保护大型语言模型：威胁、漏洞和负责任的实践

Sara Abdali*
saraabdali@microsoft.com
微软
雷德蒙德，华盛顿州，美国

Richard Anarfi†
ranarfi@microsoft.com
微软
马萨诸塞州波士顿，美国

CJ Barberan†
cjbarberan@microsoft.com
微软
马萨诸塞州波士顿，美国

Jia He †
hejia@microsoft.com
微软
马萨诸塞州波士顿，美国

摘要

大型语言模型（LLMs）已经显著改变了自然语言处理（NLP）的格局。 它们的影响跨越了各种任务的广泛领域，彻底改变了我们对语言理解和生成的方式。 然而，除了它们引人注目的实用性外，LLMs还引入了重要的安全和风险考虑。 这些挑战需要仔细审查，以确保负责任的部署并防范潜在的漏洞。 本研究全面调查了与LLMs相关的安全和隐私问题，从五个主题角度分析：安全和隐私问题，对抗性攻击的漏洞，LLMs误用可能造成的潜在危害，解决这些挑战的缓解策略，同时识别当前策略的局限性。 最后，本文建议未来研究的有希望的方向，以增强LLMs的安全和风险管理。[1]

1 引言

最近，大型语言模型（LLMs）在自然语言处理（NLP）领域，包括自然语言生成（NLG），引发了重大的范式转变。 LLMs通常以大量参数为特征，通常范围从百万到万亿不等，并且是使用深度神经网络构建的，主要是变压器架构（Vaswani等，2017年；Lin等，2021年）.

它们经历大量文本数据的预训练，通常是从网络收集而来的，以及利用自监督（焦等，2023年）、半监督（史等，2023a年）或强化学习（RL）（欧阳等，2022年；古尔切雷等，2023年）方法进行预训练和微调。 研究人员继续探索方法，对这些LLMs进行微调和适应特定应用，使它们成为自然语言处理社区及其他领域中不可或缺的工具。

LLMs展示了生成连贯、类似人类文本的显著能力，通常是针对给定的文本输入，也被称为提示（赵等，2023b年）。

例如，LLMs可以帮助用户有效沟通，提供一致且具有上下文意识的回应。 它们使用户能够快速获取信息，总结大量文本或回答复杂查询。 此外，LLMs有助于生成各种创意内容，无论是生成诗歌、故事还是代码片段。 除了个人用例，它们还在教育、研究和创新中发挥着关键作用，涵盖科学、艺术和文学等各个领域。

更具体地说，LLMs在各种自然语言处理任务上取得了令人印象深刻的成果，如文本生成（Senadeera和Ive，2022）、问答（Zaib等，2021；Bhat等，2023）、情感分析（Batra等，2021；Kheiri和Karimi，2023），以及通过改善人机交互（HCI）来增强人类能力（Oppenlaender和Hamalainen，2023；Hämäläinen等，2023）。虽然LLMs在各种任务上表现出显著的性能提升，但它们也在安全、隐私和道德规范方面面临重要挑战。

例如，LLM通常是在网络上大量文本的预训练，这些文本可能包含敏感、个人或机密信息。 这会导致泄露或被对手滥用的风险。 (Weidinger等，2021).

* 通讯作者
† 这些作者贡献相同。
[1]本研究代表作者进行的独立研究，不一定代表任何组织的观点或意见。

它们也可能被用于生成有偏见、有毒、有害和歧视性内容（Kuchnik等，2023年），侵犯知识产权（Peng等，2023年；Stokel-Walker，2022年），绕过企业安全协议（Shayegani等，2023年；Mozes等，2023年），或其他恶意目的，如生成网络安全攻击（Han等，2023年）和传播错误信息和宣传（Vykopal等，2023年；Mozes等，2023年）。

为了促进大型语言模型的负责任和道德使用，必须开发能够根据公平、问责、透明和可解释性原则评估、改进和管理大型语言模型的方法和框架。 这项任务需要从安全、伦理和风险缓解的角度全面跨学科地调查大型语言模型。

虽然已经有几项现有研究（Ganguli等，2022年；Huang等，2023年；Sun等，2023年；Deshpande等，2023年；Wang等，2023年）对大型语言模型的安全性和风险进行了研究，但这一领域的快速发展和创新需要更严格和系统的分析。

为了提高对大型语言模型相关潜在威胁和漏洞的意识，并促进负责任的实践，我们探讨了与大型语言模型相关的模型基础、训练时和推断时漏洞，并将它们分类。

此外，我们探讨解决方案和最佳实践，以确保它们的安全和负责任的使用。我们的方法涉及对与LLMs相关的安全和风险缓解方面进行严格的调查和评估。通过这样做，我们旨在突出现有研究中的差距和局限性，并提出未来的方向。 总之，我们研究的关键方面是：

- **安全风险：**我们确定了由LLM使用引起的安全问题，包括信息泄露、记忆和LLM生成的代码中的安全漏洞。
- **易受攻击的漏洞和风险：**我们讨论LLMs对攻击的易受性，包括基于模型、基于训练和基于推断的攻击。
- **滥用风险：**我们分析了与LLMs相关的风险和滥用，包括偏见、歧视和错误信息。
- **风险缓解策略：**我们的全面评估涵盖了缓解策略，如红队演练、模型编辑、水印技术和AI生成文本检测技术，同时讨论了局限性和权衡。
- **未来研究方向：**我们探索旨在解决LLMs相关安全和风险问题的新研究方向。

本文的其余部分组织如下：作为初步步骤，我们在第2节提供了一个主要术语表，以促进本文的可读性并消除不必要的重复。

然后，在第4节，我们通过将其分类为三个主要类别：基于模型、基于训练和基于推断的攻击及其相应的对策，介绍了LLMs的一些主要漏洞。 此外，在第3节，我们调查了LLMs使用中出现的安全问题。 我们在第2节详细阐述了LLMs的更多风险和误用情况。 随后，在第6节，我们讨论减少此类风险的缓解策略，然后在第7节提出新的研究方向。 最后，在第8节我们得出结论。 本文内容概述如图1所示。

2 背景

为了提高本文的清晰度并避免冗余，我们提供了一个词汇表，其中列出了本工作中经常使用的关键术语。
我们呈现表1，显示了术语、它们的简明定义以及详细阐述它们的论文部分。 我们鼓励读者在遇到陌生或不清楚的术语时参考词汇表。 词汇表旨在作为一个快速参考指南，而不是概念的全面解释。

术语	描述	部分
遗忘	LLMs重新训练或微调的数据预处理步骤。它明确地从数据集中删除被识别为泄漏易受攻击的数据点，并在处理后的数据集上重新训练或微调LLMs。(Cao and Yang, 2015)	3.1
记忆	记忆化是指LLMs保留并复制其训练数据中的信息的现象。 记忆化对于需要事实或语言知识的任务可能有益，但对于隐私、安全和质量原因也可能有问题（Hartmann等，2023a; Zhong等，2023）。	3.2
关联	关联是LLMs形成不同信息之间的连接的能力，如单词、实体、概念或事件。 LLMs中的关联能够实现各种应用，如知识检索、文本摘要和问答。 然而，LLMs中的关联也可能带来挑战，如隐私泄露、幻觉和偏见（Shao等，2023; Du等，2023a; Chen和Ding，2023）。	3.2
审计	审计是进行检查以了解LLMs记忆化的含义和后果。 例如，在审计逐字记忆中，检查将包括一个设置，用于生成任意生成的字符串，以便检测LLM是否能够以高概率提供所述任意生成的字符串的输出。(Hartmann等人，2023b)	3.2
对抗攻击	对抗攻击是一种利用LLM的漏洞或不足之处诱导出错误或欺骗性输出的方法。 对抗攻击可以被用于恶意目的，比如制造错误信息、规避安全协议或破坏模型的可靠性。(Shayegani等人，2023)	4
攻击成功率 (ASR)	攻击成功率是对对抗攻击对机器学习模型的有效性的衡量。 它被计算为成功攻击的比例占总攻击次数。 成功的攻击是使模型产生错误预测或输出的攻击。 攻击成功率可能会根据攻击类型、模型类型、任务类型和扰动程度而变化（吴等，2021年）。	4
模型提取攻击	一种对抗性攻击形式，利用大量查询及其对应的响应来提取大型语言模型的知识或参数。然后可以使用提取的信息来训练一个近似目标大型语言模型的减少参数模型，或者对大型语言模型或其他模型进行后续攻击。 提示提取（柯克等，2023年）、模型寄生（伯奇等，2023a年）和侧信道攻击（托尔和苏纳，2023年）是模型提取攻击的常见示例。	4.1.1
数据毒化	是一种破坏大型语言模型训练数据的攻击，影响其性能、行为或输出。 数据毒化可能导致模型中的偏见、虚假、有毒、后门或漏洞等问题。 数据污染可能是恶意行为者故意为了损害或劫持模型，也可能是由疏忽或不明智的数据提供者意外造成的，他们忽视了数据质量和安全标准。 通过使用可靠的数据来源、检查和清理数据、检测模型中的异常情况，并评估模型的弹性，可以避免或减少数据污染（Chen等，2017年；Schwarzschild等，2020年；Yang等，2021年）。	4.2.1
后门攻击	一种恶意操纵的类型，将隐藏的触发器嵌入模型中，导致其在良性样本上正常运行，但在受污染的样本上表现出性能下降。 这个问题在通信网络中尤为令人担忧，可靠性和安全性至关重要（Yang等，2023年）。 输入触发、指令触发和演示触发是一些在LLMs上发动后门攻击的常见方式（Zhao等，2023年；Yao等，2023年；Huang等，2023年；Zhu等，2022年）。	4.2.2

续页

术语	描述	部分
改写攻击	一种利用改写模型重写人工智能生成文本并规避检测的攻击。 它可以增强人工智能生成文本的自然性和人类感，绕过检测器的签名或模式。 改写攻击可能挑战大型语言模型及其应用的安全性和可靠性（Krishna等，2023年；Sadasivan等，2023年）。	4.3.1
欺骗攻击	在大型语言模型的背景下，欺骗攻击是一种模仿特定大型语言模型并使用修改后的大型语言模型创建类似输出的对抗性攻击。 它可以产生有害、欺骗性或与其预期功能或声誉不符的输出。 例如，伪装的聊天机器人可以模仿流行的大型语言模型并生成辱骂和虚假话语，或泄露机密信息，从而危及基于大型语言模型的应用的安全性和隐私（Shayegani等，2023年）。	4.3.1
提示注入	是一种对LLM进行的敌对攻击，通过提供指令来改变其输出，这些指令会覆盖或与预期的指令冲突。(Liu等，2023a; Greshake等，2023)	4.3.2
提示泄露	提示泄露是一种提示注入的类型，这是一种恶意策略，利用语言模型的漏洞来通过欺骗性提示改变其输出。 提示泄露可能会暴露原始提示中嵌入的敏感或专有信息，如数据信息。 提示泄露可能危及依赖语言模型的应用程序的安全和隐私（Perez和Ribeiro，2022年）。	4.3.2
越狱攻击	越狱攻击是一种利用LLMs的漏洞来改变其输出以欺骗性提示的攻击形式。 越狱攻击可以导致LLM产生不适当、有害或与其预期功能不一致的输出。 例如，越狱攻击可能导致LLM聊天机器人泄露机密信息，生成滥用或虚假话语，或者承认其人工性。 越狱攻击可能危及基于LLM的应用程序的安全和隐私（Zhang等，2023b; Deng等，2023b）。	4.3.3
黑盒检测	黑盒检测是识别大型语言模型输出中的不准确性或在不访问其内部状态或训练数据的情况下检测由大型语言模型生成的文本的任务。 通常涉及提出跟进问题，分析模型的响应，并应用分类器来检测欺骗模式（匿名，2023年）。 这是一个具有挑战性和重要性的问题，因为大型语言模型可以生成看似真实但错误的陈述，可能会误导或伤害用户。	6.3
白盒检测	通过完全访问目标模型来检测由大型语言模型生成的文本的任务。 这种方法可以防止未经授权使用大型语言模型并监视它们的生成行为（王等，2023年）。	6.3
水印	大型语言模型中的水印技术是一种将隐藏信号嵌入到大型语言模型生成的文本中，使其在算法上可识别为合成的，同时对人类来说是不可感知的技术。 水印技术可以帮助减轻大型语言模型的潜在风险，如虚假信息、抄袭或冒充，通过证明文本的所有权、真实性和完整性（Kirchenbauer等，2023a；Tang等，2023a）	6.3.4

表1：常用术语表

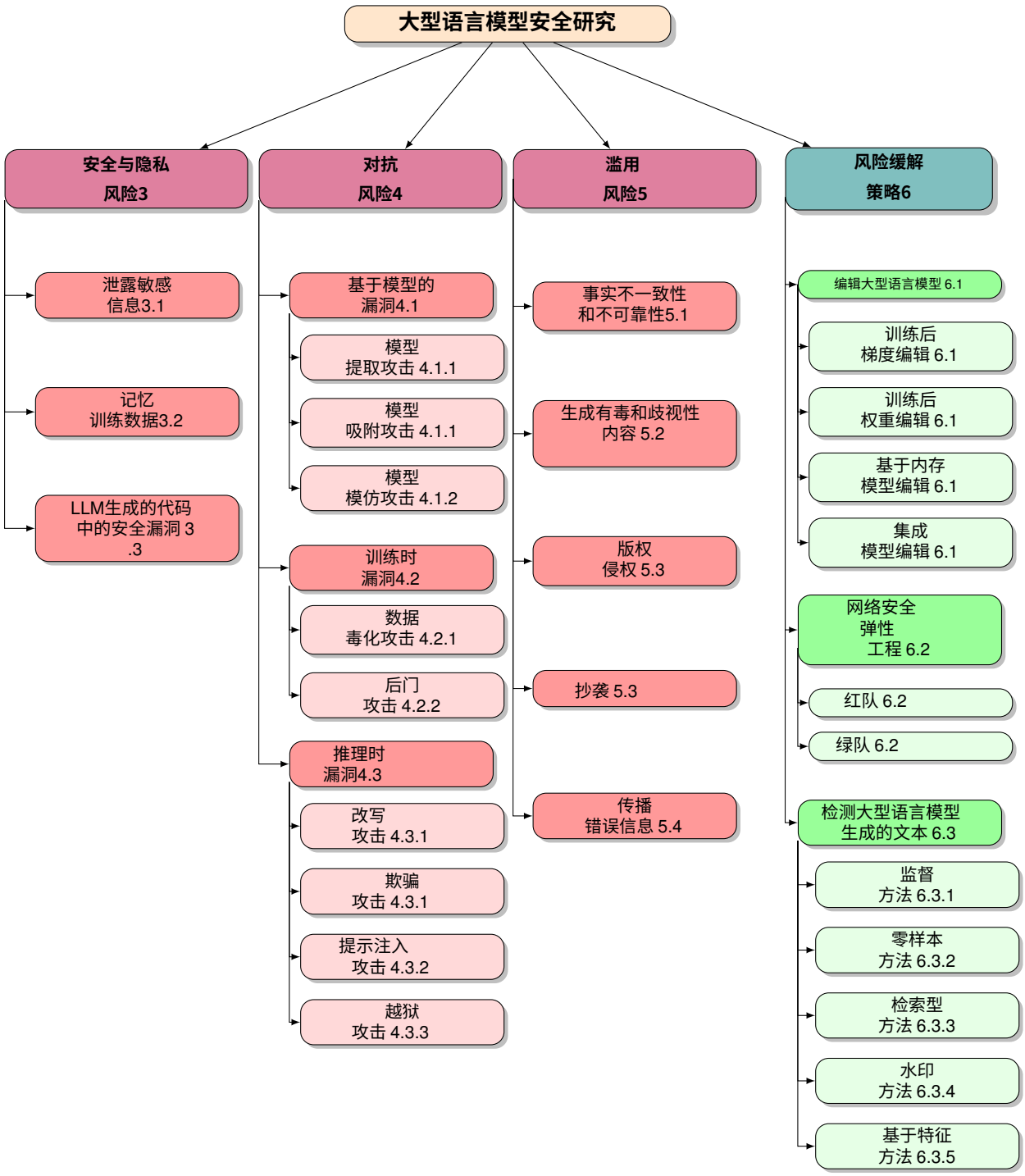

图1：LLM安全研究概览，包括安全、隐私、对抗性和误用风险以及现有的缓解策略。

LLM的3个安全和隐私问题

LLM是强大的工具，可能给企业和个人带来多种安全风险。 在本节中，我们探讨关键问题，包括敏感信息泄露、记忆训练数据以及生成代码中潜在的安全漏洞。 通过了解这些风险，我们可以促进负责任的人工智能使用，并增强整体安全实践。

3.1 信息泄露 LLMs被训练在大量的

网络收集数据上，这些数据不可避免地包含敏感或个人信息。 这种情况引发了关于个人可识别信息（PII）泄露的重大担忧。 常见的PII包括姓名、电子邮件地址和电话号码。 几乎任何在网上可以访问到PII的人都有可能受到隐私问题的影响。 因此，评估当前LLMs的隐私状态至关重要，包括预训练和微调模型。 这种评估有助于更好地了解隐私风险，并指导制定策略来减轻这些风险。

鉴于数据泄露，先前的研究已经审查了LLMs中隐私泄露的潜在风险。例如，Jaydeep等人（Borkar, 2023）调查了微调模型的隐私泄露。 他们使用起始序列标记或来自精调或预训练的随机十个标记来促使精细调整模型

数据，并通过查找训练数据和模型生成数据之间的常见n-gram来评估记忆。 研究表明，预训练和微调数据泄漏也会发生在精细调整的模型中。

此外，他们发现，通过取消学习（Cao和Yang，2015）来减轻精细调整模型中PII泄漏的现有解决方案可能会对先前安全的数据造成潜在危害。

隐私风险应该从PII所有者和模型提供者的角度进行评估，他们对LLMs具有黑盒访问权限，但他们的PII在训练数据中，以及模型提供者，他们根据Kim等人的提议具有白盒访问权限。 （Kim等人，2023年）。 他们提出了两种探测方法，通过在黑盒设置中设计策略性提示来赋予PII所有者和LLM服务提供者权力，并在白盒设置中微调有效提示。他们探测记忆

在黑盒设置中，通过提供 $n-1$ PII 并测试模型的响应是否包含剩下的一个 PII。白盒设置旨在自动调整一个手工设计的提示，可以导致最坏情况的泄漏。 为此，他们利用了公开可用的 Open Pre-trained Transformers (OPT) 模型，并优化以预测可以最大程度重建目标 PII 的标记。

这些方法为了解与 LLMs 相关的隐私风险提供了宝贵的见解，并为在 LLMs 环境中保护敏感信息提供了指导。通过评估不同 LLMs 中的隐私风险，我们可以更好地了解和解决隐私风险，并制定减轻风险的策略。

数据泄露发生在通过模型的完成暴露训练数据或真实用户输入中的敏感、个人或私人信息时。 例如，如果语言模型生成一个属于真实身份的信用卡号码或电子邮件地址，就被视为数据泄露。

另一方面，记忆是指语言模型回忆训练数据中的特定示例并在推理过程中重现它们的能力。 例如，如果模型生成了它训练过的文章或新闻标题的确切文本，就发生了数据记忆。 需要注意的是，数据泄露可能是由记忆引起的，但并非总是如此。

有时，大型语言模型可以生成训练数据中并非明确存在但仍然敏感或私密的数据。 这些信息可能是从文本的其他部分推断或推理出来的。 在响应生成过程中提供适当的指导对避免此类意外披露起着至关重要的作用。

