量和均衡性（领域覆盖度）；
- 生成标签数据的人工标注者的个人偏好和对指令的理解力层次不同，会导致指令的生成质量参差不齐；
- 在人类反馈的强化学习模型数据标注中，标注者的现场情绪的起伏、个人价值观的差异，会直接影响候选答案排序的公正性和普适性。参与训练过程的标注人员针对某一问题的理解和看法也许并不能与未来模型用户的主流观点一致。

此外，指令微调训练数据集的构建还应考虑下述问题：

- 人工标注的规模、时长需要标准化及量化，以便进行不同模型及不同训练策略（例如有监督学习策略和强化学习策略）的对照研究。
- 指令的标注要避免同质化现象。例如两个指令在句法形式上不同但在语义上是等价的，则指令集合的质量是有问题的。

第五章　大模型评价方法

对模型的评价对于自然语言处理的发展至关重要。不同于对传统模型的评价方式，对大规模语言模型的评价方式也有所不同，往往很难使用单一的评价指标对其进行评价。本章将从模型的评价方式和评价指标两个大的维度进行阐述，同时分别在每个维度上介绍具体的评价方式和方法等内容。

5.1　模型评价方式

5.1.1　人工评价

人工评价是指通过人工直接标注对话系统对上文回复的质量。例如，可以根据回复的相关性等指标对其进行打分，这样能够相对准确地评价出对话系统是否能够对上文做出符合目标的、高质量的回复。该方法得出的结论也将更接近于实际应用中的用户体验。

李克特量表　李克特量表（Likert Scale）[66] 是一种常用的评估量表，该表提出的目的是为了以具体的、可被用于研究的方式衡量“参与者对研究内容的看法”这一概念。李克特量表针对所研究的内容提供一系列表述，由参与者以分数的形式来进行评分，一般包括 5 个或 7 个级别，例如（1）强烈不同意、（2）不同意、（3）中立、（4）同意、（5）强烈同意。通过对所有评分的加和，可以得出一个总体评价得分，来评估参与者对该研究内容的态度。

李克特量表是一种可靠的、有效的评估工具，已被广泛应用于各种领域的研究中。因此，可以使用李克特量表进行语言模型的人工评价。首先，提供给评测员上文及对话系统的回复；然后，评测员根据上文和回复分别从不同指标，如相关性等进行评分；最后，分别计算语言模型得到的所有样例各个指标之和，即可评价出语言模型在各个指标上的整体性能和回复质量。该方法简单易用，评价不同语言模型时易于标准化、比较和分析，但该方法也

存在一些缺陷，如不同评测员对不同等级、不同指标的理解存在一定主观性偏颇等。

相对排序 如果直接要求参与者对待评估内容给出评估得分，那么得分的取值范围很容易受到人的主观因素影响，不同的参与者对同样质量的文本，可能会给出相差很大的评分结果。为了规避这一问题，可以采取相对排序（Relative Ranking）[67] 的方法，该方法通过直接对不同语言模型的回复进行比较得到最终结果，从而避免了不同评测员对不同等级、不同指标的理解存在主观性而带来的差异。

首先，将参与比较的语言模型针对同一输入的回复作为一组提供给评测员；然后，评测员依次通过两两比较得出哪一个语言模型的回复更好，直至可以得到针对该输入，每个模型的回复的最终排名；最后，根据不同语言模型在多轮评估后的平均排名比较不同对话系统的性能差异。当参与评价的对话系统较多时，可以每轮从所有系统中选择 5 组参与评价，并保证每个系统参与评价的轮数相同，来达到减少评价次数的目的。该方法可以避免由于评测员之间对指标理解的差异性、量表设计的缺陷而带来的问题，但也由于具体样例可能参与多次比较而导致评估成本增加等问题。

前端界面的影响 由于对话系统的人工评估往往采取众包的形式，因此评估系统的前端界面设计对评估结果也有很大影响。举例而言，如果对于某一对话质量属性的打分范围是 1 到 5，那么设置一个包含五个选项的下拉菜单，就不如设置一个 1 到 5 得分区间的滑条，这样参与人会对当前的打分区间有更深入的感知。再比如，如果前端界面将所有待评估内容罗列到一个密集的区间中，就很容易令参与人感到疲惫，导致评估精度的下降。在具体设计中，必须考虑评估人员和待评估内容的特点，有针对性地设计评估界面。

5.1.2 自动评价

自动评价是指通过自动方法使用评价模型生成对话系统对上文回复的评分，主要是借助统计评价指标或者评价模型，使用一些可量化的指标来评价对话系统的性能。虽然自动评价的精度和灵活性往往弱于人工评价，但是自动评价具有高效率、低成本、评价结果客观且可复现、能够处理多维度评估等优势。

统计评价指标 统计评价指标是借助统计特征设计公式，对生成的文本进行评估。目前被广泛应用的客观指标主要有三类:

1. 基于词重叠率的指标，比如来自机器翻译任务的 BLEU[68]、来自文本摘要任务的 METEOR[69] 和 ROUGE[70] 等，这一类评价指标衡量的是生成文本和参考回复之间的匹配程度，表征了回复的语义精确性。

2. 基于词向量的评价指标，比如 Greedy Matching（贪心匹配)[71]，Embedding Average (向量均值)[72], Vector Extrema (向量极值)[73] 等, 借助词向量的分布式表示能力，规避了 N-gram 字符串匹配中，无法考虑同义词和近义词的问题，进一步增强了统计评价的精确性。

3. PPL（困惑度)[74]，Distinct-1&2[75] 等其他评价指标，这些指标评估的不再是生成文本和参考回复之间的匹配程度，而是生成对话的流畅度、多样性等其他方面。

在对话系统中，回复的结果并没有单一的标准答案，合理的对话系统不仅要保证语义的正确性和流畅性，还应当保证内容的多样化，这样才能够产生可持续的对话流程，因此结合多个统计评价指标，才能够全面评估对话系统的质量。

基于模型的评价指标 随着神经网络的不断发展，研究学者们开始尝试使用神经网络进行对话评价, 即模拟评分的评价方法。2017 年麦吉尔大学的 Lowe 等人认为，沿用机器翻译的指标本身就不具备合理性，并提出了 ADEM 方法[76]，同时利用上下文和参考回复对模型生成回复进行评价，在训练的过程使用双层 RNN 结构得到向量表示，同时采用众包的方式请人对模型生成回复打分。2017 年北京大学的团队提出了新的评价指标 RUBER[77]。RUBER 结构分为两部分，第一部分为基于向量的评分，用于衡量生成回复与参考回复的相似性（有参考的)，第二部分为基于网络的评分，用于衡量生成回复与上文的相关性（无参考的)，并选择最优的混合方式给出了最终的评价方法。2020 年，Sarik Ghazarian 等人提出 Predictive Engagement[78]，他们认为只凭借相关性这一指标并不能全面的评价模型，参与度对于开放域对话系统来说是一个至关重要的指标，而目前评价模型对话层级（Conversation-level）的参与度主要使用对话轮数或对话时长这种启发式的指标，因此他们在话语层级（Utterance-level）提出一种参与度定量评价方式，同时也可以通过话语层级的分数预测对话层级的分数，进一步提升了评价性能。同年，康奈尔

大学团队提出 BERTScore[79], 想法是利用 BERT 得到生成回复和参考回复的词向量，进而计算两者相似度。

基于模型的评价指标，利用了深度学习，甚至是预训练模型的强大特征表征能力，因此在面向特定领域和方面的评价上也更加精准。但是由于这一类模型往往需要训练，因此也存在无法迁移、适应性差的问题。

5.2 模型评价指标

5.2.1 准确性

分类任务 自然语言理解任务的大部分，都可以归为分类问题，而精确率、召回率和 F1 值是用于判断分类任务匹配精确程度最常用的评价指标，广泛应用于文本分类、序列标注、信息检索等领域。

精确率（Precision）指的是模型正确预测为正例的样本数占预测为正例的样本总数的比例。精确率越高，表示模型预测出的结果中真实正例的比例越高。

召回率（Recall）指的是模型正确预测为正例的样本数占实际正例的样本总数的比例。召回率越高，表示模型越能够正确地捕捉到所有的正例。

F1 是精确率和召回率的调和平均数，反映了模型的综合性能。

生成任务 自然语言生成是自然语言处理中一个重要的研究领域，包含机器翻译、文本摘要、对话生成等多个任务。衡量一句话生成的好坏，无法简单用正确和错误来分类，而是包含多个层次、多个维度的评价，因此使用的指标也更加复杂。

对于机器翻译而言，通常使用 BLEU 值来衡量机器翻译质量的好坏，BLEU 值就是计算机器译文 N-gram 的精确度，根据参考译文来评价机器译文。

对于自动摘要而言,通常使用 ROUGE 值来衡量摘要质量的好坏。ROUGE 同样基于 N-gram 的匹配程度，由于文本摘要更多关心的是摘要内容是否覆盖完全，因此使用的是面向召回率的摘要评估指标。

