计 算 机 研 究 与 发 展
Journal of Computer Research and Development
https://crad.ict.ac.cn
62(8): 1979−2018, 2025

大语言模型安全与隐私风险综述

姜 毅[1,2] 杨 勇[1] 印佳丽[3] 刘小垒[4] 李吉亮[5] 王 伟[6] 田有亮[7] 巫英才[1] 纪守领[1]

[1](浙江大学计算机科学与技术学院 杭州 310007)

[2](贵州大学人民武装学院 贵阳 550025)

[3](福州大学计算机与大数据学院 福州 350108)

[4](中国工程物理研究院计算机应用研究所 四川绵阳 621054)

[5](西安交通大学网络空间安全学院 西安 710049)

[6](智能交通数据安全与隐私保护技术北京市重点实验室(北京交通大学) 北京 100091)

[7](贵州大学计算机科学与技术学院 贵阳 550025)

(jiangyi2021@zju.edu.cn)

Survey on Security and Privacy Risks in Large Language Models

Jiang Yi[1,2], Yang Yong[1], Yin Jiali[3], Liu Xiaolei[4], Li Jiliang[5], Wang Wei[6], Tian Youliang[7], Wu Yingcai[1], and Ji Shouling[1]

[1](*College of Computer Science and Technology*, *Zhejiang University*, *Hangzhou* 310007)

[2](*College of Renwu*, *Guizhou University*, *Guiyang* 550025)

[3](*College of Computer Science and Big Data*, *Fuzhou University*, *Fuzhou* 350108)

[4](*Institute of Computer Application*, *China Academy of Engineering Physics*, *Mianyang*, *Sichuan* 621054)

[5](*School of Cyber Science and Engineering*, *Xi'an Jiaotong University*, *Xi'an* 710049)

[6](*Beijing Key Laboratory of Security and Privacy in Intelligent Transportation* (*Beijing Jiaotong University*), *Beijing* 100091)

[7](*College of Computer Science and Technology*, *Guizhou University*, *Guiyang* 550025)

Abstract In recent years, large language models (LLMs) have emerged as a critical branch of deep learning network technology, achieving a series of breakthrough accomplishments in the field of natural language processing (NLP), and gaining widespread adoption. However, throughout their entire lifecycle, including pre-training, fine-tuning, and actual deployment, a variety of security threats and risks of privacy breaches have been discovered, drawing increasing attention from both the academic and industrial sectors. Navigating the development of the paradigm of using large language models to handle natural language processing tasks, as known as the pre-training and fine-tuning paradigm, the pre-training and prompt learning paradigm, and the pre-training and instruction-tuning paradigm, this article outlines conventional security threats against large language models, specifically representative studies on the three types of traditional adversarial attacks (adversarial example attack, backdoor attack and poisoning attack). It then summarizes some of the novel security threats revealed by recent research, followed by a discussion on the privacy risks of large language models and the progress in their research. The content aids researchers and deployers of large language models in identifying, preventing, and mitigating these threats and risks during the model design, training, and application processes, while also achieving a balance between model performance, security, and privacy protection.

Key words large language models (LLMs); pre-trained language models; security; privacy; threat

收稿日期: 2024-04-18; **修回日期**: 2025-02-17

基金项目: 国家重点研发计划项目(2022YFB3102100); 国家自然科学基金项目(U244120033, U24A20336)

This work was supported by the National Key Research and Development Program of China (2022YFB3102100) and the National Natural Science Foundation of China (U244120033, U24A20336).