在下一节中，我们将深入探讨记忆的概念。

3.2 记忆训练数据 大型语言模型

的出现，以及其庞大的参数数量，引发了人们对其记忆程度的担忧。 具体来说，当这些模型从训练数据中学习时，存在一个问题，即是否可以通过请求访问LLM内部机制的提示来访问该数据中包含的信息。

此外，由于LLM有各种不同大小，

关于它们是否有一种明显的模式，即它们如何轻松地记住信息？ 更大的模型是否更容易记住特定类型的数据？ 在本节中，我们旨在通过检查多个研究研究来解决这些问题，这些研究研究调查了LLM记忆和提取训练数据的复杂性。LLM具有大量参数，这引发了一个问题，即它们从训练数据中记住了

多少。 尽管大多数作品（Tirumala等，2022年；Carlini等，2021年；Biderman等，2023b年）使用术语“记忆”来描述学习训练数据的过程，但这种现象通常仅限于一定长度。 不同模型记忆训练数据的能力不同，导致了这种细致的行为。 大型语言模型通过其复杂的参数结构，在复杂性和适应性之间取得平衡，使其能够识别复杂模式，同时避免过度拟合和计算需求。

例如，根据(Tirumala等人，2022)的说法，记忆被表示为：

$$Mem(f) = \frac{\Sigma_{(s,y)\in C} argmax(f(s)) = y}{|C|}$$

其中， C是包含元组 (s, y)的上下文集合，其中 s作为文本输入块， y作为地面真实标记的索引。 因此，如果 argmax(f(s)) =y，那么上下文c就被记忆了。 这种记忆方式也在(Kuchnik等人，2023)中用于URL提取。

为了解决记忆问题，各种研究已经进行，以更全面地评估或从不同角度来评估它。

例如，De等人（de Wynter等人，2023年）检查了九个LLM，以查看生成数据中有多少是被记忆的。

此外，一些研究已经进行，以确定不同大小的LLM是否会记忆相同的训练材料（Biderman等人，2023a年）。 进行这项研究的Biderman等人提出，如果较小的模型记忆了一部分训练数据，并不意味着较大的LLM会记忆相同的训练数据。

在最近的一项研究中，Nasr等人（Nasr等人，2023年）进一步探讨了LLM中有多少训练数据被记忆。他们通过生成十亿个输出令牌来实现这一点。

LLM的研究发现，记忆化的比例在0.1%到1%之间。 为了进一步调查，他们进行了额外的实验，评估可以提取的独特50个标记字符串的数量。

这些独特的50克拉姆在不同模型之间表现出相当大的变化，从数十万到数百万不等。 具体来说，像LLaMA和Mistral这样的模型展示了更广泛的记忆化（数百万），而OPT则在数十万范围内。 这一观察结果促使进一步检查LLM中的记忆化。总的来说，随着参数数量的增加，理解它们的记忆化变得至关重要。 前面提到的研究已经建立了一个框架，用

于分类记忆化以便进行量化。 观察这项研究带来的其他途径以及新型LLM可能展示的记忆化类型将是很有趣的。 表2显示了不同的记忆研究，以及用于记忆的数据集和技术。

在最近的讨论中，研究人员探讨了LLMs中记忆和泛化能力之间的关系。具体来说，他们研究了LLM中高度记忆是否会妨碍其有效泛化的能力。

例如，Hartmann等人（Hartmann等人，2023b）引入了“审计”概念，以区分LLM是仅从事逐字记忆还是利用该信息进行更深入洞察。作者进一步提出了一个全面的分类法，将LLMs中不同形式的记忆进行分类，包括逐字文本、事实、理想、算法、写作风格、分布特性和对齐目标。

应该记住，虽然记忆对于某些任务（如问答）可能有利，但它也引发了与隐私、安全和版权相关的担忧。

黄等人（Huang et al., 2022）对预训练语言模型对隐私泄露的易感性进行了调查，特别关注个人信息的一个特定类别—电子邮件地址。 该研究确定了导致隐私泄露的两种不同机制：“记忆”和“关联”.

记忆化指的是模型的容量

来记忆敏感数据，并随后在用户查询时检索出来。 另一方面，关联指的是模型将攻击者精心设计的提示与训练过程中遇到的个人信息联系起来的能力。

为了量化记忆化，黄等人为大型语言模型提供了一个前缀序列，该序列在目标电子邮件地址之前，以引出目标。
为了量化关联，他们设计了不同的提示，基于电子邮件地址在句子中的出现方式。

他们的研究发现，虽然大型语言模型确实因为记忆而泄露了私人数据，但在关联任务中的表现相对较弱。 有趣的是，风险随着模型大小增加而增加，符合预期——更大的模型展现出更强的复杂性和记忆能力。

这一领域现有的研究仍处于萌芽阶段，需要额外的努力来确定记忆化是否是下游任务的可靠指标。 例如，一个大型语言模型可能会记忆事实信息来构建论点，但关键在于它能够连贯地连接这些记忆的事实。 尽管进行了这些调查，但在这一时刻，泛化和记忆之间的明确联系或相关性仍然难以捉摸。

3.3 大型语言模型生成的代码中的安全和隐私漏洞

大型语言模型有潜力帮助各种编码任务，如代码摘要（Alon等，2018）、代码补全（Bruch等，2009）、错误识别和定位（Wang等，2016）以及程序合成（Shin等，2019）。 尽管大型语言模型具有许多有用的应用，但担心它们可能被滥用以生成恶意工具是一个严重问题。 最近，研究人员对大型语言模型在代码生成中可能存在的潜在危害和后果进行了批判性调查。 Charan等人（Charan等人，2023）的一项研究表明，ChatGPT和Google的Bard可以用于生成顶级MITRE TTPs的代码[2]。 根据这项研究，ChatGPT使攻击者，尤其是业余者，更容易执行更专业化和复杂的任务。

[2]麻省理工学院枪火战术（MITRE）公司是一家与美国政府密切合作的非营利组织，已创建了战术、技术和程序（TTPs）以提供评估网络安全解决方案有效性的框架。

通过快速构建各种复杂的擦除和勒索软件攻击。

另一项研究调查了大型语言模型在制作网络钓鱼攻击中的应用（Roy等人，2023）。 本研究设计了几个针对ChatGPT的恶意提示，以构建功能性钓鱼网站。 研究表明，即使没有先前的对抗性越狱，仅使用迭代方法，ChatGPT也能够开发出类似流行公司并模拟几种常用的逃避策略，以避免被钓鱼机构检测到。

不幸的是，由LLM生成的代码的安全性并未得到应有的重视。 不安全的编程可能会在下游应用中产生深远的影响。 本节讨论了一些关于LLM生成代码安全评估的最新努力，然后回顾了该领域存在的一些挑战。

3.3.1 代码生成安全研究随着LLM对公众

用户的可访问性增加，基于人工智能的工具用于辅助开发人员进行编码活动变得越来越普遍（Sadik等，2023年）。 Copilot是其中一个工具，它使用Codex，这是一个在公共GitHub存储库上训练的模型，即可能包含缺陷和漏洞的代码。 最近的研究表明，Codex复制了训练中出现的弱点，并产生了单语句错误，也称为简单愚蠢错误或SStuBs（Jesse等，2023年；Pearce等，2022年；Asare等，2023年）。在使用大型语言模型时的一个主要挑战是评估和改进生成代码模型的校准。 校准是衡量模型

信心反映准确性的度量。 一些传统技术，如Platt缩放，据说可以增强代码生成模型的校准，从而使决策更明智（Spiess等，2024年）。 然而，评估和提升模型的校准仍然是一项困难的任务。

代码生成不仅面临准确性和校准的问题，还面临学生将其用于闭卷编码任务的风险。 如果学生继续依赖大型语言模型和其他聊天机器人作为编程助手，这可能会严重损害他们的编程技能。 另一个困难是保护私人信息。 事实上，大型语言模型可能会生成直接或间接包含专有企业数据的文本。

因为一些员工使用聊天机器人来帮助他们撰写文件或代码。 由于用户和大型语言模型之间的通信存储在聊天机器人的知识库中，这可能会泄露商业机密。 对于希望保持其代码机密性的组织来说，这将是一个问题，因为涉及知识产权。

话虽如此，最近有一些研究通过安全的视角评估了大型语言模型生成的代码。 例如，Khoury等人调查了Chat-GPT生成的代码的安全性(Khoury等人，2023)。 他们的实验表明，ChatGPT经常生成不安全的代码。 问题在于ChatGPT在生成内容时根本没有考虑对抗模型。 他们的探索表明，ChatGPT在一定程度上意识到其生成的代码中存在一些关键漏洞。 在某些情况下，它甚至可能向用户提供一个有说服力的解释，说明为什么代码可能存在漏洞。 如果用户对网络安全和攻击有所了解，他们可能会提出跟进问题，以揭示代码中的进一步问题。

然而，当用户对模型进行询问时，存在泄露关键安全信息的严重风险，比如密码存储等。规避这种漏洞的一种方法是依靠单元测试来检测LLM生成的代码，并相应地更正代码（Khoury等，2023年）。 将LLMs用作教学工具，或作为交互式开发工具似乎是一个合理的用例。 然而，可能会发生LLM错误地将安全程序识别为有漏洞。 Khoury等人发现的一个有趣特性是，ChatGPT拒绝创建攻击代码，但允许创建有漏洞的代码，即使道德考虑可能是相同的，甚至更糟。 此外，在某些情况下，ChatGPT明知可能发生攻击，但无法创建安全代码，却故意创建有漏洞的代码。 在其他情况下，有时ChatGPT会误解提示中提供的请求。

Khoury等人还发现，在几种情况下，指示ChatGPT使用特定编程语言执行任务会导致不安全的代码，而在不同语言中请求相同任务则会产生安全的代码。 尽管多次向聊天机器人询问，他们仍无法理解导致这种差异的过程，因此无法制定互动策略。

最大化代码安全性的方法。

在另一项研究中（Pearce等人，2021年），Pearce等人评估了GitHub Copilot生成的代码的安全性。他们推测，由于Copilot是在GitHub上可用的开源代码上进行训练的，可变的安全质量源自社区提供的代码的性质。 也就是说，在开源存储库中某些漏洞更容易被发现，Copilot会更频繁地复制这些漏洞。然而，人们不应该就GitHub上存储的开源存储库的安全质量得出结论。

为了解决版权问题，李等人提出使用Code LLM数字水印技术，以鼓励对LLM的安全使用（李等人，2023b年）。 他们发现现有的水印和LLM生成文本检测方法在代码生成任务中无法正常运行。 失败出现在两种模式中：要么1）代码未正确加上水印（因此无法检测），要么2）带有水印的代码无法正确执行（质量下降）。他们提出了SWEET，一种新的水印方法，通过引入选择性熵阈值过滤对执行质量最不相关的标记，从而在一定程度上解决这些失败情况。事实上，使用SWEET的实验结果并没有完全恢复原始的非水印性能；然而，他们认为这是朝着实现这一雄心勃勃的目标迈出的重要一步。

在另一项研究中，Sandoval等人（Sandoval et al., 2022）调查了LLM代码建议对参与代码编写研究的参与者的网络安全影响。 他们得出结论，LLM对功能正确性可能有积极影响；并且不会增加在涉及指针和数组操作的低级C代码中严重安全漏洞的发生率。 鉴于已有的关于LLM建议代码容易受攻击的研究（Pearce等人，2021），这有些令人惊讶。 在考虑发现的错误的来源时，数据表明用户并未利用额外的生产力优势来修复代码中的错误 - 尽管建议正在被修改，如果一个建议包含错误，则可能不会被修复。 这表明需要进一步研究以突出问题代码行，鼓励用户实时检查安全性。 此外，LLM的代码应该得到改进，以生成更安全的代码

而不是用户现有的代码（Siddiq等人，2022年）。

3.3.2 代码生成安全研究中的挑战

尽管前述所有工作，但是评估LLM生成的代码安全性存在挑战。根据Siddiq等人（Siddiq等人，2022年）进行的实验，此类挑战的非穷尽列表如下：可复现的代码生成：在大多数情况下，包括Copilots在内的生成模型的输出并不直接可复现。 事实上，对于相同的提示，Copilot可能在不同时间生成不同的答案。 由于Copilot通常是由远程服务器上的API提供的黑匣子模块，外部人员无法直接检查用于生成的模型。

生成大型语料库和统计有效性的限制:往往存在诸如令牌速率或解码样本数量等限制，特别是当LLMs驻留在远程服务器上时。 这使得生成大型数据集，这对进行任何统计上有意义的分析是必要的，变得极具挑战性。

场景创建的限制: 对于安全评估，我们通常需要人为设计一些安全测试场景来识别潜在的弱点。 然而，由于真实世界的代码在上下文方面（如类、函数、库等）要大得多，因此合成设计的场景可能无法完全反映现实世界的软件。

生成对提供提示的敏感性:正如我们之前讨论的，即使提示中有微小的变化，也会影响LLM生成的代码。 通常，通过安全代码示例提供上下文和演示，会产生更安全的代码。 然而，大型语言模型对提示的敏感性使得生成的见解高度依赖于提示工程。 因此，一个给定的代码可能通过特定的测试场景，但如果我们操纵提示，它可能在同一场景中失败。

对编程语言的敏感性区分LLM固有安全漏洞和编程语言相关的弱点非常重要。 例如，一些编程语言通过封装和自动内存管理提供更安全的代码。 如果测试场景足够复杂，这种区分可能并不是一项微不足道的任务。

尤其是在与黑盒LLM一起工作时。

网络安全的演变性质在任何网络安全研究中存在的另一个挑战是时间因素。 在代码生成时被视为‘安全实践’的做法，可能由于网络安全研究的演变性质逐渐变成‘不安全实践’。 这种进化方面影响了安全代码生成管道的所有模块，如训练数据和评估指标。 例如，密码哈希在一段时间内发生了相当大的演变。 多年前，MD5被认为是安全的，然后被单轮SHA-256取代。如今，最佳实践甚至进一步发展。 重新审视测试场景，重新设计它们并重新评估结果都是耗时且昂贵的必要性。

解决我们上面列举的每一个挑战都是提升LLM网络安全的可能途径。通过克服这些挑战，我们可以利用LLM网络在各种应用中的好处，同时最大程度地减少它们可能给个人、组织和整个社会带来的风险和危害。

4 对抗性攻击和LLM网络漏洞

最近关于LLM网络的研究强调了它们的弱点，特别是对抗性攻击的漏洞（Mozes等，2023年）。
开放式网络应用安全项目（OWASP）已经整理了LLM应用程序中经常观察到的前10个关键漏洞的列表 [3]。 这些发现突显了在实际场景中部署LLM时谨慎行事的重要性。

提示注入、数据泄漏、不足的沙箱等漏洞的例子显示了在实际应用中利用LLM是多么简单。 为了更清晰和结构化地呈现LLM中的漏洞，我们将这些漏洞分为三大类：基于模型、训练时和推理时漏洞。 每个类别对应于针对LLM生命周期不同方面的特定攻击。

[3] https://owasp.org/

表2：不同作品采用的不同记忆策略。 每个LLM记忆策略都显示了使用在其上的模型以及所采用的记忆类型。

论文	记忆标准	模型	数据集
(Carlini等人，2021)	个人姓名，电子邮件地址，电话号码，传真号码，和实际地址	GPT-2	无
(de Wynter等人，2023)	提示回忆	BLOOM，ChatGPT，Galactica，GPT-3，GPT-4，OPT，OPT-IML和LLAMA	无
(Kuchnik等人，2023)	正则表达式URL	GPT-2XL	LAMBADA

4.1 基于模型的漏洞 这些漏洞源

于LLM的固有设计和架构。突出的例子是模型提取和模型模仿攻击。 在本节中，我们简要讨论这种类型的攻击。

4.1.1 模型提取攻击 基于LLM的

服务容易受到模型提取攻击的影响。 这些攻击涉及通过大量查询复制模型的功能，这对其独特性和知识产权构成威胁。 这种攻击可能导致模型所有者遭受重大损失。 考虑到训练大型语言模型是一个昂贵的过程，这些提取攻击可能严重影响模型的完整性和安全性。

随着最近的进展，包括变换器在内的大量预训练语言模型现在可用于创建API。在考虑模型提取攻击时，一个“受害模型”的概念就出现了。 如果受害模型配备有API，一个秘密用户可以查询受害模型并近似其行为。

提取模型的常见方法包括从受害模型构建一系列查询-预测元组。 随后，这个集合被用来近似受害模型。 在模型提取方面，存在几种方法。 在接下来的内容中，我们将探讨其中一些问题。

例如，*EmbMarker*（P*e*ng等人，2023）是一种利用基于后门的水印技术来提取模型的方法。 通过嵌入微妙的标记，它允许在保留模型功能的同时进行模型提取。

Mondarin（S*i*等人，2023）是另一种方法，它专注于API级别，提供了与其他服务相比较廉价的API。其目标是为寻求LLM服务的用户创造一种具有成本效益的替代方案。

此外，还有一种特定类型的模型提取，即“模型寄生”（Birch等人，2023b），攻击者查询“受害模型”以从中提取知识。随后，偷偷摸摸的用户利用这些提取的信息来训练自己的模型。

模型寄生的主要目标是从受害模型中获取见解，而不直接访问其内部参数或架构。 基本上，它使攻击者能够创建一个新模型，该模型近似于原始受害模型的行为。 这种技术通常用于测试对抗性攻击或开发替代服务。

值得注意的是，这种方法允许对对抗性攻击进行无限制测试，但其有效性严重依赖于提示的质量。 换句话说，不足够的提示会使寄生模型失效。

在Si等人（Si等人，2023年）最近的一项研究中，作者从一个新的角度研究了模型提取攻击。 他们的目标是通过利用原始LLM及其API，设计出一个比现有LLM API服务更便宜的替代方案。主要思想是减少发送给原始LLM API的输入提示大小，从而最小化利用它的成本。这种技术有效地整合了输入提示，使恶意用户能够向毫无戒心的用户提供更实惠的语言模型服务。

4.1.2 模型模仿

随着新型大型语言模型及其相关的API的兴起，“模型模仿”的概念日益受到重视（Gudibande等，2023年）。 这种做法涉及通过API调用收集数据集，然后利用这些获取的数据对自己的模型进行微调。特别是，这种现象

这种现象对于旨在通过利用专有LLM的输出来实现与专有LLM相媲美性能水平的开源LM是相关的。
包括Alpaca（Taori等，2023年）、Vicuna（Chiang等，2023年）和Koala（Geng等，2023年）在内的几项研究工作已经报道了成功模仿专有LLM在性能方面的尝试。

然而，必须承认，虽然开源LM可以从吸收专有对手的见解中受益，但某些限制仍然存在，特别是在事实性、编码和问题解决等领域。

例如，古迪班德等人（古迪班德等人，2023年）的一项研究表明，开源语言模型可以从使用专有语言模型来改进它们中受益，但在事实性和问题解决方面仍然落后。因此，类似于模型提取，从大型语言模型中获取关键见解的努力仍处于早期阶段。

上述作品展示了模型提取和模仿的一些方面。随着大型语言模型变得越来越普遍，了解恶意用户可以实现的影响至关重要。然而，对于基于模型的攻击和有效的防御机制，存在着大量机会进行进一步研究。

4.2 训练时漏洞 这一类别涉及在模型训练阶段引入的漏洞。关键问题包括数据中毒，即恶意数据被插入到训练集中，以及后门攻击，其中隐藏的触发器被嵌入到模型中。在本节中，我们详细讨论这些攻击。