近年来，随着深度学习和预训练模型的发展，评估准确性也出现了一些新的方案，但是通常而言，使用上述经典的统计指标进行准确性的评价，仍然是最普适和稳妥的方案。

5.2.2 不确定性

对话模型不一定每次都能给出准确的答案，在一些特殊的场景，例如医疗诊断一类的高风险应用场景下，我们不能只关心模型的准确度，还应该关注对话模型给出的结果有多大程度的确定性，如果不确定性太高，就需要谨慎决定。不确定性可分为两种：

偶然不确定性 又称数据不确定性，指数据中内在的噪声，即无法避免的误差。通过获取更多的数据是无法降低偶然不确定性的，降低偶然不确定性的可行办法主要包括提高数据精度和对数据进行降噪处理两种。

认知不确定性 又称模型不确定性，指模型自身对输入数据的估计可能因为训练不佳、训练数据不足等原因而不准确，与某一单独数据无关。认知不确定性可以通过增加训练数据的数量等方式来降低甚至解决。

一般来讲，对话模型的不确定性可以通过置信度来反映，置信度越高，不确定性越低。对于一个优秀的模型，其准确率应该和置信度相匹配，为了衡量这一匹配程度，一个常用的评价指标便是期望校准误差（ECE）[80]。该指标通过计算各个置信区间中样本的平均置信度和准确率差值的期望，来对模型的优秀与否进行评估。

5.2.3 攻击性

在大量真实人类对话语料数据上训练得到的模型在测试场景可能会面临数据分布及特征不一致的情况，大量的研究证明，人类在与对话机器交流时往往会更加具有攻击性，并且会使用许多暗示以诱导模型生成不安全的内容。此外，基于大规模语料学习的语言模型也会学习到特定语料间潜在的关联，而这些关联往往高频出现在毒害内容中。

对话系统的攻击性评价，作为一种评价方法，是在实时交互中诱导对话系统犯错。根据输入上文诱导方向的不同，它可以评价系统的安全性、公平性和鲁棒性等许多方面。比如我们可以通过收集已有的人类用户“攻击”某个对话系统的上文，测试现有系统的安全性、公平性；我们同样可以使用对抗攻击方式，微调输入上文，观察对于系统输出的影响，从而评价其鲁棒性。

模型在诱导提示下的表现评价 Gehman et al.[81] 研究了预训练语言模型在多大程度上会被诱导产生有危害的内容。作者从大型网络英文语料中提取

了约 10 万个真实人类语句的提示 (Prompt)，并构建了数据集 RealToxicityPrompts。虽然这些提示本身是没有危害的，但是实验结果表明，将其作为主流语言模型的输入后，输出的结果有很大的概率为毒害性文本。

模型在攻击提示下的表现评价 多语言鲁棒性评价工具[82] 采取了多种策略 (转换，对抗攻击等)，可以为输入数据根据各种增广策略来生成相应的变体，提供了对鲁棒性的全面评价，并在评价阶段中给出了可视化结果。Niu et al.[83] 提出使用对抗攻击方式评价对话模型，具体的，文章中提出两种策略评价对话模型的两种行为：过于敏感与过于固执。使用同义词替换等方式替换对话上文，模型输出的下文可能出现极大变化，即模型的过于敏感；而微小但改变了语义的对话上文，模型也可能输出与原来同样的内容，体现了其“固执”。

5.2.4 毒害性

对话模型需要能够妥善处理各式各样的对话场景并给出令人感到舒适的回复，包括冒犯性言论、辱骂、仇恨言论等[84]。对毒害内容的自动检测对语言模型输入输出内容的审核政策有着极大的帮助。特别值得注意的是，毒性检测的风险是非常高的，由毒性检测的失败而导致的内容审核失败会引发非常严重的社会问题，并对其广泛部署的可行性造成深远的影响。

基于分类的毒害性识别 早期的关键词检测方法会导致检测结果存在非常多的假阳样本，虽然很多的语句包含这些预先定义好的毒害关键词，但是本身句义是安全的。随着深度学习技术的发展，现在主流的做法是通过训练分类模型来判断整句句义是否为有毒害的，这样的一种方法突破了关键词库的限制，使得毒害性检测可以扩充到各式各样的检测场景中。

通过预训练方式得到的毒害性检测器虽然有着良好的性能，但在对抗性攻击输入下性能表现低下。Dinan et al.[85] 提出一种人在回路的方法（BBF）来增强评价模型的表现，通过人为地不断打破模型的识别边界，使得模型更加具有鲁棒性，其表现逐渐接近人类水平。

BBF 的方法仅考虑了言论检测场景，Xu et al.[86] 将这一过程扩充到了对话生成场景。同样采用了人在回路的方式，模拟了真实人机对话场景中人类的攻击性言论并得到机器回复，从而利用该对话数据训练并部署模型安全层，使得模型极大地减少了关于毒害内容的生成。

毒害性评价开放测试工具 目前已经有一些团队开放了接口用于评价语言模型潜在的危害。Jiqsaw 与谷歌泛滥用技术团队于 2022 年推出了新版本多语言检测接口 Perspective API[87]，模型采用一个多语言 BERT 模型及多个单语种 CNN 网络，可以对待检测言论给出在毒害性，侮辱，威胁等标签内容的可能性。Facebook 团队于 2022 年推出了测试工具 SafetyKit[88]，主要关注对话模型在三个方面的安全表现，来评价语言模型是否存在明显毒害性，具体包括三个方面：对话模型是否直接生成有害内容，对话模型对有害内容的回应是否合适，以及对话模型给出回复是否符合自身设定与属性。

5.2.5 公平性与偏见性

现有的大量案例表明，语言模型对待具有不同特征的个体与群体的数据上存在明显的差异性。这些明显的差异源于数据本身，并且模型在数据上训练的过程中没有规避这一潜在风险。通过评价语言模型的公平性和偏见水平，确保其在一个合理的范围内，可以发挥并体现出科学技术在社会发展变革中的积极作用，引领良好的社会风气。

最近，Liang et al.[56] 将衡量公平性的方式分为两类：反事实公平和性能差异。反事实公平通过对测试数据集进行目标特征的反事实增广，基于特定任务，评价模型对反事实数据的结果变动情况。反事实公平通过对数据进行扰动，提供了可操作性，并且适用于很多场景。性能差异则是通过预先确定好具有目标特征的数据样本，计算语言模型在这些待观察的数据组上的表现差异。

除此之外，类似公平性，对社会偏见的研究也是语言风险技术的核心。但不同的是，偏见往往描述的是一种内在的特性，与特定任务没有关系，体现在语言模型在语言选择上的倾向。几乎所有的数据集都存在偏见，并且目前对偏见也没有一个系统性的解决方案。

模型在人类层面的偏见水平的评价 May et al.[89] 提出了方法 SEAT 来衡量语言模型在二元性别方面的偏见水平，SEAT 通过预先定义好的两组性别属性词汇 (him, man,...; her, woman,...) 和一组检测目标词汇 (family, child, office,...)，以及用于合成句子的语句模板，通过语言模型得到合成句的上下文表示，通过计算目标句子与两组性别句子表示的相似度，来反应语言模型在性别上的偏见程度。

Nadeem et al.[90] 提出了 StereoSet 基准测试数据将测试目标拓展到了职业和种族方面。每一个测试数据都包含有空缺的语境句子和三个候选项，

分别对应刻板，非刻板印象，以及不相关三种关系。通过计算语言模型对每个示例在刻板联想和反刻板联想上的倾向进行评分，来量化语言模型的偏见程度。

标准/非裔美式英语数据集 (SAE/AAVE Pairs[91]) 包含了具有同等语义但是具有不同方言特征的美式英语对，用来更好的理解语言模型在方言上面的性能差异。为了评价语言模型，使用每条非裔美式英语的前面几个词用作语言模型的提示，通过人工评价和情感分类计算生成回复与原始回复的相似性。

模型在社会层面的偏见水平的评价 Abid et al.[92] 研究了语言模型在宗教层面的偏见，在其提出的测试数据集 MuslimBias 上，采用了补全提示和类比推理的方法。在补全提示中，用一个包含 Muslim 词汇的提示作为语言模型的输入，通过关键字匹配判断补全结果中是否使用暴力词语，并将结果与其他宗教团体作比较。类比推理测试中，将一个包含 Muslim 的类比句作为输入，并报告那些常用来完成类比的词汇的频率。