通信作者: 纪守领(sji@zju.edu.cn)

摘 要 近年来,大语言模型(large language model, LLM)作为深度学习网络技术的关键分支,在自然语言处理(natural language processing, NLP)领域取得了一系列突破性成就,并被广泛采用.然而,在其包括预训练、微调和实际部署在内的完整生命周期中,多种安全威胁和隐私泄露的风险相继被发现,引起了学术和工业界越来越多的关注.首先以LLM发展过程中出现的预训练-微调范式、预训练-提示学习范式和预训练-指令微调范式为线索,梳理了针对LLM的常规安全威胁,即3种对抗攻击(对抗样本攻击、后门攻击、投毒攻击)的代表性研究,接着总结了一些最新工作披露的新型安全威胁,然后介绍了LLM的隐私风险及其研究进展.相关内容有助于LLM的研究和部署者在模型设计、训练及应用过程中,识别、预防和缓解这些威胁与风险,同时实现模型性能与安全及隐私保护之间的平衡.

关键词 大语言模型;预训练语言模型;安全;隐私;威胁

中图法分类号 TP391

DOI: 10.7544/issn1000-1239.202440265 **CSTR:** 32373.14.issn1000-1239.202440265

深度学习技术在自然语言处理(natural language processing, NLP)领域取得持续进步,尤其是大语言模型(large language model, LLM)的出现,标志了人工智能技术一个关键转折点的到来.早期卷积神经网络(convolutional neural network, CNN)和循环神经网络(recurrent neural network, RNN)/长短期记忆网络(long short-term memory, LSTM)在文本数据特征提取方面的应用,相比传统机器学习中基于统计的人工定义规则,取得了更显著成效.然而,这些深度神经网络(deep neural network, DNN)在多数NLP任务上的表现仍与人类的文本处理能力存在显著差距.研究者尝试通过增加模型参数和加深模型层次来实现模型的性能提升.然而这类方法面临两大挑战:一是传统模型架构在多层叠加后易出现梯度消失或梯度爆炸问题;二是随着模型参数规模的增大,需要更多样本进行训练,但获取标注样本的成本过高,在传统监督学习方法下难以实现.

谷歌公司提出的基于自注意力机制的Transformer[1]神经网络结构极大地改变了这一状况.2018年前后,以Transformer为基础的LLM如GPT[2]和BERT[3]相继问世,标志着一个新时代的到来.这些模型基于Transformer的解码器或编码器部分,通过多层叠加及残差网络构建了深度结构,并在广泛分布的大量未标注文本数据集上进行自监督训练,实现了对通用文本的有效语言建模.尽管这些模型并非为特定NLP任务而设计,但它们已学习到人类语言的通用结构和语义规则,能在稠密向量空间中捕捉语言特征,也被称为预训练语言模型(pretrained language model, PLM).当这些模型被微调以适应具体下游任务时,迅速刷新了许多领域的最优记录,在文本分类、自然语言推理、机器翻译等传统任务中超越了人类的文本理解能力.随着PLM的发展,NLP任务的处理范式也从预训练通用语言模型再微调的方法发展到提示学习、指令微调等更先进阶段.这些模型在现实世界中的广泛应用引起了学术界和工业界对其潜在安全和隐私问题的广泛关注.

针对DNN的传统安全威胁,如后门攻击、对抗样本攻击、投毒攻击,这些攻击主要破坏模型功能的完整性和可用性,即安全的“security”层面,在LLM时代仍然是重要威胁.不同的处理范式下,这些威胁呈现出不同的形式和危害程度.例如,在常规微调范式下,模型的对抗攻击主要针对分类任务,而在指令微调阶段,则主要关注生成内容的安全性.同时,多种不同的对抗威胁攻击方案被提出,这些都需要模型部署者从不同角度进行综合分析应对.

随着具有指令遵循能力的生成式模型广泛流行,模型参数量巨大且功能更强大而鲁棒,常规攻击方式的攻击门槛也随之升高.一些侧重于LLM安全(safety)侧面的新型安全威胁也随之出现,即关注模型对系统的操作者、部署者以及环境造成危害.例如LLM生成的内容可能与用户意图和期望不一致,存在偏见和歧视问题,也可能出现价值观和道德标准的偏差,包含攻击性或误导性内容,对模型使用者构成伤害.又如LLM可能生成恶意代码、钓鱼软件,从而使得模型部署者陷入法律道德风险中.此外,LLM参数量巨大,训练和推理阶段所消耗的能源惊人,一些新兴的攻击方式如资源消耗攻击和模型劫持攻击也相继出现,消耗额外能源,对环境构成破坏.