4.2.1 数据毒化

数据毒化的概念在机器学习领域，特别是在自然语言处理模型中，引起了新的关注。这种攻击形式，即将恶意数据巧妙地引入AI模型的训练集中，会产生隐藏的漏洞，可能危及关键系统的完整性和功能性。

Wallace等人（Wallace et al., 2021）深入探讨了这个阴暗领域，揭示了自然语言处理模型中数据毒化的隐秘危险。他们引入了一种新的攻击方法，其中精心设计的触发短语嵌入到训练数据中-

允许攻击者以有针对性的方式操纵模型输出。这种基于梯度的方法，针对文本数据进行了精细调整，能够规避传统的检测方法。它在各种自然语言处理任务中展示出其强大的潜力，将无害的术语如“詹姆斯·邦德”转化为偏颇情感分析的催化剂，或将“苹果iPhone”转化为负面语言模型输出的触发器。

这些方法的微妙和有效性要求重新评估自然语言处理模型的防御，Wallace等人提出了一组策略，虽然有效，但也伴随着各自的权衡。

在像ChatGPT这样的特定上下文中，Wan等人（Wan等人，2023）调查了它们对数据毒化的敏感性。
他们揭示了在训练数据中包含少量毒化样本如何导致模型输出中的一致性、有针对性的错误。这一发现尤其令人担忧，考虑到在训练这些模型时用户生成的内容的普遍性。Wan等人的实验表明，嵌入约100个毒化示例可以扭曲各种任务的输出，揭示了一个“反比例”现象，即更大的模型更容易受到这种攻击形式的影响。他们的发现强调了在大型、指导调整模型时代对数据审查和强大训练方法的迫切需求。

为了对抗这些威胁，Prabhumoye等人（Prabhumoye等，2023年）提出了旨在减少预训练语言模型中毒性的创新数据增强技术。通过将直接毒性评分或描述性语言说明整合到训练数据中，他们实现了对有毒模型输出的显著减少。

这种策略应用于Megatron-LM模型，导致毒性水平显著降低，而不会影响标准NLP任务的准确性。他们的方法提出了AI训练中的新范式，即直接将道德考虑因素整合到训练过程中可以产生更安全、更负责任的AI模型。

这些研究的集体发现揭示了人工智能安全和伦理面临的新挑战。随着机器学习和NLP模型越来越深入地融入我们的数字基础设施，保护它们免受隐蔽数据毒害攻击的需求变得日益关键。解决这些挑战需要多方面的方法，融合技术

创新与政策制定和用户教育。 Wallace等人，Wan等人和Prabhumoye等人的研究强调了AI发展需要平衡的方法，安全和道德考虑与效率和可扩展性同等重要。

总之，机器学习中数据中毒的不断演变的格局涵盖了AI安全和道德的更广泛问题，呼吁全面和积极的应对，以确保这些强大工具的安全和道德部署。 表3显示了选定的训练时间数据中毒攻击和缓解技术的摘要。

4.2.2 后门攻击

后门攻击对LLMs的安全构成严重威胁。这些攻击涉及在LLM的训练阶段秘密植入触发器。 在推理过程中激活时，此触发器会导致模型生成特定的，通常是有害的输出或行为。 这些攻击尤其危险的地方在于它们能够避开检测并保持休眠状态，直到触发，绕过标准的安全措施。

一类后门攻击的重点是操纵输入空间。 具体来说，这些攻击涉及将特定的触发机制嵌入到模型提示中。 这些触发器的示例包括使用不常见的词语（Chen等，2021年）或句法结构（Qi等，2021年），短语（Xu等，2022a年）等。减轻这类攻击的一种方法是识别和理解触发器本身。

一种称为 *Bad-Prompt*的任务自适应后门技术（Cai等，2022年），例如，会自动生成最适合每个个体样本的触发器。 *BadPrompt*包括两个阶段：触发器候选生成和自适应触发器优化。 在触发器候选生成阶段，根据它们与目标标签的相关性和与非目标样本的不相似性，从受污染的输入数据集中选择触发器。 这个阶段产生了一组触发器候选集。 在第二阶段，自适应触发器优化确定了每个个体样本最合适的触发器，认识到一个常见的触发器可能并不对所有样本都同样有效。 最后，模型使用干净和被毒化的数据进行训练，优化背门攻击目标。

这种方法的有效性已经得到证明在各种分类任务和受害模型中展示了其有效性，包括PaLM、RoBERTa-large以及两个连续提示模型：P-tuning（刘等，2021）和DART（张等，2021）。 根据这项研究，BadPrompt实现了高准确率，即使训练数据中毒率降低也能保持稳健。

尽管这些技术有效，但它们存在一个共同的缺点：使用触发器可能导致异常的语言表达，使其容易被防御算法检测到。

为了解决这个问题，ProAttack（赵等人，2023a）采用了一种不同的方法，即一种干净标签的后门攻击方法。 它不依赖于显式的外部触发器，而是通过提示本身诱使模型学习触发模式。
具体来说，像BERT-large、RoBERTa-large、XLNET-large和GPT-NEO-1.3B这样的语言模型都容易受到这种攻击，其中GPT-NEO-1.3B是最容易受攻击的模型。赵等人假设提示可以作为后门攻击的触发器，这与不同提示诱使模型学习不同特征表示的观察相一致。

另一类后门攻击的目标是嵌入空间。 输入空间攻击通常具有有限的可转移性，因为它们将后门注入到词嵌入向量中。 因此，在不同任务上重新训练并采用不同提示策略后，它们的效果会减弱。 为了使攻击机制更具普适性，一个替代方案是向预训练的大型语言模型的编码器部分注入后门。

例如，NOTABLE（Mei等，2023年）利用自适应的语言化器将触发器绑定到特定单词，使攻击独立于下游任务和提示策略。 研究表明，与其他后门攻击（如BToP（Xu等，2022b）和BadPrompt（Cai等，2022年）相比，NOTABLE在三个不同的分类任务上实现了更高的ASR。我们列出的所有攻击都突显了大型语言模型在各种任务中的脆弱性，并引起

了社区创建适当的缓解和防御机制的关注。 缓解措施可以在几个阶段进行，包括在预处理步骤中基于特征分布隔离受污染的样本，扩展对抗训练在预训练中，或微调（Liu等，2018年）和知识蒸馏（Li等，2021年）等。

表3：训练时间数据投毒攻击和缓解技术

论文	主要思想	触发示例/方法	对模型的影响	缓解技术
(Wallace等人，2021)	使用基于梯度的机制对文本数据进行隐蔽数据投毒	“詹姆斯·邦德”将情感转为积极，“苹果iPhone”引发负面反应	模型的预测受特定短语控制，影响各种自然语言处理任务的可靠性	过滤方法；模型容量减少；预测准确性和增加人工监督之间的权衡
(Wan等人，2023)	研究指导调整的大型语言模型中的数据投毒，特别是用户生成内容	微妙的引入毒害数据；在评估过程中检测到触发器，导致一致的错误	在翻译和摘要等任务中的错误分类；更大的模型更容易受到投毒攻击	增强用户生成的数据审核；自适应训练方法论；平衡模型大小和易受攻击性
(Prabhumoye等人，2023)	通过创新的数据增强减少语言模型中的毒性	将原始毒性评分和描述性语言指令纳入训练数据	毒性水平显著降低；在标准NLP任务中保持性能；改进偏见检测	将毒性指标直接整合到训练数据中；专注于预训练阶段以减轻毒性而不影响性能

4.3 推理时漏洞 这一类别关注的是模型与最终用户或系统交互过程中出现的漏洞。 它包括一系列攻击，包括越狱、释义、欺骗和提示注入，每种攻击都以不同方式利用模型的响应机制。

4.3.1 改写和欺骗攻击 改写攻击是一种对大型语言模型进行的敌对攻击，攻击者使用改写模型修改输入文本，以不同措辞重新陈述文本，同时保留整体含义。 这些攻击的主要目标是规避依赖于LLM生成文本中特定签名或模式的检测或过滤机制。

此外，释义攻击可能被滥用于恶意目的，如抄袭和生成误导性内容（Krishna等，2023年；Sadasivan等，2023年）。

例如，攻击者可以使用释义器删除用于识别LLM输出的水印或风格特征（Sadasivan等，2023年）。 释义攻击也可以用于绕过基于检索的防御措施，这些措施将输入文本与已知人类文本数据库进行比较，并标记那些太相似的文本。

通过改写输入文本，攻击者可以降低相似性分数并避免被检测到（Sadasivan等，2023年）。

欺骗攻击是指对抗LLM或其创建者的对手使用修改或定制的LLM来生成类似输出。 欺骗的LLM可以被操纵以产生有害、误导或不一致的输出与其预期功能或声誉相悖。

例如，欺骗的LLM聊天机器人可以产生冒犯性或虚假陈述，并泄露敏感信息。 欺骗攻击可能危及基于LLM的系统的安全和隐私（Shayegani等，2023年）。

检测LLM上的改写和欺骗攻击极具挑战性，因为这些攻击利用了语言的固有歧义性。 然而，有一些提出的策略可以保护LLM免受此类攻击。

Jain等人（Jain等人，2023）提出的一个简单解决方案是在将输入文本馈送给LLM之前应用一个释义器或重新标记化，以消除对抗性扰动并恢复原始含义。 然而，这种方法可能会导致在输入文本中引入噪音或错误，并且可能对强大的释义攻击不起作用。

另一种技术是使用基于困惑度的策略，衡量输入文本的可能性，并标记那些困惑度较低的文本为可疑输入（Jiao等人，2023）。

例如，胡等人提出了一种基于标记级别的检测方法，通过预测下一个标记的概率来识别对抗性提示，测量模型的困惑度，并添加邻近标记信息来增强检测（胡等人，2023）。

另一种训练时防御机制是对抗性训练，它通过增加训练数据中的改写查询和相应的答案来增强训练。 通过向大型语言模型暴露各种输入，对抗性训练可以帮助模型更好地泛化，从而更好地抵抗改写攻击。

措辞攻击（焦等，2023年）。

总结一下，我们可以将防御机制对抗改写和欺骗攻击分类为预处理、训练时间和推理时间（例如检测）策略。 鉴于许多人工智能生成文本检测算法对这些攻击的高易受攻击性，正如Sadasivan等人所示（Sadasivan等人，2023年），开发更加健壮和有效的防御技术对抗此类攻击至关重要。

4.3.2 大型语言模型中的提示注入和泄漏提示

在语言模型中的操纵，包括注入和泄漏，对现代大型语言模型的安全和隐私构成严重威胁。基本上，这些漏洞使对手能够劫持模型的输出甚至暴露其训练数据。

提示注入是指对手故意构造输入数据，利用模型现有的偏见或知识，以产生针对性或欺骗性输出。 另一方面，提示泄漏是这种攻击的更专注变体，涉及以使模型在响应中原样重复其原始提示的方式查询模型。

一种常见的提示注入策略是通过向提示中添加常见词、不常见词、符号、句子等触发器来欺骗LLMs。 例如，为了攻击少样本示例， *adv ICL*（Wang等人，2023年）利用词级扰动，如字符插入、删除、交换和替换。

在非训练环境中，唐等人（唐等人，2023年）研究了集成内容学习（ICL）的韧性，并探讨了LLMs依赖提示快捷方式的程度。在另一项工作中，徐等人（徐等人，2022年）利

用波束搜索技术识别减少LLMs准确预测掩码词的可能性的触发器。 这种技术基于攻击者可以访问公共LLMs并寻找触发器的假设。研究表明，诸如GPT和LLaMA系列的LLMs容易受到这种类型的攻击。 LLMs对攻击的高易感性使得解决和减轻触发器中毒变得更加困难。

诸如询问或消除提示中的每个标记并评估其对后续任务的影响的技术，是常见的检测方法之一（Ribeiro等，2016年；Qi等，2020年）。

减少触发器负面影响的另一个有前途的缓解策略是过滤导致性能下降的异常标记（Xu等，2022年b）。

在快速发展的LLMs领域中，Greshake等人（Greshake等人，2023年）引入了一种新的威胁：“间接提示注入”。 在这种情况下，对手巧妙地将提示嵌入到LLMs访问的外部资源中，例如网站。 这种方法标志着与传统的直接与LLMs互动即远程利用它们的分离。 这种攻击带来了重大风险，包括数据窃取、恶意软件传播和内容篡改。 这一揭示突显了对LLM利用方法的重大转变，扩大了潜在漏洞的范围。

在prompt操纵概念的基础上，刘等人（刘等人，2023a）探讨了整合LLMs的商业应用程序中的漏洞。他们的研究确定了启发式攻击方法的缺陷，导致HOUYI的开发。这种结构化方法，从传统基于网络的攻击策略中汲取灵感，通过成功地通过prompt操纵来妥协多个服务，展示了其有效性。 HOUYI的引入标志着在商业应用程序中理解和对抗prompt注入漏洞的一个关键步骤。

康等人（Kang等人，2023年）深入探讨了LLM领域，特别关注擅长遵循指令的模型，例如Chat-GPT。他们强调了一个讽刺的转折：这些模型的增强指令遵循能力无意中增加了它们的脆弱性。 这些LLM在接受策略性设计的提示时，可能会被操纵生成有害输出，如仇恨言论或阴谋论。 康等人的这一观察为围绕LLM的安全问题增加了一层复杂性，暗示它们的先进能力也可能是它们的致命弱点。

佩雷斯和里贝罗（Perez和Ribeiro，2022年）则专注于提示操纵的一个特定方面，即提示泄露。 他们展示了LLM（如GPT-3）如何通过目标劫持或揭示机密训练提示而偏离其预期功能。 他们开发的PROMPT-INJECT框架成功地绕过了内容

OpenAI的过滤防御措施的漏洞，突显了他们在操纵LLM行为方面的有效性。

然而，提示操纵的影响不仅仅局限于劫持模型输出。 随着LLM的发展，人们对意外数据记忆和暴露产生了担忧。Kim等人（Kim等人，2023年）通过引入ProPILE框架来解决这些隐私问题。 这个工具允许利益相关者评估LLM中个人可识别信息（PII）泄露的风险。ProPILE在揭示潜在PII泄露的广泛范围方面的实用性标志着在LLM部署中保护隐私方面的重要进展。

总的来说，这些研究全面描绘了LLM中提示操纵所涉及的挑战和风险。它们强调了对这些先进AI系统的技术能力和潜在漏洞的细致理解的重要性，强调了在面对不断演变的威胁时开发强大的安全和隐私措施的重要性。

表4概括了一些重要的作品，涉及到对提示注入攻击的处理。

4.3.3 越狱隐私攻击“越狱”LLM的现象代表了技术创新和新兴安全挑战的重要交汇点。 这一过程涉及操纵输入提示以规避内置的安全和审查功能，引发了人们对这些先进AI工具的安全、隐私和道德使用的重大关注。

研究人员积极探索这一领域，揭示了不同LLM对复杂越狱方法的不同易感性水平。例如，李等人(Li et al., 2023a)指出，虽然ChatGPT对直接提示攻击表现出抗性，但仍然容易受到多步越狱提示(MJPs)的攻击，这些攻击可以提取诸如电子邮件地址之类的敏感数据。 相比之下，New Bing更容易受到旨在提取个人信息的直接提示攻击，突显了不同LLM平台之间防御机制的差异。

进一步复杂化这一格局，邓等人（邓等人，2023a）介绍了JAILBREAKER，这是一个全面的框架，相较于ChatGPT，在Bard和Bing Chat中揭示了更先进的防御技术。这些大型语言模型采用实时关键词过滤，类似于基于时间的SQL注入防御，以阻止潜在的越狱尝试。 JAILBREAKER在使用精细的大型语言模型生成越狱提示方面的创新方法，展示了这些攻击的不断演变的复杂性。

越狱策略的动态性在沈等人的研究中进一步得到证实（沈等人，2023），他们分析了数千个真实世界的提示。 他们的发现表明，攻击者正在向更加隐蔽和复杂的方法转变，从公共领域迁移到私人平台。 这种演变使积极检测工作变得更加复杂，并凸显了攻击者日益增长的适应能力。

沈等人的研究还揭示了一些越狱提示的高效性，实现了在像ChatGPT和GPT-4这样的平台上高达0.99的攻击成功率，并强调了越狱提示所构成的威胁格局的不断演变。

作为对这些威胁的回应，饶等人（饶等人，2023年）提出了一种结构化的越狱提示分类法，根据语言转换、攻击者意图和攻击方式对其进行分类。 这种系统化的方法突显了对持续研究和开发自适应防御策略的必要性，以及理解攻击意图的广泛类别的重要性，比如目标劫持和提示泄露。

这些研究的集体见解强调了在人工智能领域创新和安全之间采取平衡方法的必要性。随着大型语言模型越来越多地融入我们数字生活的各个方面，确保它们的道德和安全部署至关重要。 这个挑战不仅仅是一个技术问题；它还需要政策制定和用户教育来减轻与这些强大人工智能工具相关的风险。

总之，越狱大型语言模型提出了一个复杂且不断发展的挑战，涵盖了人工智能安全和伦理方面的更广泛问题。 应对这一挑战需要多方面的方法，将技术创新与对攻击者不断演变的策略的全面理解相结合。 随着我们在依赖大型语言模型方面的进展，保护这些系统免受滥用变得越来越重要。表5提供了关于越狱隐私攻击的研究的简明总结。

表4：关于提示注入攻击的选定作品摘要

论文	攻击名称/类型	目标模型	主要目标	应用/平台
(Greshake等，2023)	间接提示注入	各种大型语言模型	通过外部内容来源利用大型语言模型	必应的GPT-4动力聊天，其他集成了大型语言模型的系统
(刘等人，2023a)	HOUYI（黑盒提示注入）	商业LLM	使用上下文分离的系统化提示注入	多个商业应用
(康等人，2023)	通过指令跟随进行恶意操纵	指令跟随的LLM（例如，ChatGPT）	通过绕过内容过滤器生成恶意内容	OpenAI API, ChatGPT
(佩雷斯和里贝罗，2022)	PROMPTINJECT（目标劫持，提示泄露）	GPT-3	绕过内容过滤防御，操纵LLM行为	OpenAI的GPT-3
(金等人，2023)	ProPILE（隐私泄露评估）	在公共数据集上训练的LLM	评估LLM中个人身份信息泄露的风险	通用基于LLM的服务

LLM的5个风险和误用

LLM有潜力产生有害内容或促进恶意活动，如传播有毒、偏见、有害的语言和错误信息，从事抄袭和发动网络安全攻击。 在接下来的部分中，我们将概述与LLM误用相关的潜在风险的全面但非详尽的汇编。此外，我们将讨论减轻这些风险的推荐策略，并探讨实施这些策略所固有的挑战。

5.1 LLM响应的事实不一致性和不可靠性

在推理时保持事实一致性是LLM面临的关键困难之一。 LLM倾向于在给定请求中出现条件忽视、误解和幻觉。

例如，在最近一项研究中检查GPT-3（Khatun和Brown，2023）时，研究人员发现，虽然该模型能够过滤明显的阴谋论和刻板印象，但在处理日常误解和讨论时却表现不佳。 模型的响应在不同查询和情况下表现出变化，突显了GPT-3固有的不可预测性。同样，周等人的研究（周等人，2024年）揭示了LLM（如ChatGPT和Claude）在回答问题时未能传达