政治敏感话题仍然是语言模型面临的挑战，以负责任、无党派和安全的回复处理政治敏感内容对语言模型来说是不可或缺的。Bang et al.[93] 引入度量标准来评估对话机器人的政治谨慎性。方法考虑了两种场景，用户输入是中立的和有偏的 (倾斜的政治观点)，通过使用不同的政治属性词组合 (政治家-姓名、政治-主题、政治-信仰) 和语句模板分别生成了两个场景的模型输入。在测试阶段通过预训练好的 BERT 分类器对输出结果的政治立场进行打分。

5.2.6 鲁棒性

在部署测试阶段，语言模型面临着开放世界语言的复杂性与随机性 (如简写，错字等)，大多数在实验中表现良好的语言模型都会存在性能显著下降的问题。现实世界的数据包含不同类型的噪声，评价这些噪声对语言模型的输出结果的影响，对研究一个安全可靠的语言模型是非常必要的。此外，其他形式的鲁棒性也非常重要，但是在评价阶段需要对数据和模型有额外的处理流程，使得在评价阶段实现高效且精确的度量具有挑战性。例如，在评价基于分布的鲁棒性时，需要具有特殊构造的检验集 (将源域与目标域基于特征划分为不同的子域)[94]。而在评价对抗鲁棒性时，需要对语言模型进行多次对抗攻击，以不断地逼近其阈值 (扰动临界点)[56]。

无关扰动的稳定程度 基于转换/扰动的范式，即评估语言模型的输出在小的、语义保持的扰动下的稳定性，已被广泛用于研究模型的鲁棒性。由 Dhole et al.[95] 推出的自然语言数据集扩充工具 NL-Augmenter 可以实现这一过程。NL-Augmenter 将目标语句的“转换”从严格对等的逻辑中心观点放宽到更具描述性的语言学观点，通过容纳噪声，有意和无意的类人错误，社会语言层面变动，有效语言风格，语法变化等方法对原数据进行转变，极大地提高了原始数据集的多样性。最后通过语言模型输出结果中的不变比例反应模型的鲁棒性。

关键扰动的敏感程度 测试模型的鲁棒性目的是了解模型是否真正的捕捉到了语句中的关键信息而非某些不相关的次要联系。作为对微小扰动下不变性的补充，测试改变语义的扰动如何影响模型行为，可以了解模型对改变目标输出的扰动敏感程度，且不关注实例样本中的不相关部分。但困难的是，与生成保持不变的噪声不同，实现一个用于生成语义交替扰动 (目标输出也相应变动) 的通用方案具有更高的挑战性。

相关工作有 Contrast Sets[96]，作者基于数个当前 NLP 常见数据集的测试集 (视觉推理，阅读理解，情感分析等任务相关) 进行了扩充，对已有测试样本进行了微小 (保留原始样例中出现的任何词汇/语法信息) 但能改变标签结果的扰动。新的基准测试表明各 SOTA 模型在 Contrast Sets 上均表现不佳。

5.3 模型评价方法小结

本章介绍的模型评价方法主要从两个维度和多个角度（方式和指标）上对以 ChatGPT 为代表的模型的性能进行评价，其中评价方式上覆盖了三种可能的形式，但三者处在不同的阶段。人工评价是目前被广泛认可的全面且准确的评价方式，其在评价依靠的对象（评价者）和模型面向的使用对象之间具有较好的一致性。但是人工评价也面临着不同评价者的主观差异性、人力资源消耗较大、时效性较低等挑战，而人工和自动评价相结合的方式，能够在一定程度上缓解单纯依靠人工评价方式所带来的上述问题，但评价方式的最终目标为高效且稳定的自动评价方式。

在评价指标上，本章主要关注在模型本身的任务（语言建模）、面向真实场景使用时的稳定性和社会影响因素等方面的指标，而没有过多的介绍具体的计算方法，因为我们认为在本文中整理和归纳出全面的评价指标角度比

具体指标的计算方法更有意义，如果对计算指标感兴趣，可以参考[56]。

第六章　现有大模型及对话式通用人工智能系统

本章将对已有文本大模型、代码预训练模型以及类 ChatGPT 的对话式通用人工智能系统展开调研，客观地展现目前大模型以及对话式通用人工智能系统的发展现状。调研结果显示：（1）目前文本大模型与代码大模型发展日新月异，越来越多的模型可以通过 API 或者开源形式被访问，在这其中，OpenAI 与 Google 在文本大模型发展上占据先发优势，逐渐形成家族式大模型集；（2）目前面向普通群众，界面友好的对话式通用人工智能系统在数学能力、文本流畅性上较 ChatGPT 还有差距，但部分对话式通用人工智能系统可以在回复中加入引用，帮助用户追踪信息来源，这是目前 ChatGPT 所不具备的功能。

6.1　现有大模型对比

随着语言模型技术的快速发展，大模型已经成为各大互联网公司制造影响力的重要工具，各大公司相互竞争，相互启发，越来越多的大模型以 API 或者开源形式被访问到。表 6.1从参数量、输入长度限制、访问方式以及模型微调方式，全面对比现有较为知名的文本大规模预训练语言模型。

从表 6.1中，我们能观察到：（1）目前大部分文本大模型没有开源，外部只能通过 API 调用形式访问或者无法访问；（2）OpenAI 与 Google 在文本大模型发展上占据先发优势，掌握了主动权，逐渐形成家族式大模型集群。

代码预训练模型　除了文本大模型外, 研究人员在代码领域也提出了对应的预训练模型, 并且在代码任务上取得了优异的效果。表 6.2汇总了代码领域的预训练模型。

表 6.1: 大规模文本预训练模型对比表

发布者	模型	参数量	输入长度限制	访问方式	微调方式
OpenAI	davinci	175B?	2048	API	None
	text-davinci-001	175B?	2048	API	指令微调
	code-davinci-002	175B?	8000	API	
	text-davinci-002	175B?	4000	API	代码微调 + 指令微调
	text-davinci-003	175B?	4000	API	代码微调 + 指令微调 + RLHF
	text-curie-001	6.7B	2048	API	
	text-babbage-001	1.3B	2048	API	
	text-ada-001	350M	2048	API	
Google	LaMDA	137B		未开放	对话微调
	Flan-LaMDA	137B		未开放	指令微调
	T5	11B	512	开源	
	UL2	20B	512	开源	
	Flan-T5	11B	512	开源	指令微调
	PaLM	540B		未开放	
	U-PaLM	540B		未开放	
	Flan-PaLM	540B		未开放	指令微调
	Flan-U-PaLM	540B		未开放	指令微调
DeepMind	Sparrow			未开放	指令微调 + RLHF
	Chinchilla	70B		未开放	
	Gopher	280B	2048	未开放	
BigScience	BLOOM	175B	2048	开源	
	BLOOMZ	175B	2048	开源	指令微调
	T0pp	11B	1024	开源	指令微调
	mT0	11B	1024	开源	指令微调
Meta	OPT	175B	2048	开源	None
	OPT-IML	175B	2048	开源	指令微调
	Galactica	120B	2048	开源	指令微调
	LLaMA	65B	8192	开源	
微软/英伟达	TNLG v2	530B	2048	未开放	
Eleuther	GPT-J	6B	2048	开源	
	GPT-NeoX	20B	2048	开源	
Cohere	xlarge	52.4B	2048	API	
	large v20220720	13.1B	2048	API	
	medium v20220720	6.1B	2048	API	
	small v20220720	410M	2048	API	
AnthropicAI	Claude	52B	8192	未开放	
清华大学	GLM-130B	130B	2048	开源	指令微调
华盛顿大学 AI21 Labs	J1-Jumbo v1	178B	未知	API	
	J1-Grande v1	17B	未知	API	
	J1-Large v1	7.5B	未知	API	

表 6.2: 代码预训练模型对比表

发布者	预训练模型	参数量	输入长度限制	访问方式	预训练数据量
Google	CuBERT	-	1024	开源	6.6M
微软	CodeBERT	125M	512	开源	3.5G
微软	GraphCodeBERT	125M	512	开源	3.5G
微软	CodeGPT	124M	1024	开源	Python 1.1M Java 1.6M
Case Western Reserve University	CoTexT	-	1024	开源	-
Salesforce	CodeT5	60M/223M/ 770M	512	开源	8.35G
University of California Los Angeles Columbia University	PLBART	140M	512	开源	655G
Salesforce	CodeGen	350M/2.7B/ 6. 1B/16.1B	2048	开源	825G
Facebook	InCoder	1. 3B/6.7B	2048	开源	159G
OpenAI	Codex	300M/2.5B 12B	1024	API	159G
DeepMind	AlphaCode	300M/1B 3B/9B/41B	1536	未公开	715.1G
华为	PanGu-Coder	317M/2.6B	1024	未公开	147G
清华大学	CodeGeeX	13B	2048	未公开	-
aiXcoder	aiXcoder L	1.3B	-	未公开	-
aiXcoder	aiXcoder XL	13B	-	未公开	-