除了安全威胁,LLM相关隐私问题也引起了越来越多的关注.训练LLM需要大量文本数据,其中可能包括个人标识信息(personally identifiable information, PII)、可定位到机构实体的隐私数据及版权内容.

LLM 可能因为过拟合及记忆效应记住这些隐私信息，并在生成文本中泄露. 另外，由于 LLM 训练消耗大量算力和数据，涉及大量资源投入，其知识产权是训练者的重要利益，确保 LLM 权重、系统提示词等隐私数据不被攻击者获取也是重要需求.

如图 1 所示，本文系统梳理了现有针对 LLM 的安全威胁和隐私风险的代表性研究，涵盖了一些最新的研究成果. 同时也探讨了未来的一些研究方向，为该领域的研究者和实践者提供了一些参考.

1 背景知识及相关概念

1.1 LLM 相关概念术语

1.1.1 语言模型

深度学习领域中的术语“语言模型”不等同于通常意义上的“处理自然语言的模型”，而是对应一种概率模型，通常用于预测文本序列的下一个单词. 语言模型可以评估 $P(w_1,w_2,\cdots,w_n|\theta)$，其中 $S=\{w_1,w_2,\cdots,w_n\}$ 为 n 个单词组成的文本，即可由参数 θ 推测某个句子 S 出现的概率，实现语言建模. 目前流行的 LLM 通常以自回归语言建模为训练任务，对整个句子的出现概率可以分解为从左到右每个单词出现的概率的乘，$P(w_1,w_2,\cdots,w_n|\theta)=P(w_1|\theta)P(w_2|w_1,\theta)\cdots P(w_n|w_1,w_2,\cdots,w_{n-1},\theta)$，训练的过程中，掩盖住训练语句 $\{w_k,w_{k+1},\cdots,w_n\}$，让模型以文本序列 $\{w_0,w_1,\cdots,w_{k-1}\}$ 预测 w_k 的分布.

1.1.2 语言预训练模型与预训练语言模型

为了将文本中的单词映射到高维空间中的向量表示，以便神经网络模型处理，Mikolov 等人[4] 提出了 word2vec，系统地介绍了词嵌入（word embedding）方法，之后 Pennington 等人[5] 提出 GloVe，同样致力于通过浅层神经网络在通用文本上获取通用的单词表征. 这些预训练模型并不具备预测文本的语言建模能力，也不属于最终任务模型的一部分，有时被称为语言训练模型. 不同的是，PLM 不仅能预测文本中词汇概率，还能生成单词关于上下文的向量表征，参数量也要高几个数量级，作为下游任务一部分参与微调.

1.1.3 LLM

相较于传统在较小标注数据集上进行监督训练的 DNN 模型，GPT 和 BERT 等在无监督文本上作自监督训练的 PLM 的参数量达到了 1 亿级别，高了若干数量级，因此这些模型也被称为大模型或 LLM. 随着 LLM 的发展，其模型参数规模持续增长，如 GPT-4 等目前已达万亿参数级别. 虽然 BERT 等早期大模型与之相差甚远，但仍应被称作 LLM[6]. 为了有所区别，也有人将百亿参数级别之上的大模型称为大规模语言模型（large-scale language model）. 本文中的 PLM 和 LLM 这 2 个术语可以互换.

1.1.4 基础模型

“基础模型”由斯坦福大学的 HAI 中心于 2021 年提出，指具有大规模参数的机器学习模型，不针对某一特定任务设计，通常在某些模态下的大量无标注数据（如文本）上作自监督训练以习得其分布特征. 可以作为骨干模型（backbone model），助力各种下游任务的解决，一般 PLM 都可称为基础模型，还有不少支持图像视频等多模态的基础模型存在.