不确定性。 包括ChatGPT和Claude在提供答案时很难传达不确定性。 令人惊讶的是，这些模型即使在回答错误时也可能表现出过度自信。 虽然可以提示LLMs表达信心水平，但这种方法通常会导致高错误率。 此外，研究突出了一个关键挑战：用户发现很难-

由于模型的语调和风格引入的偏见，很难评估LLM响应的正确性。 这个问题尤为重要，因为针对不确定文本的偏见可能会影响LLM的训练和评估。

在另一项由Laban等人进行的研究中（Laban等人，2023年），LLM作为事实推理者的能力通过文本摘要中的事实判断进行了调查。观察到LLM在表面上表现与专门的非LLM评估者类似，但在更复杂的评估场景中，性能显著下降。

在类似的研究中，Laban等人（Laban等人，2023年）通过提出一个名为SUMMEDITS的新评估基准程序来检验LLM响应的不一致性，结果显示大多数现有的LLM，包括最佳模型GPT-4，仍然不及人类表现，难以生成一致的响应。

然而，为了减轻这种错误，通过微调（Lewkowycz等人，2022年；Rajani等人，2019年；Zelikman等人，2022年）、提示工程技术如验证、草稿本（Cobbe等人，2021年；Nye等人，2022年）、思维链（CoT）（Wei等人，2022年）、RLHF（Ziegler等人，2019年；Christiano等人，2017年）、迭代自我反思（Shinn等人，2023年；Madaan等人，2023年）等各种策略已被提出。
修剪真实数据集（Christiano等人，2023年），外部知识检索（Guu等人，2020年）以及基于似然估计的无需训练的方法（Kadavath等人，2022年）。

例如，Wang等人提出了一种新的提示方法，即自一致提示（Wang等人，2023年），该方法采样了一组多样化的推理路径，而不仅仅是

选择贪心的答案，然后通过边缘化采样推理路径来选择最一致的答案。 这种方法背后的原理很简单：一个复杂的推理问题通常有多种不同的思考方式，导致其独特正确答案（Wang等，2023年）。

尽管已经引入了各种方法来减轻不一致性，但只有少数几种方法能够有效地确定LLM提供的响应是否准确。为了解决这个问题，薛等人最近开发的一种名为逆向思维链（RCoT）的方法旨在自动检测事实上的差异并修复LLM生成的文本中的错误。为此，RCoT利用模型的输出、说明和示例来重建问题。 它将原始问题和重建问题分解成详细的条件列表，将它们进行比较，以识别任何幻觉、疏忽、误解或事实分歧的情况。 当事实不一致出现时，RCoT会生成细致的反馈，随后指导LLMs更新他们的解决方案以纠正问题。

“心智社会”策略是增强LLMs事实性的另一种新颖方法（杜等，2023b）。 在这种策略中，多个语言模型实例在多轮中展示并辩论他们自己的回应和推理过程，以找到一个共同点。杜等人表明，这种方法在各种任务中显著改善了数学和战略推理（杜等，2023b）。 他们还说明，这种方法通过消除LLMs中常见的错误答案和幻觉，增加了生成信息的事实质量。有趣的是，LLMs本身可以用来评估语言模型的一致性。

例如，Tam等人（Tam等人，2023年）通过引入一个事实不一致基准（FIB）来做到这一点，用于总结任务。 他们比较了大型语言模型为给定新闻文章分配的得分，这些得分是针对一个事实一致与一个事实不一致的摘要而言的。 他们在这个基准上评估了多个大型语言模型，并发现大型语言模型倾向于为事实一致的摘要分配更高的分数，而不是为事实不一致的摘要分配更高的分数。

调整系统参数以限制模型创造力、整合外部知识源以改进答案验证、生成理由和参考文献等技术是改进大型语言模型响应的其他方法之一（Muneeswaran等人，2023年）。

前面提到的所有研究都强调，尽管大型语言模型是非常强大的工具，但它们仍然极易出错。 因此，任何由大型语言模型产生的输出都应该谨慎对待。

5.2 歧视、毒性和危害由LLMs生成

LLMs可能生成具有歧视性、冒犯性或对个人或群体有害的语言，这取决于它们的训练数据的质量和多样性、设计选择以及其预期或意外的应用（Gehman等，2020年； Deshpande等，2023年； Cui等，2023年）。 因此，LLMs带来了需要仔细评估和监管的伦理和社会挑战。

DeepMind发表的一项研究（Weidinger等，2021年）对LLMs相关的风险景观进行了结构化。它概述了六个具体的风险领域，包括歧视、排斥和毒性，并讨论了潜在的缓解方法和挑战。 它进一步探讨了缓解这些风险的潜在策略，强调了增强数据质量和多样性、使用公平度量标准以及建立内容管理和报告机制等实践。

由ousidhoum等人撰写的一项工作介绍了AttaQ，这是一个包含问题形式的对抗性示例的新数据集，旨在引发大型语言模型产生有害或不当回应。他们在这个数据集上评估了几个大型预训练语言模型（PTLMs），发现在许多情况下，LLMs会产生不安全的回应。

另一项由Deshoande等人（Desh- pande等人，2023a）进行的研究揭示，当ChatGPT被赋予一个人设时，它可能表现出相当大的毒性，并对脆弱人群（如学生、未成年人和患者）构成风险。 毒性程度根据选择的风格而有显著差异，当ChatGPT明确被要求说负面的事情时，有害内容显著增加。

此外，研究发现特定性别和种族面临更高风险遭遇有害内容。Deshoande等人提出，这种现象源于模型过度依赖RLHF来减轻毒性。 向模型提供的反馈可能带有偏见，

表5：越狱隐私攻击选定作品摘要

论文	研究重点	方法论	关键发现	贡献
(李等人，2023a)	大型语言模型的隐私威胁	对直接和多步越狱提示进行了广泛实验	ChatGPT对直接提示表现出抗性，但对多步提示易受攻击。 由于与搜索引擎集成，新Bing更容易受到直接提示的影响。	探讨了LLMs和应用集成LLMs的隐私影响，揭示了不同的漏洞。
(邓等人，2023a)	大型语言模型聊天机器人的越狱防御	越狱防御框架	Bard和Bing Chat使用高级防御技术，如实时关键词过滤。越狱者在生成越狱提示方面取得了更高的成功率。	引入了一种新颖的方法来理解和规避LLM的防御措施，深入了解聊天机器人防御的本质。
(沈等人，2023)	越狱提示的分析	在真实数据上进行自然语言处理和基于图的社区检测	越狱提示正变得更加谨慎和有效，从公共平台迁移到私人平台。 一些提示取得了很高的攻击成功率。	进行了对越狱提示的第一项测量研究，突出了不断演变和严重的威胁形势。
(饶等人，2023)	对越狱提示进行分类和分析	基于语言转换、攻击者意图和攻击方式的分类法	展示了不同LLM上越狱方法的不同有效性，突显了对强大防御的需求。	提出了一种结构化方法来对越狱提示进行分类和理解，有助于开发自适应的防御策略。

潜在地导致对不同性别的毒性评估产生偏见

值得注意的是，大型语言模型具有生成难以被现有分类器轻易检测到的隐含有毒回应的能力。 这些回应虽然没有明显的危害，但仍可能通过暗示负面或虚假陈述而冒犯或伤害个人或群体。 这对自然语言生成系统的安全性和可靠性构成严重威胁，同时也引发了重要的社会和伦理关切。

同样，文等人最近的一项研究调查了大型语言模型如何生成难以被现有毒性识别器检测到的隐含有毒输出。 该研究引入了基于强化学习的方法来揭示和突显大型语言模型中的隐含毒性。此外，它建议使用攻击方法获取的标注示例对分类器进行微调，以增强其检测此类毒性的能力。

考虑到从用户行为到数据质量和模型特征等多种因素对有害内容的生成产生影响，深入研究LLM的影响和毒性变得至关重要。开发预防、检测和缓解的强大方法和机制至关重要。这些研究努力不仅增强了LLM的安全性和可靠性，还推动了其他相关领域的进步。

5.3 LLM生成的文本、版权侵权和抄袭LLM可能成为学术写作的

重要威胁，增加了版权侵权和抄袭的风险。例如，作者可能使用LLM生成文章，而不是从头开始写作，或者学生可能使用LLM完成作业，这破坏了学术诚信，违背了作业和考试的目的（Khalil和Er，2023年；Stokel-Walker，

2022),

为了解决这个问题，已经开发了各种检测器，用于区分人类编写的文本和人工智能生成的文本。 这些检测器可以分为黑盒（Wang等，2023年；Quidwai等，2023年；Liu等，2023年）和白盒检测方法（Vasilatos等，2023年）。

在黑盒检测中，只能访问由LLMs生成的输出文本。这些检测器通常利用LLM将人类编写的文本和人工智能生成的文本嵌入到高维向量空间中。 然后，这些嵌入的文本作为轻量级机器学习分类器的区分特征。

例如，Quidwai等人（Quidwai等，2023年）提出了一种通过使用*text-embedding-ada-002*将答案嵌入到向量空间中来检测人工智能生成的抄袭的框架。
他们使用共现计算人机（HM）答案对和机器-机器（MM）答案对的句子级相似度得分。

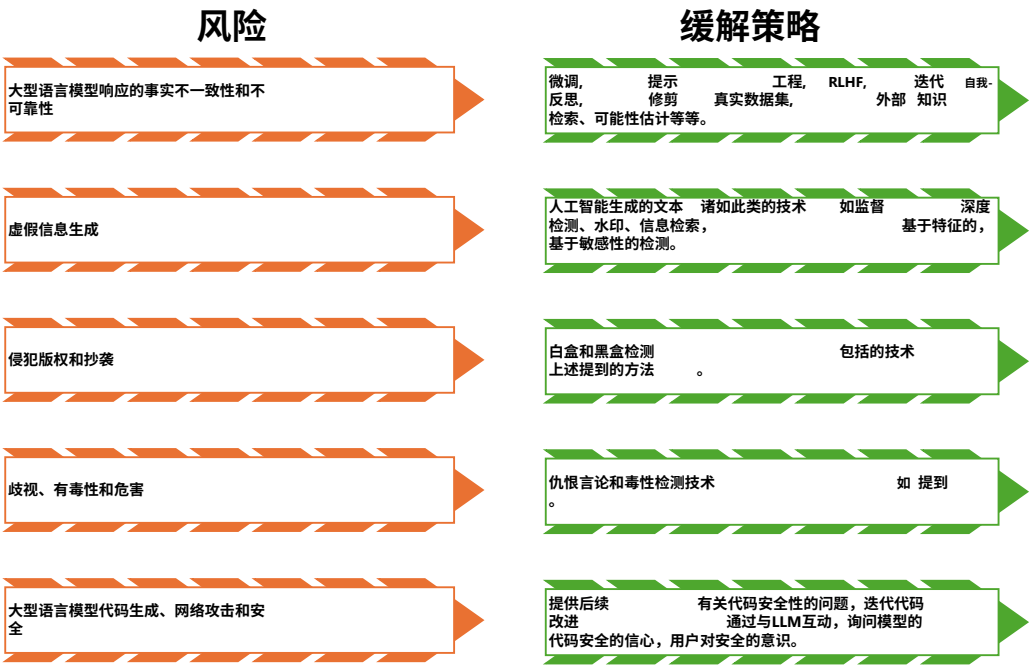

图2：LLM的风险和缓解策略摘要。

计算正弦相似度，并应用线性判别分析（LDA）分类器来决定是HM还是MM对。

同样，刘等人（刘等人，2023b）利用ChatGPT和预训练的LLM来计算摘要的嵌入表示，然后利用LSTM进行分类。 这些检测器在区分人类编写和人工智能生成的文本方面表现出高准确性。 然而，值得注意的是，由于嵌入计算所需的计算资源，这些分类器可能面临可扩展性挑战。

在同一条线上，刘等人（Liu et al., 2023b）评估了现有GPT检测器，即GPTZero（Tian, 2023）、ZeroGPT（Zer, 2023）和OpenAI的检测器（Ope, 2023）在新的基准数据集上的性能。 他们观察到，GPTZero和ZeroGPT都倾向于将输入的摘要分类为“人类撰写”。

另一方面，OpenAI的检测器在检测GPT生成的摘要方面表现显著优于其他检测器，但在检测人类撰写的摘要方面表现较差。 另一个观察结果是，给ChatGPT提供的信息越多，输出被检测为“人类撰写”的可能性就越大。 这也通过人类撰写和GPT生成文本嵌入的可视化进行了验证（Liu等，2023b）。

与黑盒方法相反，白盒方法需要额外访问每个标记的模型概率。 因此，可用的白盒检测器较少。

这类检测器的一个例子是 *HowkGPT*（Vasilatos等，2023），它利用预训练的*GPT*-2模型参数来区分学生撰写的作业和*GPT*生成的作业。 主要思想是计算学生生成和Chat*GPT*生成的答案的困惑度分数，并找到将这两类分开的最佳阈值。[4]

如前所述，与白盒方法相比，黑盒方法不需要访问每个标记的模型概率。 因此，白盒检测器很少见且不太实用，因为大型语言模型不断变化，大多数模型不提供白盒访问。 独立于模型访问并可以轻松调整到新模型的黑盒方法似乎更为可行和实用。

在第6.3节中，我们将深入探讨检测人工智能生成文本的技术。

[4] 这种技术的网络应用程序可在https://howkgpt.hpc.nyu.edu/上找到。

5.4 大型语言模型生成的文本和虚假信息

大型语言模型，特别是在开放领域问答系统中实施时，可能参与虚假信息的制造和传播（Pan等，2023a；Chen等，2023；Pan等，2023b）。

直觉上，正如潘等人所提出的，对抗ODQA系统中错误信息传播的一个简单策略是减少其流行度，换句话说，减少QA系统暴露于错误信息的比例。这可以通过检索更多段落作为读者背景来实现。

然而，研究表明，扩大上下文大小在减轻由错误信息引起的性能下降方面几乎没有或没有改善（Tam等人，2022年）。 因此，通过增加上下文大小来“稀释”错误信息的基本方法对于错误信息防御是无效的。

另一种方法是指示LLMs发布关于潜在误导内容的警告通知。 例如，读者可以收到这样的指示：“谨慎行事，因为某些文本可能旨在欺骗您”。

此外，可以根据各种特征（如内容、风格或传播结构）识别和过滤大型语言模型生成的错误信息。例如，陈等人（潘等人，2023b）提出了四种针对指令的策略，以增强大型语言模型对错误信息的检测能力。 这些策略包括指令过滤，涉及过滤出不遵循指令或包含误导信息的大型语言模型的输出，指令验证，验证大型语言模型的输出是否符合指令或外部来源，以检查其有效性和可靠性，以及指令组合，将多个指令结合起来，从大型语言模型中生成更多样化和准确的输出。

陈等人（陈等人，2023）提出的另一种有趣的方法是读者集成。 这种技术利用多个语言模型的集体力量来审查和验证给定大型语言模型产生的信息。通过交叉检查输出，集成旨在增强响应的可靠性和一致性。

此外，陈等人引入了警惕提示，这涉及精心制作LLMs的提示或指令。目标是双重的：防止生成错误信息并保持机器的身份隐秘。

虽然这些开创性的方法无疑增强了更可靠和可信赖的LLMs的追求，但人工智能生成的文本与人类撰写的内容的融合需要更有效的手段来识别和管理人工智能产生的误导性信息。在我们之前的讨论中，我们提到了白盒和黑盒检测技术。 在第6.3节中，我们将深入探讨这些方法，提供额外细节。

6 风险缓解策略

在前一节中，我们探讨了与LLMs相关的各种风险类别。现在，在这一节中，我们将调查缓解这些风险的策略。 本节的摘要如图2所示。

6.1 编辑大型语言模型

大型语言模型已经成为各个领域广泛采用的方法。 然而，当某些大型语言模型拥有数十亿个参数时，一个关键问题出现了：我们如何解决不良行为，比如生成冒犯性内容或产生错误答案，而不需要对大型语言模型进行完全重新训练呢？

解决这个问题的关键在于理解大型语言模型参数中存储信息的具体位置。 在编辑大型语言模型时，这变得至关重要，因为它指导了在哪里进行修改的决策，特别是在处理幻觉时。 根据经验，事实信息往往存储在大型语言模型的中间层中。(Meng等人，2022a,b)。 相比之下，常识知识，如Gupta等人的研究所示(Gupta等人，2023)，通常存储在早期层中。

模型编辑网络与梯度分解（MEND）（Mitchell等，2021）和具有检索增强对照模型的半参数编辑（SERAC）（Mitchell等，2022）是模型编辑方法的示例，其中对预训练的LLM进行编辑以实现更好的期望行为。

MEND涉及训练一组多层感知器（MLPs）来修改梯度，使得局部参数编辑不会对其产生不利影响

模型在无关输入上的表现。 MEND分为两个阶段：训练和随后的编辑过程。 该方法应用于T5、GPT、BERT和BART模型，并在包括zsRE问答、FEVER事实检查和Wikitext生成在内的数据集上进行评估。MEND有效地编辑了最大的可用transformers，在修改程度方面优于其他方法。

SERAC，一种基于内存的模型编辑算法，利用外部内存来增强模型行为。 这种策略涉及外部编辑内存、分类器和反事实模型。 编辑存储在内存组件中，然后使用反事实模型进行分类和评估。 如果被认为相关，这些编辑将被纳入模型进行更新。

该方法使用T5-large、BERT和BB-90M模型在诸如问答（QA）、具有挑战性的问答（QA-hard）、事实核查（FC）和对话情感（ConvSent）等数据集上进行评估，并显示出非常成功。

Meng等人提出的另一个名为Rank-One Model Editing（ROME）的框架，涉及修改前馈权重以评估事实关联回忆。 他们的方法检查网络内的神经元激活，并调整权重以识别与事实信息相关的变化。 此外，他们策划了一个反事实断言数据集（COUNTERFACT）来评估语言模型中的反事实编辑。 通过因果追踪，他们确定了保留事实信息最关键的多层感知（MLP）模块。 他们的研究突显了MLP模块中间层对于回忆事实细节的重要性。

一种名为Transformer中的Mass-Editing Memory（MEMIT）的方法（Meng等人，2022b）专注于使用额外的‘记忆’（关联）更新LLMs，可以扩展到大规模。
MEMIT的目标是修改存储在LLMs权重中的事实关联。MEMIT受ROME启发，ROME在单一基础上编辑LLM，而MEMIT能够扩展到GPT-J和GPT-NeoX的数千个关联（记忆）。此外，他们能够对多个层之间的参数进行更新。

古普塔等人（古普塔等人，2023年）扩展了MEMIT框架以适应处理常识知识。而孟等人专注于编辑语言模型，以评估它们是否存储与百科知识相关的关联，而这项工作专门针对与百科知识不同的常识知识。 百科知识围绕主谓关系，而常识知识涉及概念和主谓对。 他们的方法，称为MEMIT$_{CSK}$，有效地纠正了常识错误，并可应用于编辑主语、宾语和动词。

通过实验，他们证明常识知识往往更普遍地存在于语言模型的早期层，与百科知识相反，后者通常在中间层中发现。

王等人（Wang等人，2023年）提出了一个模型编辑框架，该框架融合了多种模型编辑技术，确保在各种大型语言模型中易于使用。该框架抽象出一个编辑器。 该编辑器应用模型编辑技术来评估需要在大型语言模型中修改的特定超参数，例如某些层或神经元。然后使用可定制的评估指标来评估模型编辑方法的性能。 他们展示了该框架在几种大型语言模型上的有效性，包括T5、GPT-J、GPT-NEO、GPT-2、LLaMA和LLaMA-2。利用ROME、MEMIT、MEND等方法。

然而，姚等人（Yao等人，2023年）进行了一项分析，评估模型编辑方法的性能。 他们引入了一个专门设计用于此目的的新数据集。 他们的主要重点是两种LLM编辑方法：一种旨在使用辅助模型保留LLM的参数，另一种涉及直接修改LLM的参数。 为了评估性能，他们利用了两个数据集，包括由GPT-4生成的新构建的数据集，其中包括相关问题和答案。 他们的研究结果突出了LLM在可移植性、局部性和效率方面仍然需要改进的持续需求。

关于编辑LLM的研究已经证明了模型编辑的重要性，并确定了特定的知识领域。 最近的贡献，如王等人的工作（王等人，2023年），引入了用户友好的框架用于LLM编辑，增强了其影响。然而，

尽管取得了这些进展，LLM仍然需要不断改进，特别是在可移植性、局部性和效率等方面。

6.2 红队/绿队

传统上，红队指的是系统性的对抗性攻击，用于测试安全漏洞。 随着大型语言模型的兴起，这个术语已经超越了传统的网络安全。 现在它包括各种形式的探测、测试和攻击人工智能系统。 大型语言模型可以产生良性和有害的输出。 针对大型语言模型的红队工作，侧重于识别潜在的有害内容，如仇恨言论、煽动暴力或色情材料（Ganguli等，2022年；Ge等，

2023).