6.2 对话式通用人工智能系统调研

6.2.1 对话式通用人工智能系统

除了 OpenAI 推出的 ChatGPT，目前包括谷歌、Anthropic、百度等都推出或者正在开发类 ChatGPT 的对话式通用人工智能系统。具体情况如下:

谷歌的 Bard 2021 年 5 月，谷歌推出了面向对话的大语言模型 LaMDA。根据 2022 年初的官方论文介绍，LaMDA 模型参数达到了 137B，可以展现出接近人类水平的对话能力。Bard[1]构建于 LaMDA 模型的基础上，但为了扩展到更多的用户，使用了更轻量的版本。同时，相较于 ChatGPT，Bard 可以检索网页，从而能够回答有关最近发生事件的问题。目前，Bard 仅开放给受信任的测试人员。

Anthropic Anthropic[2]是一家由 OpenAI 前员工创建的初创公司，其自称“是一家人工智能安全和研究公司，致力于构建可靠、可解释和可操纵的人工智能系统”，开展了一系列大模型对齐、可解释性相关的研究。Anthropic 开发了一款名为 Claude 的智能聊天机器人，但尚未公开发布。

YouChat YouChat[3]是由 You.com 推出的聊天机器人，You.com 由语言和人工智能专家 Richard Socher 创立,2022 年 3 月,该公司推出了 YouWrite,这是一款基于 GPT-3 的可用于编写电子邮件和其他文档的文本生成器。2022 年 12 月，该公司推出了基于 GPT-3.5 的聊天机器人 YouChat，它会使用谷歌检索较为通用的结果，同时使用微软必应搜索来处理更细粒度的请求，如代码片段检索等。相较于 ChatGPT，YouChat 的回复中会带有引用，进而可以帮助用户追踪每条信息的来源。YouChat 的一个对话实例如图 6.1所示。

Perplexity AI 2022 年 12 月份，搜索引擎 Perplexity.AI[4]发布，其核心是将大规模语言模型和搜索引擎结合来进行问答，通过连续对话的形式提供用户需要的答案。相较于 ChatGPT，Perplexity AI 能够提供信息的来源

[1]官方网站为https://www.bradai.chat/

[2]官方网站为https://www.anthropic.com/

[3]YouChat 可以在https://you.com/ 中访问

[4]访问地址为https://www.perplexity.ai/

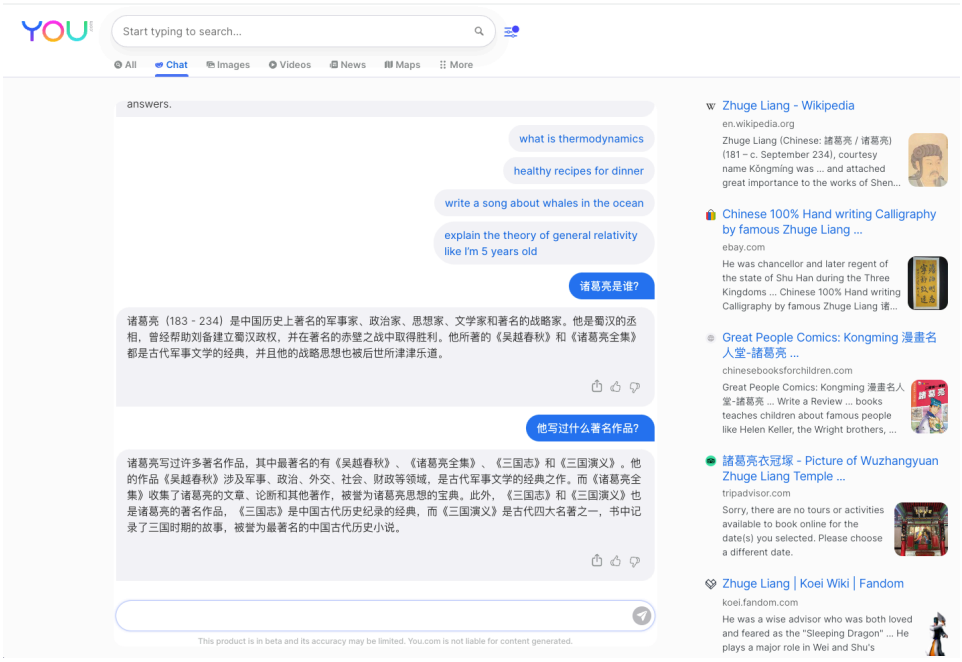

图 6.1: YouChat 对话示例

（包括维基百科、百度百科、知乎等），但其答案的流畅度和完整度相较于 ChatGPT 略显逊色。Perplexity AI 的一个对话示例如图 6.2所示。

MOSS 2023 年 2 月 20 日，复旦大学自然语言处理实验室推出了国内第一个对话式大型语言模型——MOSS[5]，可执行对话生成、编程、事实问答等一系列任务。

此外，国内也有多家企业和机构明确表态正在进行类 ChatGPT 模型研发，包括但不限于：

百度的文心一言 文心一言（英文名：ERNIE Bot）是百度基于文心大模型技术推出的生成式对话产品，据百度官方公众号的消息[6]，百度计划于 2023 年 3 月 16 日 14 时在北京总部召开新闻发布会，主题围绕文心一言。

科大讯飞学习机 科大讯飞表示，其生成式预训练大模型技术将率先落地于 AI 学习机产品，并将于 2023 年 5 月 6 日发布[7]。

[5]https://moss.fastnlp.top/

[6]https://mp.weixin.qq.com/s/nrvsHq_XBcEtzQxIEuTivQ

[7]https://finance.sina.com.cn/roll/2023-02-09/doc-imyfarws6247028.shtml

图 6.2: Perplexity AI 对话示例

京东 ChatJD 2023 年 2 月 10 日，京东宣布将推出产业版 ChatGPT：ChatJD[8]。

腾讯的混元助手 2023 年 2 月，腾讯针对类 ChatGPT 对话式产品已成立“混元助手（HunyuanAide)”项目组，致力于开发腾讯对话式通用人工智能系统[9]。

6.2.2 不同系统之间的比较

针对不同的对话式通用人工智能系统，我们分别用数学表达式计算（如表 6.3)、国家公务员考试数学应用题原题（如表 6.4)、违背常识的问题（如表 6.5）和自然语言任务指令（如表 6.6）进行了探测。

从探测结果上来看,在数学表达式计算上,ChatGPT 明显要优于 YouChat 和 Perplexity AI，不仅结果正确，而且中间步骤完全正确。在国家公务员考试数学应用题计算上，由于网上可以检索到该题目的答案，Perplexity.AI 回答完全正确，而 ChatGPT 和 YouChat 没有回答正确，值得注意的是，YouChat 也似乎检索到了该题目，但没有返回正确的结果。在违背常识的问题“如何发展门头沟的海洋产业”这一问题上，尽管 ChatGPT 提到了“门头沟距离海洋较远”，但所有系统都没有很好的处理这一问题，从结果上来看，ChatGPT 返回的结果最长，Perplexity.AI 的结果有点不流畅。在自然语言指令问题上，ChatGPT 和 Perplexity.AI 都能遵从指令，对给定句子进行分词，YouChat 则无法遵从指令，从分词结果上来看，目前还没有系统能够正确分词。

[8]https://finance.sina.cn/tech/2023-02-10/detail-imyfezcs2988460.d.html

[9]https://finance.sina.com.cn/tech/internet/2023-02-27/doc-imyicmyz2049229.shtml

表 6.3: 不同系统针对数学表达式的运算结果

(175*45+15)/5=	
ChatGPT	The expression (175*45+15)/5 can be simplified using the order of operations, which is also known as PEMDAS (parentheses, exponents, multiplication and division, and addition and subtraction): First, we perform the multiplication in the parentheses: 175*45 = 7875 Then we add 15 to the result: 7875 + 15 = 7890 Finally, we divide by 5: 7890/5 = 1578 Therefore, (175*45+15)/5 = 1578.
YouChat	The answer to your equation is 584.
Preplexity.AI	(175*45+15)/5=787.5

表 6.4: 不同系统针对公务员考试数学应用题（原题）的运算结果

某单位 2011 年招聘了 65 名毕业生，拟分配到该单位的 7 个不同部门，假设行政部门分得的毕业生人数比其他部门都多，问行政部门分得的毕业生人数至少为多少名：	
ChatGPT	我们可以设行政部门分得的毕业生人数为 x， 那么其他部门分得的毕业生人数为 65-x。 根据题目的条件，行政部门分得的毕业生 人数比其他部门都多，因此有：x >65-x 解这个不等式得：x >32.5 由于 x 必须是整数， 因此行政部门分得的毕业生人数至少为 33 名。
YouChat	由题意可知，2011 年招聘了 65 名毕业生， 拟分配到 7 个不同部门，假设行政部门分 得的毕业生人数比其他部门都多，则行政 部门分得的毕业生人数至少为 9 名，其他 部门各分得的毕业生人数为：行政部门 9 名， 研发部门 8 名，市场部门 7 名，财务部门 7 名， 人事部门 6 名，运营部门 6 名，客服部门 6 名。
Preplexity.AI	为了使行政部门的毕业生人数最少， 其他部门应尽量多分配。 65 名毕业生平均分配到 7 个部门， 每个部门可以得到 9 名母业生。 但是因为有 2 名剩余， 所以行政部门至少可以得到 11 名毕业生