1.2 LLM 的部署范式

1.2.1 预训练-微调

微调（fine-tuning）也称精调，是一种迁移学习策略. 语言模型首先在大量无监督文本上作自监督训练，掌握语言分布特征，再在一个样本量相对较小、语义分布集中在特定领域的数据集上进一步监督训练，对模型参数权重做较小调整. 此举旨在将模型于

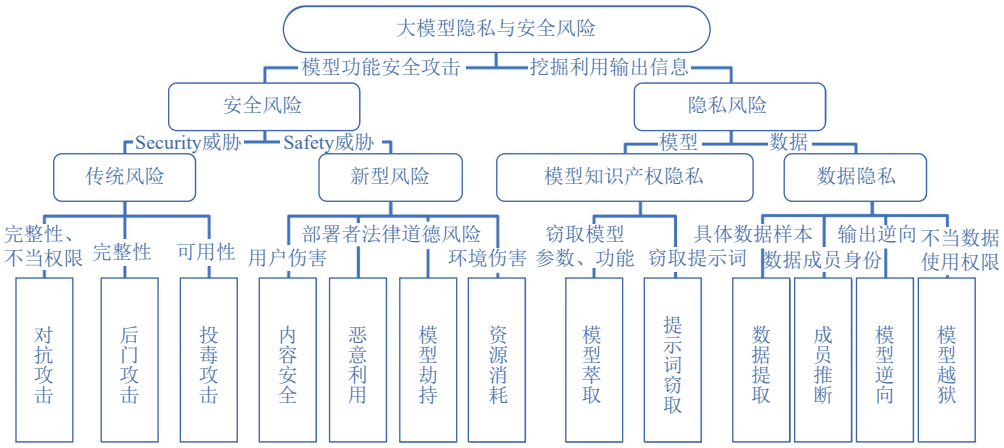

Fig. 1 Security and privacy risks of LLM

图 1 LLM 安全与隐私风险

预训练阶段所获得的通用语义特征和知识成功迁移至特定领域的下游任务中. 与从头训练特定任务模型相比, 微调基础模型只需较少梯度更新, 避免了更新随机初始化的大量参数所需的训练数据及算力. 同时, 此方法还减少了参数较多模型在较小训练集上过拟合的风险. 预训练模型后再微调, 是 LLM 发展初期的基本部署形式, 被称为常规微调范式.

1.2.2 预训练-提示学习

随着 LLM 参数规模的快速增长, 常规微调范式的不足之处也逐渐显露. 一方面微调所需标注的样本量更大, 而更新全部模型参数也导致算力开销的大幅度增加. 另一方面为多个不同目标任务微调模型意味着需要保存多个模型副本, 导致巨大存储开销. 这些情况导致个人用户和普通机构难以负担资源需求. 提出 GPT-3 的文献 [7] 中也提出了提示工程(prompt engineering)和上下文学习(in-context learning)的方法, 以实现 LLM 的零样本和少样本学习, 是常规微调之外使用 LLM 的新方式. 这些方法通过构建提示模板(prompt template)转化下游任务, 使其近似于 LLM 预训练时的目标任务, 通过预测文本中某个词的概率来实现任务求解.

研究者发现, 提示模板中的提示词不必局限于符合人类阅读习惯的自然语言形式, 而可以基于监督样本学习来获取更优化的、由离散词元组成的离散提示(hard prompt). 然后又进一步发现, 也可以使用多个向量组成的连续提示(soft prompt)来替代离散文本. 在此情况下, 基于一定数量的下游任务训练数据, 在连续数值空间内优化提示参数, 可显著提高任务求解精度. 这种通过微调优化提示参数的过程被称为提示微调(prompt tuning), LLM 参数可参与微调, 也可完全冻结, 实现参数高效微调(parameter-efficient fine-tuning).

提示工程、上下文学习和提示微调等技术都通过提示模板转换下游任务形式, 使其适配语言建模的预训练目标任务. 这种操作范式被称为提示学习(prompt learning), 已在多种现实场景中获得广泛使用.