例如，一个大型语言模型可能会被给予一个导致不良输出的提示。 这样的结果可能会被利用来对发出提示的人或其他人造成影响。 因此，至关重要的是采用红队工作来揭示在大型语言模型测试过程中可能被忽视的任何意外后果。

许多研究已经探讨了在LLMs环境中的红队行动（Ge等，2023年；Perez等，2022b年；Bhardwaj和Poria，2023年），揭示了它们的优势和劣势。 鉴于它们的显著有效性，红队行动在理解LLMs潜在不利影响方面发挥着关键作用。

例如，Zhuo等人（Zhuo等，2023年）通过采用提示注入技术调查了ChatGPT是否产生危险输出，而其他研究（Shi等，2023b年；Casper等，2023年；Perez等，2022a年）则专注于特定的红队行动方面，包括开发毒性分类器或使用红队行动LLMs来识别可能被忽视的风险生成。

Ganguli等人的一项工作（Ganguli等，2022年）展示了各种抽样技术如何能够抑制特定的红队行动元素。
例如，RLHF显示出比拒绝抽样更具弹性。 然而，这些方法通常涉及大量人力参与。

为了解决这种手动负担，其他研究人员已经寻找自动化红队的方法。例如，李等人（李等人，2023a）利用贝叶斯优化来进行红队，减少查询次数并减少对人力帮助的依赖。

有趣的是，关于一个名为“绿色团队”的概念正在兴起的研究（斯特普尔顿等人，2023年）。 与专注于识别漏洞和风险的红队不同，绿队探讨潜在不安全内容仍可能具有益处的情景。 它承认灰色地带——LLMs生成可能被视为不安全但具有一定目的的内容的情况。 例如，使用LLMs生成有意义的错误代码以供教育目的属于这一类别。

在我们探索LLMs行为的复杂性时，红队和绿队都有助于更全面地了解它们的能力和局限性。

对LLMs进行红队行动揭示了在诱使这些模型产生不安全内容方面易与难之间微妙的平衡。 确保生成有害输出仍然具有挑战性，需要持续的努力和新颖的方法。

6.3 检测AI生成的文本随着AI生成内容越来越像人类撰写的文本，区分二者已成为一项日益艰巨的任务。

检测LLM生成的文本在人类撰写的内容中就像一把双刃剑。
一方面，识别差异可以提高AI生成内容的质量；另一方面，它也使识别过程变得更加复杂。

近年来，学者们引入了一系列方法来识别AI生成的文本（Pegoraro等，2023年；He等，2023年；Tang等，2023年b）。 正如前一节简要讨论的，我们可以将这些技术分类为两大类：黑盒和白盒技术。 在黑盒设置中，我们只能访问由LLM生成的输出文本，而在白盒设置中，还可以访问每个标记的模型输出概率。

在本节中，我们将讨论一些检测技术以及它们的漏洞和限制。 最后，我们将从理论角度讨论检测的可能性。

6.3.1 将语言模型微调为监督检测器，这是白盒和黑盒检测器常用的检测方法之一。

是在AI和人类生成的文本集上对语言模型进行微调（Solaiman等，2019年；Bakhtin等，2019年；Antoun等，2023年；Zhan等，2023年；Li等，2023b年）。

然而，大多数大型语言模型需要昂贵的计算资源，几乎不可能生成涵盖广泛样本的足够大的数据集，因此这种策略并不总是最佳选择。

此外，这种方法容易受到对抗性攻击的影响，比如数据中毒。 例如，黑客可以通过获取训练过程中使用的人类参考文本和检测器排名来规避检测。 更令人担忧的是，攻击者可以在白盒环境中破坏检测器的训练。 这种漏洞的产生是因为许多检测器是在常用数据集上训练的，使它们极易受到甚至最简单的攻击的影响（Krishna等，2023年；Sadasivan等，2023年）。

另一个缺点在于它们对改写攻击的敏感性。 这些攻击涉及在生成文本模型之上添加一个释义，这可以欺骗任何形式的检测器，包括那些利用监督神经网络的检测器。

6.3.2 预训练语言模型作为零样本检测器

另一种研究途径涉及利用预训练模型在零样本设置中辨别由人工智能编写的文本，而无需额外的训练或数据收集（Su等，2023年；Zer，2023年；Wang等，2023b年；Gehrmann等，2019).

根据（Mitchell等，2023年）的说法，这些技术通常设定一个预测的每个标记对数概率的阈值来识别由人工智能生成的文本。 这种方法依赖于观察到的由人工智能生成的段落通常表现出负对数概率曲率。

虽然这种方法减轻了数据毒化攻击的风险并最小化了数据和资源开销，但仍然容易受到其他对抗性攻击的影响，如欺骗和改写（Krishna等，2023年；Sadasivan等，2023年）。

6.3.3 基于信息检索技术的检测

在信息检索技术领域中，我们遇到了专门设计用于区分人类编写和AI生成文本的方法。

这些技术通过将给定文本与由LLMs生成的文本数据库进行比较来运作。目标是识别语义上相似的匹配项，从而帮助区分过程。

通过利用这些方法，研究人员旨在增强文本检测机制的稳健性和可靠性。 无论是匹配关键字，遍历超文本链接，还是使用更复杂的算法，目标始终保持一致：识别区分人类创作内容与其AI生成对应物之间微妙差异的能力（Krishna等，2023年；Sadasivan等，2023年）。

然而，这些方法并不适用于现实世界的应用，因为它们需要一个庞大且更新的AI生成文本数据库，这可能在计算上昂贵，甚至可能在所有领域、任务或模型中不存在或无法访问。 此外，像许多其他检测方法一样，它们容易受到改写和欺骗攻击的影响。(Krishna等，2023年；Sadasivan等，2023年；Wolff，2020年；Liang等，2023年)。

6.3.4 数字水印作为检测的签名

另一种被称为数字水印技术的研究方向使用模型签名在生成的文本输出中盖上特定的模式。

例如，Kirchenbauer等人(Kirchenbauer等人，2023年)建议使用软水印技术，将标记分为绿色和红色列表，以帮助创建这些模式。 一个带有数字水印的大型语言模型从其前缀标记给出的绿色列表中，以高概率采样一个标记。 这些水印通常对人类是不可见的。

为了更好地理解Kirchenbauer等人提出的技术，假设一个自回归语言模型是在一个大小为$|V|$的词汇V上进行训练的。 在步骤t处输入一系列标记作为输入，语言模型通过输出一个包含每个词汇表中项目的$l$ogit分数向量$l_t \in \mathbb{R}^{|V|}$来预测序列中的下一个标记。 随机数生成器使用h个先前标记的上下文窗口作为种子，基于伪随机函数（PRF）$f: \mathbb{N}^h \to \mathbb{N}$。使用这个随机种子，大小为$\gamma|V|$的标记子集，其中$\gamma \in (0,1)$是绿色列表大小，被“着色为绿色”并标记为$G_t$。 现在，$l$ogit分数$l_t$被修改，以便具有硬度参数-

ter $\sigma > 0$:

$$l_{tk} = \begin{cases} l_{tk} + \sigma, & \text{如果 k} \in \text{G t} \\ l_{tk}, & \text{否则} \end{cases} \quad (1)$$

在最简单的情况下，将分数通过softmax层，并从输出分布中进行抽样，导致对来自G_t的令牌有偏好。 生成水印文本后，可以通过重新计算每个位置的绿名单并找到绿名单令牌位置的集合来检查水印，而无需访问LLM。 长度为 T的令牌序列的统计显著性可以通过推导z分数来建立：

$$z = \frac{(|S| - \gamma T)}{\sqrt{\gamma(1-\gamma)T}} \quad (2)$$

当这个z分数很大且相应的P值很小时，可以确信文本被加了水印（Kirchenbauer等人，2023a）。

然而，直到所有高度成功的LLM都得到类似的保护，水印技术才不能成为防止LLM滥用的有效策略。 此外，水印技术不幸地在现实世界中的应用受到限制，特别是当只有黑盒语言模型可用时。 由于API提供者出于商业原因选择不公开概率分布，大多数基于API的应用程序开发者发现自己无法独立为文本添加水印。

尽管如此，为了为第三方提供自主水印注入，杨等人为黑盒语言模型使用场景开发了一个水印框架（杨等，2023b）。

他们引入了一个二进制编码函数，生成与单词对应的随机二进制编码。 在没有水印的情况下，编码遵循伯努利分布，其中表示位1的单词的概率约为0.5。 为了嵌入水印，他们通过有选择地替换与位0相关联的单词，使用表示位1的基于上下文的同义词来修改分布。 随后，采用统计测试来检测水印。 值得注意的是，即使遭受句子反向翻译、句子精炼、词语删除和同义词替换等攻击，删除水印而不影响原始含义仍然是潜在攻击者面临的一项具有挑战性的任务。

Kirchenbauer等人的另一项工作(Kirchenbauer等人，2023b)研究了水印作为识别和跟踪人工智能生成文本的方法的可靠性。 他们调查了带水印文本在人类重组、非带水印大型语言模型改写和融入更长人类撰写文档中的表现。

他们发现即使经过自动化和人工改写，水印仍然可能被发现。当检测到足够的标记时，改写有统计学上的可能泄露n-gram甚至更大片段的原始文本，导致高置信度检测，尽管这些攻击削弱了水印的有效性。 他们主张将水印可靠性解释为文本长度的函数，并发现，即使有意删除水印，即使是人类作者在文本长度达到 1000字时也无法做到。 事实证明，上述解释是水印的一个重要特征。 根据这项研究，最可靠的策略是水印，因为其他范式，如检索和基于丢失的检测，并没有随着文本长度的增加显示出显著改进。

尽管先前的发现，基于水印的方法在理论上和实践上仍然容易受到改写攻击的影响。 研究表明，即使通过水印技术保护的语言模型也容易受到欺骗攻击的影响。 在这些攻击中，人类对手将他们自己的文本插入到人类生成的内容中，制造出材料起源于语言模型的假象。 对于更深入的见解，感兴趣的读者可以参考Sadasivan等人的工作（Sadasivan等人，2023年）。

此外，张等人（Zhang等人，2023a）的一项新研究表明，在合理的假设下，没有强大的水印方案可以阻止攻击者在明显降低输出质量的情况下删除水印。 我们将在第6.3.8节深入探讨这项研究的发现。

6.3.5 作为检测线索的辨别特征

另一方面的工作是基于辨别特征来识别和分类。

例如，于等人（Yu等人，2023年）已经确定了一种特定于GPT生成文本的遗传特征。根据这一点

表6：AI生成文本检测技术

论文	方法	主要思想	漏洞
（Solaiman等人，2019年；Bakhtin等人，2019年；Antoun等人，2023年；Zhan等人，2023年；Li等人，2023b年）	监督检测	对人工智能和人类生成的文本集进行微调模型。	在常用数据集上训练，使其容易受到大多数攻击，包括释义攻击。
（苏等，2023年；泽尔，2023年；王等，2023年b；（Gehrmann等人，2019年）	零样本检测	在零样本设置中使用预训练的语言模型。	降低数据毒化攻击的风险，消除数据和资源开销，但仍容易受到其他对抗性攻击，如欺骗和释义。
（克里希纳等，2023年；萨达斯瓦克等，2023年）	基于检索的检测	应用信息检索方法，将给定文本与由LLMs生成的文本集匹配，并找到意义上的相似性。	这是不切实际的，因为它需要一个庞大且更新的文本集合，这在计算上是昂贵的，或者可能在所有领域、任务或模型中都不可用。 它也容易受到改写和欺骗攻击的威胁。
（Kirchenbauer等，2023a; Yang等，2023b; Kirchenbauer等，2023b; Sadasivan等，2023）	水印	在生成的文本输出中使用模型签名来标记特定模式。	最可靠的策略，但对生成模型来说基本上是不可能的。 容易受到重新表述和欺骗等攻击的影响。
（Yu等，2023; Yang等，2023c; Mitchell等，2023; Su等，2023）	基于特征的检测	根据提取的区分特征进行识别和分类。	容易受到改写等对抗性攻击的影响。

特征，模型的输出基本上重新排列了其训练语料库中存在的内容。 简单来说，当模型重复回答问题时，其回答受其训练数据中的信息限制，导致变化有限。 这个假设表明语言模型（如ChatGPT）的输出是可预测的，这意味着对于非常相似的问题，模型将产生相应相似的答案。 类比地，亲子鉴定涉及使用DNA档案来确定一个个体是否是另一个个体的生物父母。当涉及到父母权利和责任，并且对孩子的亲子关系存在不确定性时，这个过程变得特别关键。

在另一项研究中（杨等，2023c），杨等介绍了一种名为Divergent N-Gram Analysis（DNA-GPT）的检测方法。 这种方法无需训练即可运行，并且通过n-gram分析在黑盒设置中或概率分歧在白盒环境中评估给定文本及其截断段之间的差异。

对于黑盒场景，杨等人定义DNA-GPT BScore如下：

$$BScore(S,\Omega) = \frac{1}{K}\Sigma_{k=1}^{K}\Sigma_{n=n_0}^{N} f(n)\frac{|\text{n-grams}(\hat{S}_k) \cap \text{n-grams}(S_2)|}{|\hat{S}_k||\text{n-grams}(S_2)|} \quad (3)$$

其中，S是LLM的输出，S_2是人类编写的标准答案，$f(n)$是不同n-grams的权重函数，$\Omega = \{\hat{S}1, \ldots, \hat{S}K\}$。

对于白盒场景，他们提出计算Ω和S之间的DNA-GPT WScore：

$$W\ Score(S,\Omega) = \frac{1}{K}\Sigma_{k=1}^{K} log\frac{p(S_2|S_1)}{p(\hat{S}_k|S_1)} \quad (4)$$

其中 Ω是一个大型语言模型解码器的样本集合，$\hat{S}$ =LM($S1$)而$S2$是人类编写的真实文本。 在黑盒和白盒场景中，检测准确性的关键参数有两个：截断比例和重新提示迭代次数 K。 这种策略展示了人工智能生成的文本与人类编写的文本之间的显著差异。

另一个区分特征是文本对操纵的敏感性。人工智能生成的文本和人类编写的文本都会受到小的扰动的负面影响，例如替换一些词语。 然而，一些最近的研究（Mitchell等，2023年；Su等，2023年）表明人工智能生成的文本更容易受到这种操纵的影响。

例如，为了衡量大型语言模型对扰动的敏感性，苏等人提出了对数似然对数秩比（LRR）：

$$LPR = -\frac{\Sigma_{i=1}^{t}\log p_\theta(x_i|x_{<i})}{\Sigma_{i=1}^{t}\log r_\theta(x_i|x_{<i})} \quad (5)$$

其中 $r_\theta(x_i|x_{<i}) \geq 1$是前面标记$x_i$的排名（苏等人，

2023年）。分子中的对数似然表示正确标记的绝对置信度，而分母中的对数排名则考虑了相对置信度，揭示了关于文本的互补信息。他们还提出了归一化对数排名扰动（NPR）如下：

$$\text{NPR} = \frac{\frac{1}{n}\Sigma_{p=1}^{n} \log r_\theta(\tilde{x}_p)}{\log r_\theta(x)} \tag{6}$$

在目标文本 x 上应用小扰动，以产生扰动文本 $\tilde{x}_p$。

研究表明，对于人工智能生成的文本，LRR 倾向于更大，提供了一个区分因素。 一个合理的解释是，在人工智能生成的文本中，对数秩比对数似然更加显著，使得 LRR 成为这类文本的一个有用指标。

NPR 背后的理念是，无论是人工智能生成的还是人类撰写的文本，都会受到小扰动的负面影响。 具体来说，在扰动后，对数秩分数会增加。 然而，人工智能生成的文本更容易受到扰动的影响，导致对数秩分数在扰动后增加更多。 因此，NPR 为人工智能生成的文本提供了更高的评分（苏等，2023年）。 由于本研究仅涵盖了一些检测技术，需要进行更广泛和系统的评估来验证LLM的能力。

尽管我们在这里讨论了所有方法，科学家们已经揭示出通过有效优化提示，LLM可以规避许多检测技术。

例如，卢等人（Lu等人，2023年）提出了一种新颖的基于替换的上下文示例优化方法（SICO），可以自动生成这样的提示。 为此，SICO首先从一组人类和AI生成的文本中提取出区分特征。 然后，这些特征和一个释义提示被连接到AI生成的任务中，并馈送给LLM，以修改AI生成的文本。 提示通过单词和句子级别的替换进行优化，以最大程度地减少检测的概率，并最大程度地增加AI生成文本与人类写作文本的相似性。结果明确证明了现有检测器的脆弱性。

6.3.6 检测技术的泛化能力

机器生成文本检测器在未见过的数据上的泛化能力，跨越多个维度，如多领域、多语言和各种生成模型，是另一个需要考虑的重要方面。

有一些研究，比如王等人的研究（Wang et al., 2023h），通过在跨越多个生成器、领域和语言的大规模语料库上进行实验，来调查检测器的泛化能力。 他们的调查涉及利用各种生成模型，包括ChatGPT、textdavinci-003、LLaMa、FlanT5、Co-here、Dolly-v2和BLOOMz，来创建文本文章。 随后，他们尝试使用传统机器学习方法（例如线性支持向量机）和现代基于变换器的模型（例如RoBERTa）来区分人工智能生成和人类撰写的内容，重点关注风格特征。

有趣的是，他们的研究结果显示，虽然这些文本检测方法在特定领域内表现良好，但在跨领域检测任务中遇到挑战。

此外，他们发现所有检测模型在检测展现出特定模式的内容时表现更好，这使其与人类撰写的内容（在这种情况下是ChatGPT）有所区别。

此外，他们表明在跨生成器设置中——即检测器在一个大型语言模型生成的文本上训练，但在另一个大型语言模型生成的数据上测试时——大多数模型都会遭受性能下降和缺乏泛化能力的问题。

6.3.7 检测技术的漏洞 正如前面提到的，零日攻击容易受到数据毒化等对抗技术的影响。研究人员采用监督方法来对抗这些攻击，但大多数检测策略仍然容易受到释义或欺骗的影响。

为了解决这一挑战，基于检索的检测器充当一种防御机制。 这些探测器将LLM的输出存储在数据库中，并执行语义搜索以识别最佳匹配，如前所讨论的。 这种方法提高了探测器抵抗改写攻击的能力。 然而，重要的是考虑与存储用户-LLM对话相关的隐私问题。 此外，这种技术证明

在处理递归改写时是无效的（Sadasivan等，2023年）。

此外，研究人员发现，通过精心优化提示，LLM可以有效地规避各种检测技术。 例如，提示可以通过单词和句子替换精心制作，旨在最大程度地减少检测的机会，同时最大程度地增加人类和AI生成文本之间的相似性（Lu等，2023年）。

虽然水印技术被认为是一种有效的检测策略，但它面临着几个挑战。 首先，除非所有大型语言模型都得到统一保护，否则水印技术仍然无效。
其次，其实际适用性有限，特别是在处理黑盒语言模型时。 第三，API提供者经常隐瞒概率分布，阻止第三方开发者独立为文本加水印。

最后，最近的研究表明，没有强大的水印方案可以防止攻击者在明显降低输出质量的情况下删除水印。

因此，为生成模型加水印可能从根本上是不可实现的，需要采取其他方法来保护模型开发者和大型语言模型用户的知识产权。

表6展示了检测策略的概览，突出了与每个类别相关的漏洞。

6.3.8 关于检测可能性的讨论

鉴于对由大型语言模型生成的文本检测越来越感兴趣，研究人员最近从理论角度探讨了检测AI生成文本的可能性，探索了与这一任务相关的基本可行性和边界。

例如，Sadasivan等人提出了一个不可能性发现（Sadasivan等人，2023年）： “随着语言模型变得更加复杂并且更擅长模拟人类文本，即使是最好的检测器的性能也会急剧下降”。 Sadasivan等人提出了任何解码器$\mathcal{D}$的ROC曲线下面积的上限为：

$$AU\ ROC(\mathcal{D}) \leq \frac{1}{2} + TV(\mathcal{M}, \mathcal{H}) - \frac{TV(\mathcal{M}, \mathcal{H})^2}{2} \tag{7}$$

其中$TV(\mathcal{M}, \mathcal{H})$是机器生成文本和人类生成文本之间的总变差距离。这公式表明，当人类生成文本和机器生成文本非常相似，即$TV(\mathcal{M}, \mathcal{H})$非常小时，即使是最好的检测器也只能比随机分类器表现略好一点。

然而，查克拉博提等人（查克拉博提等人，2023年）的另一项有趣研究表明：“只要人类生成的文本和机器生成的文本的分布不完全相同，在大多数情况下都是如此，如果我们收集足够的样本，就有可能检测到人工智能生成的文本”.