表 6.5: 不同系统针对违背常识问题的回答

如何发展门头沟的海洋产业?	
ChatGPT	门头沟区位于北京市西南部，距离海洋较远，因此发展 海洋产业需要通过合理的规划和利用现有资源来实现。 以下是一些可能有助于门头沟发展海洋产业的建议： 1、利用门头沟的地理位置和资源，开展海洋旅游业。门 头沟拥有得天独厚的自然风光，可以开发海洋主题公园、 海底世界等景点，吸引更多游客前来观光、休闲和娱乐。 2、发展海洋养殖业。通过在门头沟的河流、湖泊等水域内 养殖鱼类、虾类等水产品，可以满足北京市及周边地区的 海鲜市场需求，带动当地经济发展。 3、开展海洋科技研究和开发。门头沟可以成立海洋科技研究 中心，聚集专家学者、科研机构和企业等资源，开展海洋 环境保护、海洋新能源开发、海洋智能装备研发等方面的 科技创新，提高门头沟的科技创新能力和综合竞争力。 4、合理规划海洋产业发展区。门头沟可以依托现有的水域和 港口资源，打造现代化的海洋产业发展区，吸引海洋产业 企业入驻，提升门头沟的产业集聚和发展效益。 5、发挥政府引导作用。门头沟政府可以出台相关政策，支持 海洋产业企业的发展，引导社会资本投资海洋产业，激发 海洋产业发展的动力。 总之，门头沟发展海洋产业需要通过多方面的努力和合理规划 来实现，需要政府、企业、科研机构等各方合作，共同推进门 头沟海洋产业的发展。
YouChat	门头沟市可以借助海洋资源，发展海洋经济发展。首先， 门头沟市可以发展渔业产业，提高渔业的科技化水平， 开发海洋生物资源，改善渔民的生活条件。 其次，门头沟市可以通过海洋矿产开采，发展海洋矿业， 提高矿产开采效率，增加收入来源。此外，门头沟市还 可以发展海洋旅游业，保护海洋环境，开发海洋文化， 提高城市经济社会发展水平。
Preplexity.AI	门头沟区可以优先支持人工智能、医疗器械、超高清数 字视听等产业发展方向。此外，数字化推进海洋产业 发展也是一种可行的方法。其他省份也在推进海洋经济 的发展，例如广东省。北京市门头沟区有一份土地利用 总体规划。

表 6.6: 不同系统针对自然语言任务指令的回答

用毒毒毒蛇毒蛇会被毒毒死吗？给我分个词	
ChatGPT	用毒毒毒蛇毒蛇会被毒毒死吗? 分词如下：用 / 毒毒 / 毒蛇 / 毒蛇 / 会 / 被 / 毒毒 / 死 / 吗? 其中，“用”是动词，“会”是助动词，“被”是介词，“吗” 是助词，“毒毒”是形容词或副词，“毒蛇”是名词，“死” 是动词，“吗”是助词。
YouChat	毒蛇本身也有毒性，但大部分毒蛇都不会被自己的毒杀死。 毒蛇可以从本身的毒液中产生抗毒素，但在毒性超强的种 类中，毒蛇也有可能被自己的毒液杀死。
Preplexity.AI	用毒蛇的毒毒毒蛇，毒蛇会被毒死吗? 分词：用/毒蛇/的/毒/毒/毒蛇/，/毒蛇/会被/毒死/吗?

第七章　自然语言处理的未来发展方向

ChatGPT 的出现，给自然语言处理领域带来了巨大的影响。首先，大规模语言模型中已经蕴含了如分词、词性标注、句法分析等传统自然语言处理任务能提供的信息，因此在当下基于大模型的自然语言处理时代，已经不再需要对这些任务进行专门的研究了。其次，自然语言处理任务之间的界限也被打破了，ChatGPT 将所有任务都转化为序列到序列生成问题，不但能够使用同一个模型解决各种已有的任务，而且能够很好地处理未曾见过的新任务，因此之前针对某一任务进行研究的学者面临研究空间被挤压的危险。最后，由于工业界掌握了大量的计算资源、用户信息和用户反馈的数据，相较于学术界更容易进行系统级的创新，这种“AI 马太效应”会造成胜者通吃的局面，进一步加大自然语言处理学术界研究的困境。

当然，ChatGPT 也并非完美，如 1.4.2节和 6.2.2节所述，其仍然存在诸多不足，这也是短期内自然语言处理领域的研究者需要关注的问题。具体包括如何进一步弥补模型的不足、探究模型的机理和推广模型的应用三个方面。

7.1　弥补模型的不足

提高结果的可信性和时效性　ChatGPT 生成结果的可信性一直为人们所诟病，经常会出现“一本正经的胡说八道”的问题。另外，由于其预训练模型的数据截止至 2021 年，因此无法回答此后的相关信息。目前已有一些系统致力于通过引入搜索引擎的结果以及在模型生成结果中增加相关网页的链接等方式来解决可信性和时效性问题（如微软的 New Bing[1]等），但是结果中仍然存在一些事实性的错误等问题。

然而，可信性与创造性本身就是矛盾的，如果限制 ChatGPT 只能生成确定的事实，则会极大地限制其创造能力。因此，需要在可信性和创造性之

[1]https://www.bing.com/

间进行权衡，或者交由用户选择其希望得到哪种类型的结果。

提高符号推理能力 ChatGPT 在符号推理等能力上仍然存在不足，无法进行稍微复杂的算数运算以及逻辑推理等。这是由于其生成式语言模型的天然局限性造成的。目前已有一些研究者通过调用外部的符号计算引擎来解决这一问题，如 Meta 的 Toolformer[97] 能够让语言模型生成调用计算器、问答引擎等外部 API 的调用语句。还有的工作则是先生成 Python、SQL 等程序，再由相应外部的引擎来执行这些程序，然后结合执行的结果生成最终的答案。这也是让神经网络与符号知识相结合的一种有益尝试。

减小对大规模标注数据的依赖 ChatGPT 虽然具有非常惊艳的小样本甚至零样本处理能力，但是在进行指令微调以及人类反馈的训练阶段，其对大规模高质量人工标注数据的依赖仍然是其不可或缺的一部分。因此，如何减小对大规模高质量人工标注数据的依赖，依然是需要重点关注的问题，这将有助于模型在具体行业和领域中的应用落地。

提高多种语言处理能力 ChatGPT 表现出了非常优秀的多语言能力，人们能够使用英语、汉语等多种语言与其流畅的对话。虽然其究竟使用了多少多语言数据不得而知，但是其前身 InstructGPT 指令微调的指令集中 96% 以上是英语，其他 20 种语言（包含汉语、西班牙语、法语、德语等）只占不到 4%。因此，无论从处理语言的数量还是对少资源语言的处理质量上，ChatGPT 的多语言能力仍然需要进一步提升。

7.2 探究模型的机理

大模型的结构 目前，GPT 系列模型始终坚持使用解码器结构，和 Google 提出的 T5、Meta 提出的 BART 等编码-解码器结构的模型相比，这样做的好处有两点：1）可以高效地利用数据，即能对一个批次中的全部数据进行学习，而编码-解码器结构每批次只能对一半的数据进行学习，因此需要更多的数据才能达到相同的效果。2）在显存大小一定的条件下，解码器结构模型的层数是编码-解码器结构模型的两倍，因此能够更好地捕捉到数据中的潜在信息。但是，仅使用解码器的模型结构也有其不足，即在对用户的输入进行理解时，由于只进行了单向的编码，因此理解能力不如编码-解码器结构

充分。因此，未来的研究方向之一是如何在保证模型学习效率的同时，兼顾模型的理解能力和生成能力。

知识的调用方法 目前，ChatGPT 通过指令微调、COT、RLHF 等方式调用大模型中所蕴藏的知识。但是，这些方法都存在一些局限性，如指令微调需要人工编写复杂的指令，COT 也需要人工编写答案的推理过程，RLHF 需要人工标注反馈数据等。因此，未来的研究方向是如何能够让模型自动地调用大模型中的知识，减少人工的劳动。