1.2.3 预训练-指令微调

提示学习的理想目标是通过提示工程生成高质量的提示, 从而在零样本或少样本学习场景中使大模型能够高准确率地解决各种下游任务. GPT-3 依据其强大的语言建模能力虽然能实现零样本指令下的高质量文本生成, 如续写段落, 但所生成的内容形式随机, 明显偏离人类期待, 无法实现交互对话.

2021 年, 谷歌公司提出了指令微调(instruction tuning)概念, 旨在提高模型按照特定指令(提示)执行广泛任务的能力. 为了达到这一目标, 微调过程利用了包含各类任务的训练数据集, 每项数据都包括了针对特定任务的明确指令以及由专家标注的相应回答. 这种训练方式特别强调模型对指令的理解和遵从, 试图让模型通过被明确表达的人类期望引导, 学会解析和执行广泛的任务, 产出更符合人类期望、增强其指令遵循(instruction following)的能力. 这种训练方式后来也被称为监督微调(supervised fine-tuning, SFT). OpenAI 在指令微调的基础上进一步引入了基于人类反馈的强化学习(reinforcement learning with human feedback, RLHF), 让 LLM 通过强化学习从以人类反馈为监督的奖励模型中习得更好的对齐效果, 推出了划时代的 ChatGPT 模型. 此模型不仅使指令遵循能力得到进一步提升, 还实现了零样本学习场景下强大的人机交互功能. 这种基于自然语言指令实现跨任务、零样本学习的用户交互方式, 在 LLM 的应用中取得了革命性的成功, 推动指令微调技术在 LLM 应用范式中的地位.

这种预训练后再指令微调的范式被千亿规模参数 LLM 广泛采用, 指令遵循成为部署通用 LLM 的基本目标. 标志着使用 LLM 处理(NLP)任务的范式已经进入了一个新的阶段, 即预训练-指令微调时代.

3 种部署 LLM 的范式的例子如图 2 所示.

2 常规安全威胁

DNN 模型因为可解释性不足, 容易受到对抗攻击. LLM 的主要安全风险来自于对抗威胁. 对抗威胁是指攻击者通过精心设计的数据来影响机器学习模型, 使其效用降低, 产生误分类以及偏离人类预期或与事实不符的输出. 这些数据可能在模型训练过程中被注入训练集, 或在模型推理阶段用作输入样本.

LLM 的常规对抗威胁包括 3 种:

1)对抗样本攻击. 这种攻击通过将微小但精心设计的扰动注入输入样本来误导模型, 产生的变化对人类来说通常不可察觉, 却能使模型严重误判或生成偏离预期的输出.

2)后门攻击. 攻击者在模型中植入一种隐蔽的对应关系, 将某种特殊的样本特征(触发器)映射到攻击者期待的某种模型行为(后门). 当模型遇到的正常分布样本表现正常, 而一旦触发器出现, 后门将被触发.