事实上，查克拉博提等人证明，Sadasivan等人提出的AUROC上限在实际情况下可能过于保守，无法有效检测。 具体来说，他们通过用$TV(\mathcal{M}, \mathcal{H})$替换$TV(\mathcal{M})$引入了一个隐藏的可能性

$\otimes^n, \mathcal{H}^{\otimes n})$ 在AUROC方程中，其中 $m^{\otimes^n} := m \bigotimes m \bigotimes \cdots \bigotimes m$ (共n次)表示样本集合S上的乘积分布 $\mathcal{S} := \{s_i\}, i \in \{1, \ldots n\}$, 同样也是$\otimes^n$. 自从 $TV(\mathcal{M}^{\otimes^n}, \mathcal{H}^{\otimes^n})$ 作为一个递增序列，随着每个分布样本数量的增加，它最终收敛到 1。 很明显，如果样本数量增加，总变差距离会非常快地接近 1，从而增加AUROC。在另一项研究中，张等人（张等人，2023a）探讨了水印检测的理论方面。 他

们将水印定义为将统计信号（通常称为“水印”）嵌入模型输出的过程。 这个嵌入的水印充当验证信号，确保输出确实来自模型。 强大的水印方法可以防止攻击者在明显降低输出质量的情况下移除水印。

在这项研究中，作者提出了两个基本假设。 首先，他们引入了“质量预言机”的概念，允许攻击者访问一个能够评估模型输出质量的预言机。 这个预言者帮助攻击者评估修改后响应的质量。 其次，它们引入了“扰动预测器”，允许攻击者修改输出同时保持一定概率的质量。 实质上，扰动预测器在高质量输出上诱导了一个有效混合的随机游走。

他们发现，对于任何公开或秘钥

水印方案，满足这些假设，存在一个高效的攻击者："给定一个提示 p和一个带水印的输出 y，这个攻击者可以利用质量和扰动预测器获得一个输出 y'，其概率非常接近 1。 攻击者的目标是找到一个输出 y'，使得(1) y'没有高概率水印且(2) $Q(p, y') \geq Q(p, y)$" (Zhang et al., 2023a)。 简单地说，水印技术在不造成显著质量降低的情况下是不可能的，因此，应该利用替代方法来保护模型开发者的知识产权。

检测人工智能生成的文本是一项至关重要且具有重大影响的挑战性任务，对相关的自然语言处理任务有着重要影响。 然而，当前的最先进方法有时会受到对这一任务的基本可行性和边界缺乏全面理解的限制。

因此，进一步探索和研究人工智能文本检测的理论方面至关重要，因为这可以促进更健壮和有效的技术的发展，同时也可以识别新的研究方向和机会。

7 新机遇和未来研究

本文全面概述了大型语言模型安全和风险缓解领域的最新发展和最佳实践。为了拓宽视野，本节探讨了推进大型语言模型安全、漏洞和风险缓解研究领域的新机遇。

7.1 安全与隐私研究中的机遇

数据泄露和记忆方面的机遇 正如之前提到的，大型语言模型面临与记忆和数据泄露相关的挑战。
探索新的机会来解决这些问题可以显著推动该领域。 一些有前途的途径包括：

- **开发多方面技术以防止敏感数据泄露：** 这些技术应考虑各种维度，包括与模型相关的方面（如训练数据选择和差分隐私）、数据相关的方面（如数据分类、访问控制和监控）以及基于用户的因素（如检测异常模式用户-LLM互动和管理用户访问)

- **开发新方法来减轻记忆化:** 鉴于目前在LLM中处理记忆化的技术稀缺，提出新的方法——如模型编辑——来减轻记忆化是至关重要的。

- **调查和识别记忆模式：** 探索是否存在记忆模式，并确定模型记忆最多的特定数据类别代表了一个未被充分探讨的机会。

LLM代码生成中的机遇 如前所述，代码生成的安全影响提出了几个挑战。 然而，每个挑战也代表着改进LLM生成代码安全性的机会。其中一些机会包括：

- **开发确保代码生成的可重现性和透明性的方法：** 这可以涉及实践，如记录使用的种子，指定模型版本，并在代码生成过程中捕获相关提示细节。

- **探索生成大量和多样化的代码样本库进行安全分析的方法：** 研究人员可以采用数据增强等技术，其中包括创建现有代码片段的变体，强调测试模型鲁棒性的对抗性示例，甚至自我对弈技术，其中模型生成代码样本并根据自己的预测进行评估（Wang等，2023c）。

- **设计真实和全面的测试场景，涵盖软件安全的各个方面：** 这些场景应包括功能需求，确保软件表现如预期，以及非功能性需求，如性能、可扩展性和可靠性。 此外，纳入对抗性需求——系统针对故意攻击或误用进行测试——可以进一步增强安全评估过程。

- **提高代码生成的健壮性和泛化能力以适应不同提示：**研究人员有机会提高代码生成在不同提示下的健壮性和适应性。 最终目标是开发一个模型，能够持续生成安全的代码，适用于所有输入提示。 实现这一目标确保生成的代码始终可靠、有弹性，并且抵抗漏洞，无论使用何种提示。

- 研究编程语言对代码生成安全性的影响：通过分析各种语言和范式的优势和劣势，我们可以确定哪些语言对于代码生成更为健壮。 这种探索有利于代码生成实践以及编程语言的设计和增强。

- **适应网络安全的进化性质：**网络安全的进化性质需要积极的措施。

  这些措施包括保持训练数据的最新性，修订评估指标，并将代码生成实践与最新的行业标准和趋势保持一致。

7.2漏洞研究中的机遇 在理解LLM漏洞的基础工作基础上，仍然存在着深化我们知识和增强这些模型韧性的广泛机会。 以下领域代表了未来研究的有希望的途径：

- 在分类任务之外扩展对不同**NLP**应用的实验：虽然分类任务一直是主要关注点，但探索其他**NLP**应用——如语言生成、摘要、情感分析和问题回答——将提供对**LL**M行为和安全影响更全面的理解。

- **在模型架构和模型规模两个层面上审查大型语言模型的脆弱性：**全面评估风险并根据确定的漏洞的严重程度优先考虑减轻措施是至关重要的。 理解架构选择和模型规模之间的相互作用

可以帮助我们做出更好的决策，以增强大型语言模型的安全性。

- **为防范后门注入而调整多方面的防御策略：**研究人员有机会通过结合技术，如提示过滤机制来排除有害输入，以及专门针对安全的大型语言模型来检测可疑指令，制定多方面的防御策略来防范后门注入。

- **扩大提示注入研究范围：**包括更复杂的交互场景，如对话代理和上下文感知应用，我们可以发现新的见解，并潜在地增强对安全威胁的抵抗力。

- **调查迁移学习和微调的作用：**这涉及研究在预训练模型中最初确定的漏洞在为特定任务或领域微调这些模型时可能会被放大或减轻。

- 识别和减轻新兴风险**:** 新研究可能结合安全、行为分析、对抗学习和网络安全取证的最新进展，以检测和减轻复杂的攻击。

- 评估和评价数据集多样性和代表性的影响：这涉及检查训练数据的特征如何影响模型抵御漏洞的能力，特别是关于偏见和公平性以及更重要的数据污染方面。

7.3 风险缓解研究中的机遇 AI 生成文本检测技术 如前所述，检测 AI 生成文本是一项关键且具有挑战性的任务，当前方法经常受到不同因素的限制，并容易受到恶意攻击的影响。

因此，对AI文本检测的理论和实践方面进行进一步的探索和调查至关重要。 一些机会包括：

- **创建更多多样化和代表性的数据集：**现有数据集可能不

涵盖了用于训练和评估AI文本检测模型的AI生成内容的所有细微差别。 开发更多样化和代表性的数据集可以提高模型的泛化能力，并实现更可靠的评估。

•**探索更高级和可解释的特征：**通过识别微妙的细微差别和可解释的特征，我们可以更细致地理解人类编写和AI生成的文本之间的区别。

•**开发更健壮和领域自适应的学习方法：**考虑到人工智能生成文本领域的不断变化，探索像对抗学习、元学习和自监督学习（Weber-Wulff等，2023）这样的方法可以产生更具弹性和适应性的解决方案。

•**对基本可行性和边界的全面理解：**对于AI生成文本检测的基本可行性和限制的彻底掌握至关重要。然而，当前最先进的方法主要忽视了这项任务的理论方面。 因此，进一步探索和研究理论方面是必要的。 这可能导致更健壮和有效的技术的发展，以及揭示新的研究方向。

•**评估AI文本检测的道德和社会影响：**虽然检测合成内容至关重要，但重要的是考虑误报的潜在风险。 这些不准确性可能导致意外后果，如不公正的处罚或无端的怀疑，影响个人和整个社会。

编辑LLMs 了解LLM中保存知识的位置很重要，因为这可能导致幻觉或偏见等不良后果。 因此，识别存储的事实信息的性质，并应用缓解策略以消除潜在的不可靠来源是至关重要的。 尽管这个领域取得了显著进展，但仍有一些需要改进的地方：

- **为测试多种方法开发统一平台：**随着新方法的出现，拥有一个统一的框架可以在不同数据集之间进行高效比较。 此外，将不同类型的知识整合到这个框架中简化了对专注于不同数据或知识领域的层的评估（Wang等，2023年）。

- **在跨多样化数据集和网络层进行模型编辑研究的进一步探索：**鉴于当前研究强调自然语言处理中的某些领域，评估额外的自然语言处理数据集以评估某些信息存储位置的趋势是否一致是有益的。 此外，将这种评估扩展到可能不同的其他自然语言处理领域可能会揭示当前趋势的潜在变化。

红队/绿队类似于网络安全领域，红队对增强安全性是有益的，LLMs的红队和绿队都揭示了LLMs对恶意用户的脆弱性。 已经有一些新兴的贡献，比如RLHF，可以防止比其他方法更多的攻击，正如（Perez等，2022a）所示。 然而，在这些方面仍然有一些改进的空间，比如：

- **创建更多的保障措施以防止攻击影响LLMs：**随着LLMs的日益流行，它们正在成为许多产品中的核心组件。

  因此，有必要实施多重保障措施，因为新的攻击不断出现。

- **评估攻击对特定模型的影响：**基于Ganguli等人的研究（Ganguli等，2022）表明，利用RLHF的LLMs对红队攻击表现出比其他模型更大的韧性。 然而，进一步的实验是必要的，以揭示任何限制。 通过了解这些限制，研究人员可以制定创新策略来减轻现有的红队攻击，并预测潜在的新攻击。

- **设计一个自动化系统，减少红队/绿队中对人类的依赖性：**

由于人类对与红队和绿队相关的特定LLM输出的检查可能会对健康产生负面影响，因此自动化过程变得至关重要。这种自动化旨在最大程度地减少参与红队和绿队的个人所经历的伤害。

8 结论

本文对LLM的安全性和风险缓解方面进行了全面分析。我们研究了LLM使用中出现的安全问题，如信息泄露、未经授权访问和不安全的代码生成。

此外，我们探讨了针对LLM的各种攻击类型，并将它们分类为三大类：基于模型的攻击、训练时攻击和推断时攻击。 我们还调查了LLM的风险和滥用，如偏见、歧视、错误信息、抄袭、侵犯版权以及在不同领域应用LLM可能带来的其他潜在社会和伦理影响。 此外，我们提出了一项全面评估的缓解策略，可以用来提高LLM的安全性和稳健性，例如红队和绿队、模型编辑、水印和AI生成文本检测技术，同时讨论每种策略的局限性和权衡。 最后，我们确定了一些在这一领域的开放挑战和未来方向的研究，例如开发更有效和高效的防御机制，建立LLM开发和部署的标准和指南，促进LLM利用中涉及的利益相关者之间的合作和意识。

9 致谢

作者对Sadid Hassan博士进行多次讨论并就论文内容提供宝贵反馈表示感谢。 本研究代表作者进行的独立研究，不一定代表任何组织的观点或意见。

参考文献

2023年。Openai。ai文本分类器。

2023. Zerogpt：AI文本检测器。

Uri Alon，Shaked Brody，Omer Levy和Eran Yahav。2018年。 code2seq：从代码的结构化表示生成序列。 *ArXiv*，abs/1808.01400。

匿名。2023年。 如何捕捉AI谎言者：通过问无关问题来检测黑匣子LLMs中的谎言。 提交给第十二届国际学习表示会议。正在审阅中。

Wissam Antoun，Virginie Mouilleron，Benoît Sagot和Djamé Seddah。2023年。 朝着对语言模型生成文本的稳健检测：ChatGPT是否那么容易检测？ *ArXiv*，abs/2306.05871。

Owura Asare，Meiyappan Nagappan和N. Asokan。2023年。 GitHub的Copilot在引入代码漏洞方面和人类一样糟糕吗?

Anton Bakhtin，Sam Gross，Myle Ott，Yuntian Deng，Marc'Aurelio Ranzato和Arthur D. Szlam。2019年。真实还是虚假？ 学习区分机器生成的文本和人类生成的文本。 *ArXiv*，abs/1906.03351。

Himanshu Batra，Narinder Singh Punn，Sanjay Kumar Sonbhadra和Sonali Agarwal。2021年。 基于Bert的情感分析：软件工程视角。 在国际数据库和专家系统应用会议上。

Rishabh Bhardwaj和Soujanya Poria。2023年。 使用话语链进行红队测试大型语言模型以实现安全对齐。*ArXiv*，abs/2308.09662。

Meghana Moorthy Bhat, Rui Meng, 刘烨, Yingbo Zhou, 和 Semih Yavuz。2023年。 调查大型语言模型对长篇问题回答的可靠性。 *ArXiv*, abs/2309.08210。

Stella Biderman, USVSN Sai Prashanth, 林唐若维卡, Hailey Schoelkopf, Quentin Anthony, Shivanshu Purohit, 和 Edward Raf。2023年。 大型语言模型中的新兴和可预测的记忆化。 *arXiv*预印本*arXiv:2304.11158*。

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward

Raff等人。2023b。 Pythia：用于分析大型语言模型在训练和扩展中的套件。在国际机器学习会议, 页2397-2430。PMLR。

Lewis Birch, William Hackett, Stefan Traw- icki, Neeraj Suri和Peter Garraghan。2023a。模型寄生：一种针对llms的提取攻击。 *ArXiv*, abs/2309.10544。

Lewis Birch, William Hackett, Stefan Traw- icki, Neeraj Suri和Peter Garraghan。2023b。模型寄生：一种针对llms的提取攻击。*arXiv* 预印本 *arXiv:2309.10544*。

Jaydeep Borkar。2023。 我们可以从数据泄露和遗忘中学到什么关于法律？ *arXiv*预印本 *arXiv:2307.10476*。

Marcel Bruch，Martin Monperrus和Mira Mezini。2009年。 从示例中学习以改进代码补全系统。 在第*7*届欧洲软件工程大会和*ACM SIGSOFT*软件工程基础研讨会联合会议的论文集中，*ESEC/FSE*'09，第213-222页，美国纽约。计算机协会。

蔡祥瑞，徐海东，徐思涵，张颖和袁晓杰。2022年。 Badprompt：连续提示的后门攻击。 *ArXiv*，abs/2211.14719。

曹银志和杨俊峰。2015年。 朝着通过机器遗忘实现系统遗忘。在2015年I*EEE*安全与隐私研讨会上，第463-480页。I*EEE*。

尼古拉斯·卡林尼，弗洛里安·特拉默，埃里克·华莱士，马修·贾吉尔斯基，亚里尔·赫伯特-沃斯，凯瑟琳·李，亚当·罗伯茨，汤姆·布朗，唐·宋，乌尔法尔·埃尔林松等。2021年。 从大型语言模型中提取训练数据在第*30*届*USENIX*安全研讨会（*USENIX Security 21*）上，页码*2633-2650*。

斯蒂芬·卡斯珀，贾森·林，乔·权，加特伦·卡尔普，迪伦·哈德菲尔德-门内尔。2023年。 探索、建立、利用：从零开始对语言模型进行红队测试。 ar*Xiv*预印本ar*Xiv*:2306.09442。

Souradip Chakraborty，A.S.贝迪，朱思成，邦安，迪尼什·马诺查，黄芙荣。2023. 关于人工智能生成文本检测的可能性。 *ArXiv*，abs/2304.04736。

P. V. Sai Charan，Hrushikesh Chunduri，P. Mo- han Anand和Sandeep Kumar Shukla。2023年。从文本到mitre技术：探索大型语言模型用于生成网络攻击载荷的恶意用途。 *ArXiv*，abs/2305.15336。

Honghua Chen和Nai Ding。2023年。 探究大型语言模型的"创造力"：模型能产生不同的语义关联吗？ 在计算语言学协会发现：*EMNLP* 2023中，第12881-12888页，新加坡。计算语言学协会。

Mengyang Chen，Lingwei Wei，Han Cao，Wei Zhou和Song Hu。2023年。 大型语言模型能理解内容和传播以进行误信息检测吗：一项实证研究。*ArXiv*, abs/2311.12699.