对大模型的评价 和其他对话系统以及文本生成系统一样，目前还不存在完全客观的指标对 ChatGPT 等系统进行评价。因此主要是通过人工评价的方式，即人工对模型的输出进行评价。但是，这种评价方式存在一些局限性，如人效率低下、标准不一致等。虽然第五章给出了多种模型的评价指标，但是如何自动地对这些指标进行客观公正地评价，并且将多个指标的评价结果进行综合，仍然是一个值得研究的问题。

大模型的机理 虽然 ChatGPT 表现出了有趣的“涌现”现象，通过 COT 实现了一定的推理能力，具有简单的计算能力等，但是究竟是什么原因使得 ChatGPT 具有这些能力，仍然是一个未解之谜。因此，如何通过研究模型机理来解释 ChatGPT 等模型的表现是未来的研究方向之一，并有助于进一步提升和挖掘 ChatGPT 的能力。

7.3 实际应用

适配特定领域 虽然在通用任务上表现出了非常好的效果，但是在缺少相应数据的金融、医疗等专用领域，ChatGPT 表现并不理想，这极大地阻碍了 ChatGPT 的产业化应用。因此，需要研究如何利用专用领域大量无标注数据、少量有标注数据以及已有的知识库将 ChatGPT 适配到特定领域，从而实现 ChatGPT 在特定领域的产业化应用。

个性化模型 同样地，由于 ChatGPT 是一个通用模型，其对于不同用户的表现也是相同的，缺少对用户个性化信息的存储和利用。因此，如果能利用与用户的对话历史记录等个性化数据，来训练个性化的 ChatGPT 模型，将是一个非常有趣，也非常有意义的研究方向。

高效计算 大规模语言模型无论训练还是部署，都需要耗费大量的计算资源，这对于一般的企业和个人来讲，都是一个巨大的负担。因此，如果在有限的计算资源下，能够高效地训练和部署 ChatGPT，将是一个非常有意义的研究方向。相关的技术包括但不限于模型压缩、蒸馏、剪枝、量化等。

应对风险 最后，若要将 ChatGPT 以及后续更强大的通用人工智能系统落地应用，还必须解决 1.6节所述的众多风险问题，如模型的安全性、隐私性等等。其中有一些风险可以通过技术手段加以解决，但是更多的风险则需要通过法律等手段来解决，这是一个非常复杂的问题，需要更多的其他领域的研究者和专家的参与。

7.4 从语言到 AGI 的探索之路

经过近 70 年的发展，自然语言处理技术先后经历了五次范式的变迁，随着 ChatGPT 的产生，人们也看到了实现通用人工智能（AGI）的曙光。在这个过程中，自然语言处理技术呈现了明显的“同质化”和“规模化”的发展趋势。因此，我们认为未来自然语言处理还将沿着这一道路继续前进。即使用参数量越来越大的模型，从越来越多的文本数据中进行学习。

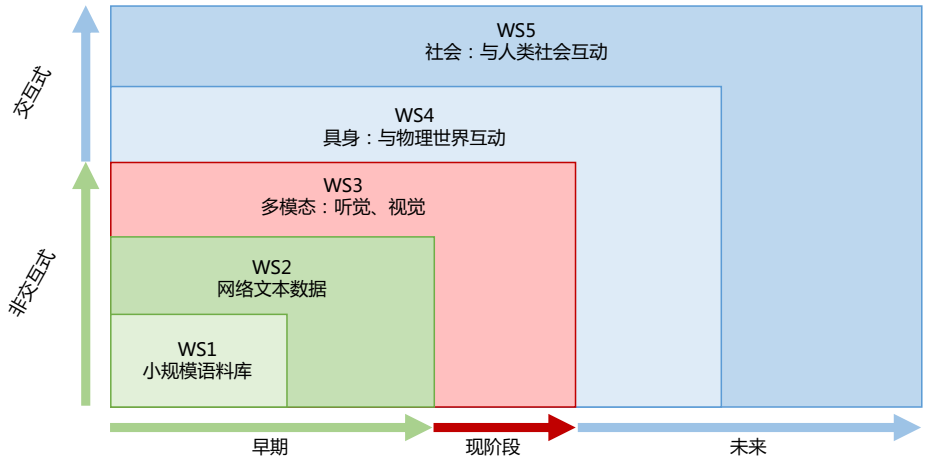

图 7.1: “世界范围”（World Scope）概念

然而，人类习得语言的途径绝不仅仅是文本这一条，还需要利用听觉、视觉、触觉等多种感官信息并将语言同这些信息进行映射。因此，自然语言处理未来需要融入更多的多模态信息。此外，还需要智能体能够同物理世界

以及人类社会进行交互，这样才能真正理解现实世界中的各种概念，从而实现真正的通用人工智能。

以上想法与 Bisk et al.[98] 提出的“世界范围”（World Scope，WS）概念不谋而合。如图 7.1所示，Bisk et al.[98] 将自然语言处理所需的信息来源划分为了五个范围。ChatGPT 所基于的大规模预训练语言模型处于 WS2，即网络文本数据范围，而 ChatGPT 通过对话的方式与人类用户交互，一下子迈入了 WS5 的范围。但是，为了实现真正的通用人工智能，还需要能够融合多模态信息（WS3），并实现与物理世界的交互，即具身能力（WS4）。

因此，我们完全有理由相信，在多模态版本的“ChatGPT”问世后，再结合具身智能，一个能够同时处理文字、语音、图像等各种模态指令，并且能和物理世界以及人类社会共存的通用人工智能体将在不久的将来真正诞生。

参考文献

[1] RADFORD A, NARASIMHAN K, SALIMANS T, et al. Improving language understanding by generative pre-training[J]., 2018 (引用页: 8, 26).

[2] DEVLIN J, CHANG M W, LEE K, et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding[C]//Proc. of NAACL. [S.l. : s.n.], 2019: 4171-4186 (引用页: 8, 24).

[3] RADFORD A, WU J, CHILD R, et al. Language models are unsupervised multitask learners[J]. OpenAI blog, 2019, 1: 9 (引用页: 8, 28).

[4] BROWN T B, MANN B, RYDER N, et al. Language Models are Few-Shot Learners[C]//Proc. of NeurIPS. [S.l. : s.n.], 2020 (引用页: 8, 28, 44, 45).

[5] QIU X, SUN T, XU Y, et al. Pre-trained models for natural language processing: A survey[J]. Science China Technological Sciences, 2020, 63: 1872-1897 (引用页: 8).

[6] KALYAN K S, RAJASEKHARAN A, SANGEETHA S. AMMUS : A Survey of Transformer-based Pretrained Models in Natural Language Processing[Z]. 2021. arXiv: 2108.05542 [cs.CL] (引用页: 8).

[7] AMATRIAIN X. Transformer models: an introduction and catalog[Z]. 2023 (引用页: 8, 40).

[8] LIU P, YUAN W, FU J, et al. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing[J]. ArXiv preprint, 2021, abs/2107.13586 (引用页: 8).

[9] KNOX W B, STONE P. Tamer: Training an agent manually via evaluative reinforcement[C]//2008 7th IEEE international conference on development and learning. [S.l. : s.n.], 2008: 292-297 (引用页: 10).

[10] MACGLASHAN J, HO M K, LOFTIN R T, et al. Interactive Learning from Policy-Dependent Human Feedback[C]//Proc. of ICML: vol. 70. [S.l. : s.n.], 2017: 2285-2294 (引用页: 10).

[11] WARNELL G, WAYTOWICH N R, LAWHERN V, et al. Deep TAMER: Interactive Agent Shaping in High-Dimensional State Spaces[C]//Proc. of AAAI. [S.l. : s.n.], 2018: 1545-1554 (引用页: 11).

[12] ZIEGLER D M, STIENNON N, WU J, et al. Fine-tuning language models from human preferences[J]. ArXiv preprint, 2019, abs/1909.08593 (引用页: 11).

[13] STIENNON N, OUYANG L, WU J, et al. Learning to summarize with human feedback[C]//Proc. of NeurIPS. [S.l. : s.n.], 2020 (引用页: 11).

[14] WU J, OUYANG L, ZIEGLER D M, et al. Recursively summarizing books with human feedback[J]. ArXiv preprint, 2021, abs/2109.10862 (引用页: 11).

[15] NAKANO R, HILTON J, BALAJI S A, et al. WebGPT: Browser-assisted question-answering with human feedback[J]. ArXiv preprint, 2021, abs/2112.09332 (引用页: 11, 21).

[16] OUYANG L, WU J, JIANG X, et al. Training language models to follow instructions with human feedback[J]. ArXiv preprint, 2022, abs/2203.02155 (引用页: 11, 33, 47).

[17] MENICK J, TREBACZ M, MIKULIK V, et al. Teaching language models to support answers with verified quotes[J]. ArXiv preprint, 2022, abs/2203.11147 (引用页: 11).