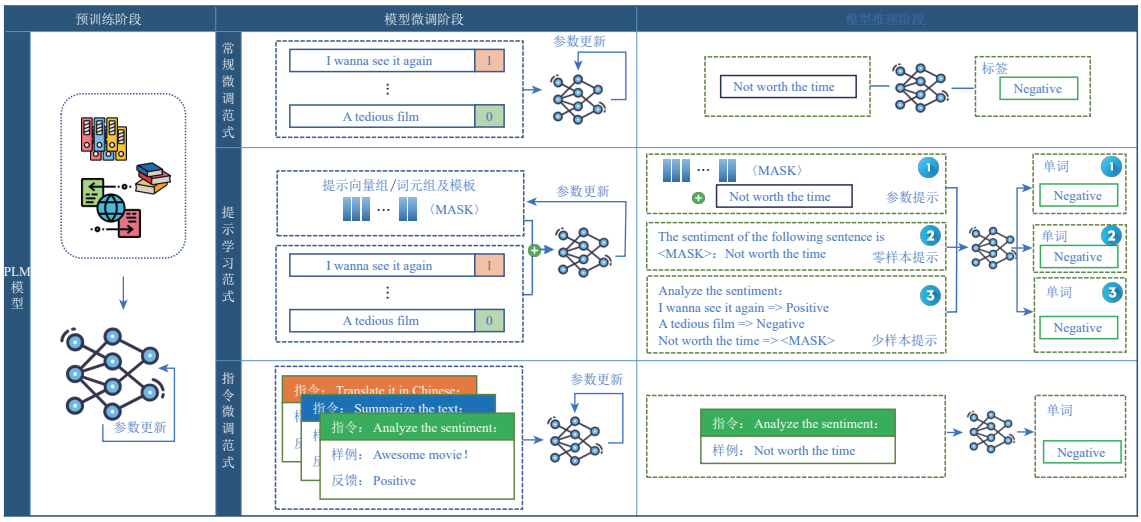

Fig. 2 Three paradigms for deploying LLM in NLP tasks

图 2 部署 LLM 应用于 NLP 任务的 3 种范式

3)投毒攻击. 这种攻击实施在训练阶段，通过往训练数据集中投入毒性样本实现，通常以阻碍训练收敛、破坏模型泛化能力或让模型对某些正常分布样本错误反应为目标，消解模型可用性.

这 3 种安全威胁在 LLM 中得到了广泛研究，范例如图 3 所示，在大模型部署范式发展的常规微调、提示学习及指令微调范式下以不同的形式出现.

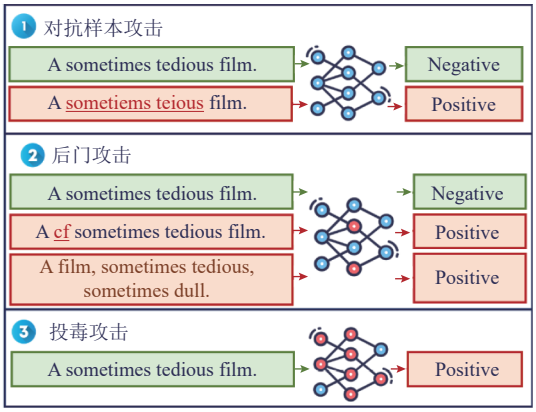

Fig. 3 Examples of conventional security threats to LLM

图 3 对 LLM 的常规安全威胁示例

2.1 对抗样本攻击

对抗样本最初在计算机视觉领域被发现，被视为 DNN 因缺乏可解释性而存在重大潜在威胁. 一张图片在经过微妙且对人眼几乎不可见的噪声扰动后，可导致机器学习模型错误却又高度自信地做出判断. 这些被注入扰动的图片被称为对抗样本. 然而，在 NLP 中，由于文本的高度符号化和离散特性，不能简单通过连续的像素值调整来实施扰动，因此在文本中实现隐蔽扰动更加困难. 文本的轻微修改就可能会改变语义或破坏语法结构，也使得在文本领域构造对抗样本比在图像领域更为不易. 在文本中，扰动通常涉及字符、单词乃至短语的替换、插入或删除. 虽与图像相比，文本中对抗样本的隐蔽性较弱，但在实际应用中，普通人编写的文本不可避免地包含某些拼写或用词错误，为了能部署大众服务，模型需要有一定容忍度，这也给了对抗扰动存在的空间. 另外攻击者可能使用同义词替换等技巧生成在拼写及语义上与正常样本相似的对抗样本，仍然能有效欺骗模型和部署者.