Xiaoyi Chen, Ahmed Salem, Dingfan Chen, Michael Backes, Shiqing Ma, Qingni Shen, Zhonghai Wu, 和 Yang Zhang. 2021. Badnl: 利用保留语义改进对自然语言处理模型进行后门攻击。 在年度计算机安全应用会议, 页码554-569.

Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, 和 Dawn Xiaodong Song. 2017. 利用数据毒化对深度学习系统进行有针对性的后门攻击。 *ArXiv*, abs/1712.05526.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng,Zhanghao Wu, Hao Zhang, Lianmin Zheng,Siyuan Zhuang, Yonghao Zhuang, Joseph EGonzalez, 等. 2023. 维库纳：一个开源聊天机器人，以90%*的ChatGPT质量令GPT-4印象深刻。请访问*https://vicuna.lmsys.org*（访问日期：*2023*年*4*月*14*日） *org*（访问日期*2023*年*4*月*14*日）。

Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg和Dario Amodei. 2023年. 从人类偏好中进行深度强化学习.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg和Dario Amodei. 2017年. 从人类偏好中进行深度强化学习. 在神经信息处理系统进展中

，第30卷. Curran Associates，Inc.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse和John Schulman. 2021年. 训练验证器解决数学文字问题。 *ArXiv*，abs/2110.14168。

崔世尧，张振宇，陈一龙，张文元，刘天韵，王思琦和刘廷文。2023年。 FFT：针对具有事实性、公平性、毒性的LLM的无害性评估和分析。 *ArXiv*，abs/2311.18580。

邓格雷，刘毅，李跃康，王凯龙，张颖，李泽峰，王浩宇，张天伟和刘洋。2023年。 越狱者：跨多个大型语言模型聊天机器人的自动越狱。 A*rXiv*预印本*arXiv:2307.08715*。

邓格雷，刘毅，李跃康，王凯龙，张颖，李泽峰，王浩宇，张天伟和刘洋。2023年。 主钥：跨多个大型语言模型聊天机器人的自动越狱。

A. 德希潘德，维什瓦克·穆拉哈里，坦迈·拉杰普罗希特，A. 卡良，和卡尔蒂克·纳拉西姆汉。2023a。 ChatGPT中的毒性：分析个性化语言模型。 *ArXiv*，abs/2304.05335。

A. 德希潘德，坦迈·拉杰普罗希特，卡尔蒂克纳拉西姆汉，和A. 卡良。2023b。 人工智能的拟人化：机遇和风险。 *ArXiv*，abs/2305.14784。

杜力，王业权，邢兴润，雅一群，李翔，姜欣，和方学智。2023a。 通过关联分析量化和归因大型语言模型的幻觉。 *ArXiv*, abs/2309.05217.

Yilun Du, Shuang Li, Antonio Torralba, Joshua Tenenbaum, 和 Igor Mordatch. 2023b. 通过多智能体辩论改进语言模型的真实性和推理能力。

Deep Ganguli, Liane Lovitt, John Kernion, Amanda Askell, Yuntao Bai, Saurav Kadavath, Benjamin Mann, Ethan Perez, Nicholas Schiefer, Kamal Ndousse, Andy Jones, Sam Bowman, Anna Chen, Tom Conerly, Nova Das - Sarma, Dawn Drain, Nelson Elhage, Sheer El-Showk, Stanislav Fort, Zachary Dodds, T. J. Henighan, Danny Hernandez, Tristan Hume, Josh Jacobson, Scott Johnston, Shauna Kravec, Catherine Olsson, Sam Ringer, Eli Tran-Johnson, Dario Amodei, Tom B. Brown, Nicholas Joseph, Sam McCandlish, Christo- pher Olah, Jared Kaplan, 和 Jack Clark. 2022. 通过红队测试语言模型以减少伤害：方法、扩展行为和经验教训。 *ArXiv*, abs/2209.07858.葛苏宇，周春婷，侯睿，马迪安哈布萨，王

一嘉，王琦凡，韩家伟，毛云宁。2023年。 Mart：通过多轮自动红队测试改进llm安全性。 *ArXiv*, abs/2311.07689.Samuel Gehman，Suchin Gururangan，MaartenSap，Yejin Choi和Noah A Smith。2020年。

重新毒性提示：评估语言模型中神经毒性退化。 *arXiv*预印本*arXiv:2009.11462*。

Sebastian Gehrmann，Hendrik Strobelt和Alexander M. Rush。2019年。 Gltr：生成文本的统计检测和可视化。 在计算语言学协会年会上。

Xinyang Geng，Arnav Gudibande，Hao Liu，EricWallace，Pieter Abbeel，Sergey Levine和Dawn Song。2023年。 考拉：用于学术研究的对话模型。博客文章，4月1日。

Kai Greshake，Sahar Abdelnabi，Shailesh Mishra，Christoph Endres，Thorsten Holz和Mario Fritz。2023年。 不是您注册的内容：通过间接提示注入来危害现实世界的llm集成应用程序。

Arnav Gudibande，Eric Wallace，Charlie Snell，Xinyang Geng，Hao Liu，Pieter Abbeel，Sergey Levine和Dawn Song。2023年。 模仿专有LLM的虚假承诺。 arXiv预印本 arXiv:2305.15717。Caglar Gulceh

re，Tom Le Paine，Srivatsan Srini-vasan，Ksenia Konyushkova，Lotte Weerts，Ab-hishek Sharma，Aditya Siddhant，Alexa Ah-ern，Miaosen Wang，Chenjie Gu，WolfgangMacherey，A. Doucet，Orhan Firat和Nandode Freitas。2023年。 强化自我训练（REST）用于语言建模。ArXiv，abs/2308.08998。

Anshita Gupta，Debanjan Mondal，Akshay Krishna Sheshadri，Wenlong Zhao，Xiang Lorraine Li，Sarah Wiegreffe和Niket Tandon。2023年。 在GPT中编辑常识知识。arXiv预印本 arXiv:2305.14956。

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, 和 Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre- training. ICML'20. JMLR.org.

Perttu Hämäläinen, Mikke Tavast, 和 Anton Kunnari. 2023. 在生成合成人机交互研究数据中评估大型语言模型：一个案例研究。 在2023年CHI人机交互会议论文集中, 第3580688页。ACM.

Shanshan Han, Baturalp Buyukates, Zijian Hu, Han Jin, Weizhao Jin, Lichao Sun, Xiaoya Wang, Chulin Xie, Kai Zhang, Qifan Zhang, Yuhui Zhang, Chaoyang He, 和 Salman Aves- timehr. 2023. FedML安全性：联邦学习和LLMs中的攻击和防御基准。 *ArXiv*，abs/2306.04959。

Valentin Hartmann，Anshuman Suri，Vincent Bindschaedler，David Evans，Shruti Tople和Robert West。2023a年。 Sok：通用大型语言模型中的记忆化。 *ArXiv*，abs/2310.18362。

Valentin Hartmann，Anshuman Suri，Vincent Bindschaedler，David Evans，Shruti Tople和Robert West。2023b年。 Sok：通用大型语言模型中的记忆化。 *arXiv*预印本 *arXiv:2310.18362*。

Xinlei He，Xinyu Shen，Zeyuan Johnson Chen，Michael Backes和Yang Zhang。2023年。 MGT-Bench：机器生成文本检测基准。 *ArXiv*，abs/2303.14822。

Zhengmian Hu, Gang Wu, Saayan Mitra, Ruiyi Zhang, Tong Sun, Heng Huang, 和 Vishy Swaminathan。2023年。 基于困惑度度量和上下文信息的令牌级对抗提示检测。 *ArXiv*，abs/2311.11509。

Hai Huang, Zhengyu Zhao, Michael Backes, 云沈，和杨张。2023年。 针对大型语言模型的复合后门攻击。 *ArXiv*，abs/2310.07676。

Jie Huang, Hanyin Shao, 和 Kevin Chen-ChuanChang。2022年。 大型预训练语言模型是否泄露了您的个人信息？ 在计算语言学协会发现：*EMNLP 2022*，阿布扎比，阿拉伯联合酋长国。计算语言学协会。

Xiaowei Huang, Wenjie Ruan, Wei Huang, Gao Jin, Yizhen Dong, Changshun Wu, Saddek Bensalem, Ronghui Mu, Yi Qi, Xingyu Zhao, Kaiwen Cai, Yanghao Zhang, Sihao Wu, Peipei Xu, Dengyu Wu, André Freitas, 和 Mustafa A. Mustafa。2023b。 通过验证和验证的视角对大型语言模型的安全性和可信度进行调查。ArXiv，abs/2305.11391。

Neel Jain, Avi Schwarzschild, Yuxin Wen, Gowthami Somepalli, John Kirchenbauer, Ping yeh Chiang, Micah Goldblum, Aniruddha Saha, Jonas Geiping, 和 Tom Goldstein。2023年。 针对对齐语言模型的对抗攻击的基线防御。 ArXiv，abs/2309.00614。

凯文·杰西，图菲克·艾哈迈德，普雷姆·德瓦布和艾米莉·摩根。2023年。 大型语言模型和简单、愚蠢的漏洞。ArXiv，abs/2303.11455。

焦方凯，滕志洋，沙菲克·Joty，丁博晟，孙爱欣，刘正元和陈南希。2023年。 Logicllm：探索自监督逻辑增强训练对大型语言模型的影响。ArXiv，abs/2305.13718。

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, 和 Jared Kaplan。2022年。 语言模型（大多数情况下）知道自己知道的东西。

Daniel Kang，Xuechen Li，Ion Stoica，Carlos Guestrin，Matei A. Zaharia和Tatsunori

Hashimoto。2023年。 利用llms的程序行为：通过标准安全攻击实现双重用途。 *ArXiv*，abs/2302.05733。

Mohammad Khalil和Erkan Er。2023年。 聊天-gpt会让你露馅吗？ 重新思考抄袭检测。

Aisha Khatun和Daniel Brown。2023年。 可靠性检查：对gpt-3对敏感话题和提示措辞的响应进行分析。ArXiv，abs/2306.06199。

Kiana Kheiri和Hamid Karimi。2023年。 情感gpt：利用gpt进行高级情感分析及其与当前机器学习的脱节。 *ArXiv*，abs/2307.10234。

Raphaël Khoury, Anderson R. Avila, Jacob Brunelle, 和 Baba Mamadou Camara. 2023. ChatGPT生成的代码有多安全？ *ArXiv*, abs/2304.09655.

Siwon Kim, Sangdoo Yun, Hwaran Lee, Mar-tin Gubri, Sungroh Yoon, 和 Seong JoonOh. 2023. Propile：探究大型语言模型中的隐私泄漏。 *arXiv*预印本*arXiv:2307.01881*.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, 和 Tom Goldstein. 2023a. 大型语言模型的水印。

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Manli Shu, Khalid Saifullah, Kezhi Kong, Ka-sun Fernando, Aniruddha Saha, Micah Gold- blum, 和 Tom Goldstein. 2023b. 关于大型语言模型水印可靠性的讨论。 *ArXiv*, abs/2306.04634。

詹姆斯·柯克，罗伯特·雷伊，彼得·林德斯。2023年。 通过代理分析改进从llms中提取知识以进行任务学习。

Kalpesh Krishna，Yixiao Song，Marzena Karpin- ska，John Wieting和Mohit Iyyer。2023年。 改写规避ai生成文本检测器，但检索是一种有效的防御手段。 *ArXiv*, abs/2303.13408。

迈克尔·库奇尼克，弗吉尼亚·史密斯和乔治·阿姆弗罗西亚迪斯。2023年。 用relm验证大型语言模型。 机器学习和系统会议论文集, 5。

菲利普·拉班，沃伊切赫·克里辛斯基，迪维安什·阿加瓦尔，亚历山大·法布里，蔡明雄，沙菲克·乔蒂和吴建生。2023年。 LLMs作为事实推理者：现有基准和更多见解。 *ArXiv*, abs/2305.14540.

Deokjae Lee, JunYeong Lee, Jung-Woo Ha, Jin-Hwa Kim, Sang-Woo Lee, Hwaran Lee和Hyun Oh Song。2023a。 通过贝叶斯优化进行查询高效的黑盒红队行动。 *arXiv*预印本*arXiv:2305.17444*。

Taehyun Lee, Seokhee Hong, Jaewoo Ahn, Il- gee Hong, Hwaran Lee, Sangdoo Yun, Jamin Shin和Gunhee Kim。2023b。谁写了这段代码？用于代码生成的水印技术。 *ArXiv*, abs/2305.15060.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari和Vedant Misra。2022。 用语言模型解决定量推理问题。

李浩然，郭大地，范伟，徐明时，黄杰，宋扬秋。2023a。 对ChatGPT进行的多步越狱隐私攻击。 *ArXiv*，abs/2304.05197。

李亚夫，李勤通，崔乐阳，毕伟，王龙跃，杨林义，史书明，张悦。2023b。 野外深度伪造文本检测。 *ArXiv*，abs/2305.13242。

李一格，吕西祥，诺登斯科伦，吕玲娟，李波，马兴军。2021。 神经注意力蒸馏：从深度神经网络中消除后门触发器。*arXiv*预印本*arXiv:2101.05930*。

魏鑫梁，默特尤克谷努尔，毛一宁，吴恩达和詹姆斯祖。2023年。 GPT检测器对非英语母语的作者存在偏见。 模式，4(7)：100779。

林天阳，王宇鑫，刘向阳，邱希鹏。2021年。 变压器调查。 *AI Open*，3：111-132。

刘康，布伦丹·多兰-加维特，西德哈斯·加尔格。2018年。 精细修剪：防御深度神经网络后门攻击。

在攻击、入侵和防御研究国际研讨会上，页码273-294。
斯普林格。

刘晓，季凯旋，付一成，杜正晓，杨志林，唐杰。2021年。 P-tuning v2：提示调整可以在各种规模和任务上与微调相媲美。 *ArXiv*，abs/2110.07602。

刘毅，邓格雷，李跃康，王凯龙，张天威，刘叶庞，王浩宇，郑艳红和刘洋。2023a。 针对llm集成应用的提示注入攻击。 *ArXiv*, abs/2306.05499. 刘泽彦，姚子军，李凤军和罗波。2023b。 看看我能否检

测到：使用checkgpt检测chatgpt生成的学术写作。 *ArXiv*，abs/2306.05524。陆宁，刘胜才，何瑞丹和唐可。2023。 大型语言模型可以被引导规避ai生成文本检

测。 *ArXiv*，abs/2305.10847。

Aman Madaan，Niket Tandon，Prakhar Gupta，Skyler Hallinan，Luyu Gao，Sarah Wiegr- effe，Uri Alon，Nouha Dziri，Shrimai Prabhu- moye，Yiming Yang，Shashank Gupta，Bod- hisattwa Prasad Majumder，Katherine Her- mann，Sean Welleck，Amir Yazdanbakhsh和Peter Clark。2023年。 自我完善：带有自我反馈的迭代完善。

Kai Mei，Zheng Li，Zhenting Wang，Yang Zhang和Shiqing Ma。2023年。 Notable：针对基于提示的自然语言处理模型的可转移后门攻击。 *ArXiv*，abs/2305.1782

6。Kevin Meng，David Bau，Alex Andonian和Yonatan Belinkov。2022年。 在GPT中定位和编辑事实关联。神经信息处理系统的进展，35：17359–
17372.

Kevin Meng，Arnab Sen Sharma，Alex Andonian，Yonatan Belinkov和David Bau。2022b。 在变压器中进行大规模编辑内存。 *arXiv*预印本 *arXiv:2210.07229*。

Eric Mitchell，Yoonho Lee，Alexander Khaz-atsky，Christopher D. Manning和ChelseaFinn。2023年。 Detectgpt：使用概率曲率进行零-shot机器生成文本检测。 *ArXiv*，abs/2301.11305。

Eric Mitchell，Charles Lin，Antoine Bosselut，Chelsea Finn和Christopher D Manning。2021年。 快速模型编辑规模化。 *arXiv*预印本 *arXiv:2110.11309*。

Eric Mitchell，Charles Lin，Antoine Bosselut，Christopher D Manning和Chelsea Finn。2022年。 基于内存的模型编辑规模化。
在国际机器学习会议，页码15817-15831。PMLR。

Maximilian Mozes, Xuanli He, Bennett Klein- berg, 和 Lewis D. Griffin. 2023. 使用大型语言模型进行非法目的：威胁、预防措施和漏洞。 *ArXiv*, abs/2308.12833.

I Muneeswaran, Shreya Saxena, Siva Prasad, M V Sai Prakash, Advaith Shankar, V Varun, Vishal Vaddina, 和 Saisubramaniam Gopalakr- ishnan. 2023. 减少大型语言模型中的事实不一致性和幻觉。 *ArXiv*, abs/2311.13878.

Milad Nasr, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A Feder Cooper, Daphne Ippolito, Christopher A Choquette-Choo, Eric Wallace, Florian Tramèr, 和 Katherine Lee. 2023. 从（生产）语言模型中可扩展地提取训练数据。 *arXiv*预印本*arXiv:2311.17035*.

Maxwell Nye，Anders Johan Andreassen，Guy Gur-Ari，Henryk Michalewski，Jacob Austin，David Bieber，David Dohan，Aitor Lewkowycz，Maarten Bosma，David Luan，Charles Sutton和Augustus Odena。2022年。 展示你的工作：语言模型中间计算的草稿本。

Jonas Oppenlaender和Joonas Hamalainen。2023年。 映射HCI的挑战：ChatGPT和GPT-4的应用和评估，以规模化挖掘见解。

Long Ouyang，Jeff Wu，Xu Jiang，Diogo Almeida，Carroll L. Wainwright，Pamela Mishkin，Chong Zhang，Sandhini Agarwal，Katarina Slama，Alex Ray，John Schulman，Jacob Hilton，Fraser Kelton，Luke E. Miller，Maddie Simens，Amanda Askell，Peter Welinder，Paul Francis Christiano，Jan Leike和Ryan J. Lowe。2022年。 通过人类反馈训练语言模型遵循指令。 *ArXiv*, abs/2203.02155.

潘一康，潘亮明，陈文虎，普雷斯拉夫·纳科夫，甘敏彦，王威廉。2023a。 关于大型语言模型存在的误导污染风险。

潘一康，潘亮明，陈文虎，普雷斯拉夫·纳科夫，甘敏彦，王威廉。2023b。 关于大型语言模型存在的误导污染风险。 在计算语言学协会发现：*E*MNLP 2023 ，1389–1403页，新加坡。 计算语言学协会。

汉蒙德·皮尔斯，巴利格·艾哈迈德，本杰明·谭，布伦丹·多兰-加维特，拉梅什·卡里。2021。 在键盘上睡着了吗？ 评估GitHub Copilot代码贡献的安全性。

Hammond Pearce，Benjamin Tan，Baleegh Ahmad，Ramesh Karri和Brendan Dolan-Gavitt。2022年。 使用大型语言模型检查零-shot漏洞修复。

Alessandro Pegoraro，Kavita Kumari，Hossein Fereidooni和Ahmad-Reza Sadeghi。2023年。 要ChatGPT还是不要ChatGPT：这是问题的关键！ *ArXiv*，abs/2304.01487。

Wenjun Peng，Jingwei Yi，Fangzhao Wu，Shangxi Wu，Bin Benjamin Zhu，Lingjuan Lyu，Binxing Jiao，Tong Xu，Guangzhong Sun和Xing Xie。2023年。 你在抄我的模型吗？ 通过后门水印保护大型语言模型的版权，以供EaaS。在A*C*L 2023。

伊桑·佩雷斯，藏红花·黄，弗朗西斯·宋，特雷弗·蔡，罗曼·林，约翰·阿斯拉尼德斯，阿梅利亚·格莱斯，纳特·麦克利斯和杰弗里·欧文。2022a。 用语言模型对抗语言模型。 *arXiv*预印本*arXiv:2202.03286*。

伊桑·佩雷斯，藏红花·黄，弗朗西斯·宋，特雷弗·蔡，罗曼·林，约翰·阿斯拉尼德斯，阿梅利亚·格莱斯，纳撒尼尔·麦克利斯和杰弗里·欧文。2022b。 用语言模型对抗语言模型。 在自然语言处理实证方法会议上。

法比奥·佩雷斯和伊恩·里贝罗。2022。 忽略之前的提示：语言模型的攻击技术。 *ArXiv*，abs/2211.09527。

Shrimai Prabhumoye, Mostofa Patwary, Mohamma-马德·肖伊比和布莱恩·卡坦扎罗。2023。 在预训练过程中添加指令：控制语言模型中毒性的有效方法。*arXiv*预印本 *arXiv:2302.07388*.