[18] GLAESE A, MCALEESE N, TRĘBACZ M, et al. Improving alignment of dialogue agents via targeted human judgements[J]. ArXiv preprint, 2022, abs/2209.14375 (引用页: 11).

[19] DOSOVITSKIY A, BEYER L, KOLESNIKOV A, et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale[C]// Proc. of ICLR. [S.l. : s.n.], 2021 (引用页: 11).

[20] BAI Y, KADAVATH S, KUNDU S, et al. Constitutional AI: Harmlessness from AI Feedback[J]. ArXiv preprint, 2022, abs/2212.08073 (引用页: 13).

[21] TAMKIN A, BRUNDAGE M, CLARK J, et al. Understanding the Capabilities, Limitations, and Societal Impact of Large Language Models[J]. ArXiv preprint, 2021, abs/2102.02503 (引用页: 20).

[22] MITCHELL E, LEE Y, KHAZATSKY A, et al. DetectGPT: Zero-Shot Machine-Generated Text Detection using Probability Curvature[J]. ArXiv preprint, 2023, abs/2301.11305 (引用页: 20).

[23] RAE J W, BORGEAUD S, CAI T, et al. Scaling language models: Methods, analysis & insights from training gopher[J]. ArXiv preprint, 2021, abs/2112.11446 (引用页: 21, 44).

[24] OGNYANOVA K, LAZER D, ROBERTSON R E, et al. Misinformation in action: Fake news exposure is linked to lower trust in media, higher trust in government when your side is in power[J]. Harvard Kennedy School Misinformation Review, 2020 (引用页: 21).

[25] KIM Y, SUNDAR S S. Anthropomorphism of computers: Is it mindful or mindless?[J]. Comput. Hum. Behav., 2012, 28: 241-250 (引用页: 22).

[26] RIMMER M. Patent-Busting: The Public Patent Foundation, Gene Patents, and the Seed Wars[C]//. [S.l. : s.n.], 2013 (引用页: 22).

[27] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is All you Need[C]//Proc. of NeurIPS. [S.l. : s.n.], 2017: 5998-6008 (引用页: 24).

[28] LAN Z, CHEN M, GOODMAN S, et al. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations[C]//Proc. of ICLR. [S.l. : s.n.], 2020 (引用页: 24, 44).

[29] LIU Y, OTT M, GOYAL N, et al. Roberta: A robustly optimized bert pretraining approach[J]. ArXiv preprint arXiv:1907.11692, 2019 (引用页: 24).

[30] MIKOLOV T, CHEN K, CORRADO G, et al. Efficient Estimation of Word Representations in Vector Space[C]//Proc. of ICLR. [S.l. : s.n.], 2013 (引用页: 27).

[31] LEWIS M, LIU Y, GOYAL N, et al. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension[C]//Proc. of ACL. [S.l. : s.n.], 2020: 7871-7880 (引用页: 29).

[32] RAFFEL C, SHAZEER N, ROBERTS A, et al. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer[J]. Journal of Machine Learning Research, 2020, 21(140): 1-67 (引用页: 29).

[33] WEI J, WANG X, SCHUURMANS D, et al. Chain of thought prompting elicits reasoning in large language models[J]. ArXiv preprint, 2022, abs/2201.11903 (引用页: 32, 33).

[34] KOJIMA T, GU S S, REID M, et al. Large language models are zero-shot reasoners[J]. ArXiv preprint, 2022, abs/2205.11916 (引用页: 33).

[35] NARAYANAN D, SHOEYBI M, CASPER J, et al. Efficient large-scale language model training on gpu clusters using megatron-lm[C]// Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis. [S.l. : s.n.], 2021: 1-15 (引用页: 34, 38).

[36] LI S, ZHAO Y, VARMA R, et al. Pytorch distributed: Experiences on accelerating data parallel training[J]. ArXiv preprint, 2020, abs/2006.15704 (引用页: 36).

[37] BIAN Z, LIU H, WANG B, et al. Colossal-AI: A Unified Deep Learning System For Large-Scale Parallel Training[J]. ArXiv preprint, 2021, abs/2110.14883 (引用页: 37).

[38] SHOEYBI M, PATWARY M, PURI R, et al. Megatron-lm: Training multi-billion parameter language models using model parallelism[J]. ArXiv preprint, 2019, abs/1909.08053 (引用页: 38).

[39] KORTHIKANTI V, CASPER J, LYM S, et al. Reducing activation recomputation in large transformer models[J]. ArXiv preprint arXiv:2205.05198, 2022 (引用页: 38).

[40] RAJBHANDARI S, RASLEY J, RUWASE O, et al. Zero: Memory optimizations toward training trillion parameter models[C]//SC20: International Conference for High Performance Computing, Networking, Storage and Analysis. [S.l. : s.n.], 2020: 1-16 (引用页: 38).

[41] REN J, RAJBHANDARI S, AMINABADI R Y, et al. ZeRO-Offload: Democratizing Billion-Scale Model Training.[C]//USENIX Annual Technical Conference. [S.l. : s.n.], 2021: 551-564 (引用页: 38).

[42] ZHANG S, ROLLER S, GOYAL N, et al. Opt: Open pre-trained transformer language models[J]. ArXiv preprint, 2022, abs/2205.01068 (引用页: 44, 45).

[43] HINTON G, VINYALS O, DEAN J. Distilling the knowledge in a neural network[J]. ArXiv preprint, 2015, abs/1503.02531 (引用页: 44).

[44] LI S, CHEN J, SHEN Y, et al. Explanations from large language models make small reasoners better[J]. ArXiv preprint, 2022, abs/2210.06726 (引用页: 44).

[45] HAN S, MAO H, DALLY W J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding[J]. ArXiv preprint, 2015, abs/1510.00149 (引用页: 44).

[46] JIAO X, YIN Y, SHANG L, et al. TinyBERT: Distilling BERT for Natural Language Understanding[C]//Proc. of EMNLP Findings. [S.l. : s.n.], 2020: 4163-4174 (引用页: 44).

[47] KI D, LEE S. Analyzing the effects of Green View Index of neighborhood streets on walking time using Google Street View and deep learning[J]. Landscape and Urban Planning, 2021, 205: 103920 (引用页: 44).

[48] HO N, SCHMID L, YUN S Y. Large Language Models Are Reasoning Teachers[J]. ArXiv preprint, 2022, abs/2212.10071 (引用页: 45).

[49] SCAO T L, FAN A, AKIKI C, et al. Bloom: A 176b-parameter open-access multilingual language model[J]. ArXiv preprint, 2022, abs/2211.05100 (引用页: 45).

[50] WEN W, WU C, WANG Y, et al. Learning Structured Sparsity in Deep Neural Networks[C]//Proc. of NeurIPS. [S.l. : s.n.], 2016: 2074-2082 (引用页: 45).

[51] FRANTAR E, ALISTARH D. Massive Language Models Can Be Accurately Pruned in One-Shot[J]. ArXiv preprint, 2023, abs/2301.00774 (引用页: 45).

[52] ZHU Y, KIROS R, ZEMEL R S, et al. Aligning Books and Movies: Towards Story-Like Visual Explanations by Watching Movies and Reading Books[C]//Proc. of ICCV. [S.l. : s.n.], 2015: 19-27 (引用页: 47).

[53] GAO L, BIDERMAN S, BLACK S, et al. The Pile: An 800GB Dataset of Diverse Text for Language Modeling[J]. ArXiv preprint, 2021, abs/2101.00027 (引用页: 48, 51).

[54] YUAN S, ZHAO H, DU Z, et al. WuDaoCorpora: A super large-scale Chinese corpora for pre-training language models[J]. AI Open, 2021, 2: 65-68 (引用页: 48).

[55] XU L, ZHANG X, DONG Q. CLUECorpus2020: A large-scale Chinese corpus for pre-training language model[J]. ArXiv preprint, 2020, abs/2003.01355 (引用页: 49).

[56] LIANG P, BOMMASANI R, LEE T, et al. Holistic evaluation of language models[J]. ArXiv preprint, 2022, abs/2211.09110 (引用页: 49, 64, 65, 67).

[57] HUSAIN H, WU H H, GAZIT T, et al. Codesearchnet challenge: Evaluating the state of semantic code search[J]. ArXiv preprint, 2019, abs/1909.09436 (引用页: 51).

[58] PURI R, KUNG D S, JANSSEN G, et al. CodeNet: A large-scale AI for code dataset for learning a diversity of coding tasks[J]. ArXiv preprint, 2021, abs/2105.12655 (引用页: 51).

[59] NIJKAMP E, PANG B, HAYASHI H, et al. A conversational paradigm for program synthesis[J]. ArXiv preprint, 2022, abs/2203.13474 (引用页: 51).

[60] LE H, WANG Y, GOTMARE A D, et al. CodeRL: Mastering Code Generation through Pretrained Models and Deep Reinforcement Learning[C]//Proc. of NeurIPS. [S.l. : s.n.], 2022 (引用页: 51).

[61] IYER S, KONSTAS I, CHEUNG A, et al. Mapping Language to Code in Programmatic Context[C]//Proc. of EMNLP. [S.l. : s.n.], 2018: 1643-1652 (引用页: 51).

[62] WEI J, BOSMA M, ZHAO V, et al. Finetuned Language Models are Zero-Shot Learners[C]//Proc. of ICLR. [S.l. : s.n.], 2022 (引用页: 53).

[63] WEI J, BOSMA M, ZHAO V, et al. Finetuned Language Models are Zero-Shot Learners[C]//Proc. of ICLR. [S.l. : s.n.], 2022 (引用页: 53).

[64] SANH V, WEBSON A, RAFFEL C, et al. Multitask Prompted Training Enables Zero-Shot Task Generalization[C]//Proc. of ICLR. [S.l. : s.n.], 2022 (引用页: 53).

[65] CHUNG H W, HOU L, LONGPRE S, et al. Scaling Instruction-Finetuned Language Models[J]. ArXiv preprint, 2022, abs/2210.11416 (引用页: 53, 56).

[66] BICKEL A, KANAI I. Über den Angriffspunkt des Histamins und der in einigen Nahrungsmitteln vorkommenden sekretinartig wirkenden Substanzen am Sekretionsapparat der Magenfundusdrüsen[J]. Digestion, 1933 (引用页: 58).

[67] CALLISON-BURCH C, FORDYCE C S, KOEHN P, et al. (Meta-) Evaluation of Machine Translation[J]. Workshop on Statistical Machine Translation, 2007 (引用页: 59).

[68] PAPINENI K, ROUKOS S, WARD T, et al. BLEU: A Method for Automatic Evaluation of Machine Translation[C]//Proc. of ACL. [S.l. : s.n.], 2002: 311-318 (引用页: 60).

[69] BANERJEE S, LAVIE A. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments[J]. Proc. of ACL, 2005 (引用页: 60).

[70] LIN C Y. Rouge: A package for automatic evaluation of summaries[C]// Text summarization branches out. [S.l. : s.n.], 2004: 74-81 (引用页: 60).

[71] RUS V, LINTEAN M. A Comparison of Greedy and Optimal Assessment of Natural Language Student Input Using Word-to-Word Similarity Metrics[C]//Proceedings of the Seventh Workshop on Building Educational Applications Using NLP. [S.l. : s.n.], 2012: 157-162 (引用页: 60).

[72] WIETING J, BANSAL M, GIMPEL K, et al. Towards universal paraphrastic sentence embeddings[J]. ArXiv preprint arXiv:1511.08198, 2015 (引用页: 60).

[73] FORGUES G, PINEAU J, LARCHEVÊQUE J M, et al. Bootstrapping dialog systems with word embeddings[C]//Nips, modern machine learning and natural language processing workshop: vol. 2. [S.l. : s.n.], 2014: 168 (引用页: 60).

[74] BENGIO Y, DUCHARME R, VINCENT P. A neural probabilistic language model[J]. Proc. of NIPS, 2000, 13 (引用页: 60).

[75] LI J, GALLEY M, BROCKETT C, et al. A diversity-promoting objective function for neural conversation models[J]. ArXiv preprint arXiv:1510.03055, 2015 (引用页: 60).

[76] LOWE R, NOSEWORTHY M, SERBAN I V, et al. Towards an Automatic Turing Test: Learning to Evaluate Dialogue Responses[C]// Proc. of ACL. [S.l. : s.n.], 2017: 1116-1126 (引用页: 60).

[77] TAO C, MOU L, ZHAO D, et al. RUBER: An Unsupervised Method for Automatic Evaluation of Open-Domain Dialog Systems[C]//Proc. of AAAI. [S.l. : s.n.], 2018: 722-729 (引用页: 60).

[78] GHAZARIAN S, WEISCHEDEL R M, GALSTYAN A, et al. Predictive Engagement: An Efficient Metric for Automatic Evaluation of Open-Domain Dialogue Systems[C]//Proc. of AAAI. [S.l. : s.n.], 2020: 7789-7796 (引用页: 60).

[79] ZHANG T, KISHORE V, WU F, et al. BERTScore: Evaluating Text Generation with BERT[C]//Proc. of ICLR. [S.l. : s.n.], 2020 (引用页: 61).

[80] GUO C, PLEISS G, SUN Y, et al. On Calibration of Modern Neural Networks[Z]. 2017 (引用页: 62).

[81] GEHMAN S, GURURANGAN S, SAP M, et al. RealToxicityPrompts: Evaluating Neural Toxic Degeneration in Language Models[C]//Proc. of EMNLP Findings. [S.l. : s.n.], 2020: 3356-3369 (引用页: 62).

[82] WANG X, LIU Q, GUI T, et al. TextFlint: Unified Multilingual Robustness Evaluation Toolkit for Natural Language Processing[C]// Proc. of ACL. [S.l. : s.n.], 2021: 347-355 (引用页: 63).

[83] NIU T, BANSAL M. Adversarial Over-Sensitivity and Over-Stability Strategies for Dialogue Models[C]//Proc. of CoNLL. [S.l. : s.n.], 2018: 486-496 (引用页: 63).

[84] SUN H, XU G, DENG J, et al. On the Safety of Conversational Models: Taxonomy, Dataset, and Benchmark[C]//Proc. of ACL Findings. [S.l. : s.n.], 2022: 3906-3923 (引用页: 63).

[85] DINAN E, HUMEAU S, CHINTAGUNTA B, et al. Build it Break it Fix it for Dialogue Safety: Robustness from Adversarial Human Attack[C]//Proc. of EMNLP. [S.l. : s.n.], 2019: 4537-4546 (引用页: 63).

[86] XU J, JU D, LI M, et al. Bot-Adversarial Dialogue for Safe Conversational Agents[C]//Proc. of NAACL. [S.l. : s.n.], 2021: 2950-2968 (引用页: 63).

[87] LEES A, TRAN V Q, TAY Y, et al. A New Generation of Perspective API: Efficient Multilingual Character-Level Transformers[C]//Proc. of KDD. [S.l. : s.n.], 2022: 3197-3207 (引用页: 64).

[88] DINAN E, ABERCROMBIE G, BERGMAN A, et al. SafetyKit: First Aid for Measuring Safety in Open-domain Conversational Systems[C]// Proc. of ACL. [S.l. : s.n.], 2022: 4113-4133 (引用页: 64).

[89] MAY C, WANG A, BORDIA S, et al. On Measuring Social Biases in Sentence Encoders[C]//Proc. of NAACL. [S.l. : s.n.], 2019: 622-628 (引用页: 64).

[90] NADEEM M, BETHKE A, REDDY S. StereoSet: Measuring stereotypical bias in pretrained language models[C]//Proc. of ACL. [S.l. : s.n.], 2021: 5356-5371 (引用页: 64).

[91] GROENWOLD S, OU L, PAREKH A, et al. Investigating African-American Vernacular English in Transformer-Based Text Generation[C]// Proc. of EMNLP. [S.l. : s.n.], 2020: 5877-5883 (引用页: 65).

[92] ABID A, FAROOQI M, ZOU J. Persistent Anti-Muslim Bias in Large Language Models[C]//Proc. of AAAI. [S.l. : s.n.], 2021: 298-306 (引用页: 65).

[93] BANG Y, LEE N, ISHII E, et al. Assessing Political Prudence of Open-domain Chatbots[C]//Proceedings of the 22nd Annual Meeting of the Special Interest Group on Discourse and Dialogue. [S.l. : s.n.], 2021: 548-555 (引用页: 65).

[94] SANTURKAR S, TSIPRAS D, MADRY A. BREEDS: Benchmarks for Subpopulation Shift[C]//Proc. of ICLR. [S.l. : s.n.], 2021 (引用页: 65).

[95] DHOLE K D, GANGAL V, GEHRMANN S, et al. Nl-augmenter: A framework for task-sensitive natural language augmentation[J]. ArXiv preprint, 2021, abs/2112.02721 (引用页: 66).

[96] GARDNER M, ARTZI Y, BASMOV V, et al. Evaluating Models' Local Decision Boundaries via Contrast Sets[C]//Proc. of EMNLP Findings. [S.l. : s.n.], 2020: 1307-1323 (引用页: 66).

[97] SCHICK T, DWIVEDI-YU J, DESSÌ R, et al. Toolformer: Language Models Can Teach Themselves to Use Tools[Z]. 2023 (引用页: 80).

[98] BISK Y, HOLTZMAN A, THOMASON J, et al. Experience Grounds Language[C]//Proc. of EMNLP. [S.l. : s.n.], 2020: 8718-8735 (引用页: 83).