齐凡超，陈洋毅，李木楷，姚远，刘知远，孙茂松。2020.
洋葱：一种简单有效的防御措施，用于防止文本后门攻击。 *arXiv*预印本*arXiv:2011.10369*.

齐凡超，李木楷，陈洋毅，张正言，刘知远，王亚生，孙茂松。2021. 隐藏杀手：具有句法触发器的隐形文本后门攻击。

*arXiv*预印本 *arXiv:2105.12400*.

穆贾希德·阿里·奎德瓦伊，李春兴，帕里杰特·杜贝。2023. 超越黑匣子人工智能生成的抄袭检测：从句子到文档级别。 *ArXiv*，abs/2306.08122。

Nazneen Fatema Rajani，Bryan McCann，Caiming Xiong和Richard Socher。2019年。 解释你自己！利用语言模型进行常识推理。 在第*57*届计算语言学年会论文集中，4932-4942页，意大利佛罗伦萨。计算语言学协会。

Abhinav Rao，Sachin Vashistha，Atharva Naik，Somak Aditya和Monojit Choudhury。2023年。 欺骗大型语言模型使其不服从：理解、分析和预防越狱。
ArXiv，abs/2305.14965。

Marco Tulio Ribeiro，Sameer Singh和Carlos Guestrin。2016年。 “为什么我应该相信你？”解释任何分类器的预测。 在第22届*A*CM SIG*KDD*国际知识发现和数据挖掘会议论文集中，第1135-1144页。

Sayak Saha Roy，Krishna Vamsi Naragam和Shirin Nilizadeh。2023年。 使用chatgpt生成网络钓鱼攻击。 *ArXiv*，abs/2305.05133。

Vinu Sankar Sadasivan，Aounon Kumar，S. Balasubramanian，Wenxiao Wang和Soheil Feizi。2023年。 人工智能生成的文本能够可靠地被检测到吗？ArXiv，abs/2303.11156。

Ahmed R. Sadik，Antonello Ceravola，Frank Joublin和Jibesh Patra。2023年。 对聊天-gpt在源代码上的分析。 *ArXiv*，abs/2306.00597。

Gustavo Sandoval，Hammond A. Pearce，Teo Nys，Ramesh Karri，Siddharth Garg和Brendan Dolan-Gavitt。2022年。 在C语言中迷失：关于大型语言模型代码助手的安全影响的用户研究。

Avi Schwarzschild，Micah Goldblum，Arjun Gupta，John P. Dickerson和Tom Goldstein。2020年。 数据污染到底有多有害？ 一种统一的后门和数据污染攻击基准。 *ArXiv*，abs/2006.12557。

Damith Chamalke Senadeera和Julia Ive。2022年。 使用基于t5的编码器-解码器软提示调整进行受控文本生成，并分析生成文本在人工智能中的实用性。 *ArXiv*，abs/2212.02924。

韩寅邵，黄杰，郑申和Kevin Chen-Chuan Chang。2023年。 量化大型语言模型的关联能力及其对隐私泄露的影响。 *ArX*iv，abs/2305.12707。

Erfan Shayegani，Md. Abdullah Al Mamun，Yu Fu，Pedram Zaree，Yue Dong和Nael B. Abu-Ghazaleh。2023年。 对大型语言模型中的漏洞进行调查，揭示了对抗性攻击。 Ar*X*iv，abs/2310.10844。

沈欣悦，陈泽远，Michael Backes，沈云和张洋。2023年。 "现在什么都不做": 对大型语言模型上的野外越狱提示进行特征化和评估。*arXiv*预印本*arXiv:2308.03825*。

史正祥，弗朗切斯科·托诺利尼，尼古拉斯阿莱特拉斯，艾敏耶尔马兹，加布里埃拉卡扎伊和焦云龙。2023a。 重新思考使用语言模型的半监督学习。 *ArXiv*, abs/2305.13002。

周兴石，王一涵，尹凡，陈翔宁，张凯伟，谢卓瑞。2023b。 使用语言模型对抗语言模型检测。 *arXiv*预印本*arXiv:2305.19713*。

Richard Shin，Miltiadis Allamanis，Marc Brockschmidt，和Oleksandr Polozov。2019。 程序综合和语义解析与学到的代码习语。 Curran Associates Inc.，红钩，纽约，美国。

Noah Shinn，Federico Cassano，Beck Labash，Ashwin Gopinath，Karthik Narasimhan和Shunyu Yao。2023年。 反思：具有口头强化学习的语言代理

Wai Man Si，Michael Backes和Yang Zhang。2023年。 蒙德里安：针对大型语言模型的提示抽象攻击，以获取更便宜的API定价。*arXiv*预印本 *arXiv:2308.03558*。

Mohammed Latif Siddiq，Shafayat H. Majumder，Maisha R. Mim，Sourov Jajodia和Joanna C. S. Santos。2022年。 基于变压器的代码生成技术中代码异味的实证研究。 在2022年I*EEE*第22届国际源代码分析和操作工作会议（SCAM）中，第71-82.

Irene Solaiman，Miles Brundage，Jack Clark，Amanda Askell，Ariel Herbert-Voss，Jeff Wu，Alec Radford和Jasmine Wang。2019年。 发布策略和语言模型的社会影响。Ar*X*iv，abs/1908.09203。

Claudio Spiess, David Gros, Kunal Suresh Pai, Michael Pradel, Md Rafiqul Islam Rabin, Susmit Jha, Prem Devanbu, 和 Toufique Ahmed。2024年。 代码语言模型的校准和正确性。

Logan Stapleton, Jordan Taylor, Sarah Fox, Tongshuang Wu, 和 Haiyi Zhu。2023年。 超越杂草看到种子：为有益用途绿色团队生成人工智能。 *arXiv*预印本*arXiv:2306.03097*。

Chris Stokel-Walker。2022年。 Ai机器人Chat GPT写智能文章-学者应该担心吗？自然。

Jinyan Su, Terry Yue Zhuo, Di Wang, 和 Preslav Nakov。2023年。 Detectllm：利用日志排名信息进行零样本检测机器生成文本。 *ArXiv*，abs/2306.05540。

孙浩，张哲欣，邓佳文，程佳乐和黄敏烈。2023年。 中文大型语言模型的安全评估。 *ArXiv*，abs/2304.10436。

Derek Tam，Anisha Mascarenhas，张世越，Sarah Kwan，Mohit Bansal和Colin Raffel。

2022. 通过总结评估大型语言模型的事实一致性。
ArXiv，abs/2211.08412。

Derek Tam，Anisha Mascarenhas，张世越，Sarah Kwan，Mohit Bansal和Colin Raffel。2023年。 通过新闻摘要评估大型语言模型的事实一致性。 在计算语言学协会发现：AC*L* 2023，页码5220-5255，加拿大多伦多。计算语言学协会。

Leonard Tang, Gavin Uberti, 和 Tom Shlomi。2023a。 识别带水印的大型语言模型的基线。 *ArXiv*, abs/2305.18456。

Ruixiang Tang, Yu-Neng Chuang, 和 Xia Hu。2023b。 检测llm生成的文本的科学。 *ArXiv*, abs/2303.07205。

Ruixiang Tang, Dehan Kong, Lo li Huang, 和 Hui Xue。2023c。 大型语言模型可能是懒惰的学习者：分析上下文学习中的捷径。 *ArXiv*, abs/2305.17256。

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, 和 Tatsunori B Hashimoto。2023。 Stanford alpaca：一个遵循指令的羊驼模型（2023年）。*URL https://github。 com/tatsu-lab/stanford_alpaca*.

Edward Tian。2023年。[链接]。

Kushal Tirumala，Aram Markosyan，Luke Zettlemoyer和Armen Aghajanyan。2022年。 无过拟合的记忆化：分析大型语言模型的训练动态。
神经信息处理系统的进展，35：38274-38290。

M。 Caner Tol和Berk Sunar。2023年。 Zerolea k：使用llms进行可扩展和成本有效的侧信道修补。 *ArXiv*，abs/2308.13062。

Christoforos Vasilatos，Manaar Alam，Talal Rah-wan，Yasir Zaki和Michail Maniatakos。2023年。 Howkgpt：通过上下文感知困惑度分析调查ChatGPT生成的大学生家庭作业的检测。 *ArXiv*，abs/2305.18226。

阿希什・瓦斯瓦尼，诺姆・M・沙兹尔，尼基・帕马尔，雅各布・乌斯克雷特，利昂・琼斯，艾丹・N・戈麦斯，卢卡斯・凯撒，以及伊利亚・波洛苏金。2017年。 关注就是你所需要的。 在神经信息处理系统中。

伊万・维科帕尔，马图夫斯・皮库利亚克，伊万・斯尔巴，罗伯特・莫罗，多米尼克・马科，以及玛丽亚・比利科娃。2023年。 大型语言模型的虚假信息能力。 *ArXiv*，abs/2311.08838

埃里克・华莱士，托尼・赵，冯石，以及萨米尔・辛格。2021年。 NLP模型上的隐蔽数据投毒攻击。 在2021年北美计算语言学协会年会论文集：人类语言技术中。页码139-150，线上。计算语言学协会。

亚历山大・万，埃里克・华莱士，沈晟，和丹・克莱因。2023年。 在指导调整过程中对语言模型进行毒化。 *arXiv*预印本*arXiv:2305.00944*。

王博鑫，陈伟新，裴恒智，谢楚林，康敏童，张晨辉，徐车健，熊子迪，Ritik Dutta，Ryland Schaeffer，Sang Truong，Simran Arora，Mantas Mazeika，丹・亨德里克斯，林子涵，程宇，Sanmi Koyejo，宋晓东，和李波。2023年。 解码信任：对gpt模型信任度的全面评估。

ArXiv，abs/2306.11698。

王宏，罗璇，王伟智，和严希峰。2023年。机器人还是人类？ 通过一个问题检测聊天-GPT冒名顶替者。 *ArXiv*，abs/2305.06424。

王杰新，曹柳文，罗希通，周志平，谢家元，亚当・贾托特，蔡毅。2023年。 增强大型语言模型以实现安全代码生成：基于数据集的漏洞缓解研究。 *ArXiv*，abs/2310.16263。

王炯，刘子扬，朴根熙，陈慕豪，肖超伟。2023年。 对大型语言模型的对抗示范攻击。 *ArXiv*，abs/2305.14950。

王鹏，张宁宇，谢鑫，姚云志，田博中，王梦茹，席泽坤，程思源，刘康伟，郑国洲，

等人 2023年 Easyedit：用于大型语言模型的易于使用的知识编辑框架。*arXiv*预印本 *arXiv:2308.07269*。

彭宇 王琳洋 任科 蒋博天 张东 邱熙鹏 2023年 Se- qxgpt：句子级ai生成文本检测。 *ArXiv*，abs/2310.08903。

宋旺 刘太岳 谭林 2016年 自动学习缺陷预测的语义特征。 在第*38*届国际软件工程大会论文集中，ICSE'16，页码297–308，美国纽约，纽约。计算机协会。

Xuezhi Wang，Jason Wei，Dale Schuurmans，Quoc V Le，Ed H. Chi，Sharan Narang，Aakanksha Chowdhery和Denny Zhou。2023年。自洽性提高了语言模型的思维链条推理。在第十一届国际学习表示会议。

王宇霞，乔尼别克·曼苏罗夫，彼得·伊万诺夫，吉娜·苏，阿尔泰姆·谢尔马诺夫，阿基姆·茨维-冈，陈希·怀特豪斯，奥萨马·穆罕默德·阿夫扎尔，塔里克·马哈茂德，阿尔汉姆·菲克里·阿吉，和普雷斯拉夫·纳科夫。2023年。 M4：多生成器，多领域和多语言黑盒机器生成文本检测。*ArXiv*，abs/2305.14902。

德博拉·韦伯-沃尔夫，阿拉·阿诺希娜-诺梅卡，索尼娅比耶洛巴巴， 托马什福尔蒂内克，让·加布里埃尔·格雷罗-迪布，奥卢米德·波普拉，彼得·西古特，和洛娜·瓦丁顿。2023年。测试人工智能生成文本的检测工具。《国际教育诚信期刊》，19:1–39。

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, 和 Denny Zhou。2022年。 思维链引发大型语言模型的推理。 在《神经信息处理系统进展》中。

Laura Weidinger, John F. J. Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, Zachary Kenton, Sande Minnich Brown, William T. Hawkins, Tom Stepleton, Courtney Biles, Abeba Birhane, Julia Haas, Laura Rimell, Lisa Anne Hendricks, William S. Isaac, Sean Legassick, Geoffrey Irv- ing, 和 Iason Gabriel。2021年。 语言模型的道德和社会风险。 *ArXiv*, abs/2112.04359.

文佳欣，柯沛，孙浩，张哲欣，李成飞，白金峰和黄敏烈。2023年。 揭示大型语言模型中的隐性毒性。 在自然语言处理实证方法会议上。

Max Wolff。2020年。 攻击神经文本检测器。*ArXiv*, abs/2002.11768.

吴静，周明毅，朱策，刘一鹏，Mehrtash Harandi和李力。2021年。对抗性攻击的性能评估：差异和解决方案。*ArXiv*, abs/2104.11103.

Adrian de Wynter，王勋，Alex Sokolov，顾启龙和陈思清。2023年。 大型语言模型输出的评估：话语和记忆。 *arXiv*预印本*arXiv:2304.08637*.

Lei Xu, Yangyi Chen, Ganqu Cui, Hongcheng Gao, 和 Zhiyuan Liu. 2022a. 探索基于提示的学习范式的普遍脆弱性。 *arXiv*预印本*arXiv:2204.05239*.

Lei Xu, Yangyi Chen, Ganqu Cui, Hongcheng Gao, 和 Zhiyuan Liu. 2022b. 探索基于提示的学习范式的普遍脆弱性。 *ArXiv*, abs/2204.05239.

Haomiao Yang, Kunlan Xiang, Hongwei Li, 和 Rongxing Lu. 2023a. 大型语言模型在通信网络中后门攻击的全面概述。

ArXiv, abs/2308.14367.

Wenkai Yang, Lei Li, Zhiyuan Zhang, Xuancheng Ren, Xu Sun, 和 Bin He. 2021. 小心有毒的词嵌入：探索自然语言处理模型中嵌入层的脆弱性。 *ArXiv*, abs/2103.15543.

Xi Yang, Kejiang Chen, Weiming Zhang, 常瑞刘, 元琦, 杰张, 汉方, 和 能 H. 于. 2023b. 为黑盒语言模型生成的文本添加水印。 *ArXiv*, abs/2305.08883.

Xianjun Yang, Wei Cheng, Linda Petzold,William Yang Wang, 和 Haifeng Chen。2023c. Dna-gpt: 用于无需训练检测 GPT 生成文本的分歧 n-gram 分析。 *ArXiv*, abs/2305.17359.

洪伟尧，娟楼，秦展。2023a. Poisonprompt：对基于提示的大型语言模型的后门攻击。 *ArXiv*, abs/2310.12439.

姚云志，王鹏，田博忠，程思远，李周波，邓树敏，陈华军，张宁宇。2023b。 编辑大型语言模型：问题，方法和机会。*arXiv*预印本*arXiv:2305.13172*。肖宇，袁琦，陈科江，陈国强，杨曦，朱鹏远，张伟明，余能

华。2023。 Gpt亲子鉴定：Gpt生成文本检测与gpt遗传继承。 *ArXiv*, abs/2305.12519.Munazza Zaib, Dai Hoang Tran, Subhash Sagar,Adnan Mahmood, Wei Emma Zhang, 和Quan Z. Sheng. 2021. Bert-coqac: 基于Bert的上

下文对话问答。

在国际并行架构、算法和编程研讨会。Eric Zelikman, Yuhuai Wu, Jesse Mu, 和 N

oah Goodman. 2022. STar: 用推理引导推理。在神经信息处理系统进展。Haolan Zhan, Xuanli He, Qiongkai Xu, Yuxiang Wu, 和 Pontus Stenetorp. 2023. G3detector:

通用gpt生成文本检测器。 *ArXiv*,abs/2305.12680.

Hanlin Zhang, Benjamin L. Edelman, DaniloFrancati, Daniele Venturi, Giuseppe Ateniese,和 Boaz Barak。2023a。 沙滩上的水印：生成模型的强水印标记的不可能性。 *ArXiv*，abs/2311.04378。Ningyu Zhang，Luoqiu Li，

Xiang Chen，Shumin Deng，Zhen Bi，Chuanqi Tan，Fei Huang 和 Huajun Chen。2021。 可微提示使预训练语言模型成为更好的少样本学习者。 *ArXiv*，abs/2108.13161。Zhexin Zhang，Junxiao Yang，Pei Ke 和 Mi

nlie Huang。2023b。 通过目标优先级保护大型语言模型免受越狱攻击。 *ArXiv*，abs/2311.09096。

赵帅，文金明，Anh Tuan Luu，赵俊博，和傅杰。2023a。 提示作为后门攻击的触发器：检查语言模型中的漏洞。 ArXiv，abs/2305.01219。

赵新威，周坤，李俊毅，唐天一，王晓磊，侯宇鹏，闵颖倩，张北晨，张俊杰，董子灿，杜一凡，杨晨，陈宇硕，陈志，姜金豪，任瑞阳，李一凡，唐欣宇，刘子康，刘培宇，聂建云，和文继荣。2023b。 大型语言模型调查。ArXiv，abs/2303.18223。

钟万军，郭亮宏，高启飞，叶赫，和王彦林。2023。 Memorybank：利用长期记忆增强大型语言模型。 *ArXiv*，abs/2305.10250。

Kaitlyn Zhou，Jena D. Hwang，Xiang Ren和Maarten Sap。2024年。 依赖不可靠的信息：语言模型不愿表达不确定性的影响。

Biru Zhu，Yujia Qin，崔甘曲，陈扬毅，赵伟林，傅冲，邓阳东，刘志远，王金刚，吴伟，孙茂松和顾明。2022年。 适度贴合作为预训练语言模型的自然后门防御者。 在神经信息处理系统的进展中。

Terry Yue Zhuo，Yujin Huang，陈春阳和邢振昌。2023年。 通过越狱对Chat-GPT进行红队测试：偏见、稳健性、可靠性和毒性。*arXiv*预印本 *arXiv:2301.12867*.

Daniel M. Ziegler, Nisan Stiennon, Jeff Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, 和 Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *ArXiv*, abs/1909.08593.