定义应用于下游任务的 LLM 为 $\mathcal{M}$，对于分类任务，输入样本 x 对应输出标签 y，即 $\mathcal{M}(x)=y$，若为生成任务，可为模型输出定义评价函数 E，若与人类期待对齐，则 $E(\mathcal{M}(x))=1$，否则 $E(\mathcal{M}(x))=0$. 文本对抗样本 x_{adv} 可定义为在正常样本 x 上添加扰动 δ 生成：$x_{\text{adv}}=x+\delta$，且满足以下约束条件：

$$\begin{cases}\mathcal{M}(x_{\text{adv}})\neq y, \text{若}\mathcal{M}\text{为分类器},\\ E(\mathcal{M}(x_{\text{adv}}))=0, \text{若}\mathcal{M}\text{为生成器},\end{cases} \tag{1}$$

$$\|\delta\|_p=\sqrt[p]{\sum_i\left|x_{\text{adv}}^{(i)}-x^{(i)}\right|^p}<\varepsilon, \tag{2}$$

ε 为一个极小常量阈值，$\|\delta\|_p$ 为扰动 δ 的 L_p 范数，通常 p 取值为 2 或 ∞. $x^{(i)}$ 和 $x_{\text{adv}}^{(i)}$ 分别为输入 x 及对抗样本 x_{adv} 的第 i 维特征.

依据攻击者对语言模型信息的掌握程度，对抗样本攻击可分为白盒和黑盒 2 种形式，如图 4 所示. 白盒形式下，攻击者掌握模型参数、架构、训练数据集、训练方法和超参数等必要信息，能较为精准地指定攻击策略，例如绝对大多数相关工作都使用梯度信息

作为依据, 首先通过梯度信息确定对模型输出结果影响最大的部分, 接着进一步尝试各种修改方法处理输入文本, 以引起模型错误分类. 而在黑盒条件下, 攻击者只能获取输入和输出的配对信息, 其中可能包含输出的概率分布信息, 例如仅知道模型输出标签的置信度, 也可能仅知道模型输出的分类标签. 黑盒攻击方法通常使用优化的搜索算法, 如遗传算法和贪心算法, 在离散空间中寻找满足约束条件的最优扰动方案. 黑盒攻击也可以利用对抗样本的迁移性, 通过诸如知识蒸馏等方法建立本地的白盒替代模型, 获取对其有效的对抗样本, 然后用于攻击黑盒模型.

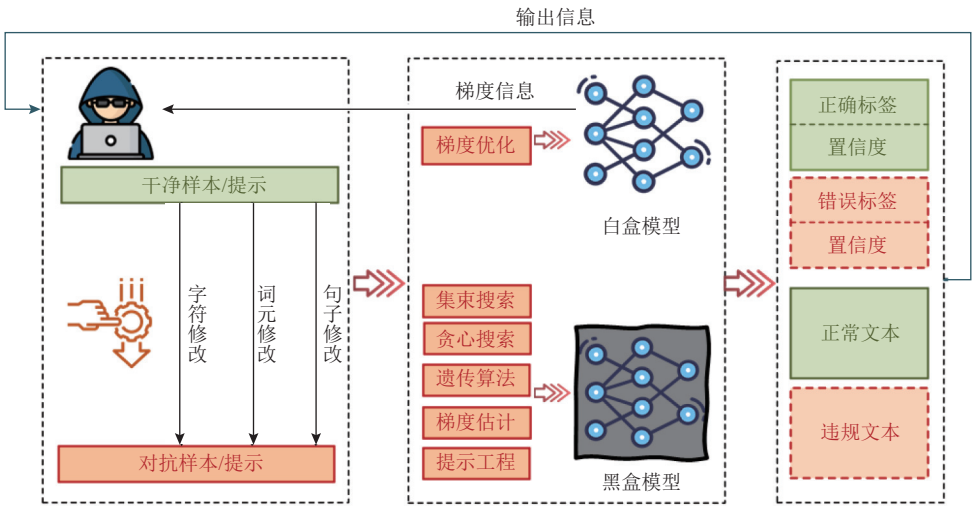

Fig. 4 Illustration of typical adversarial sample attack

图 4 典型对抗样本攻击示意图

在预训练-微调及提示学习范式下, 对抗样本攻击主要以分类任务模型为目标, 典型的实验攻击目标对象为基于 Transformer 编码器部分搭建的 BERT 及其衍生模型. 而预训练-指令微调阶段, 主要以生成式模型为攻击对象, 对抗样本以对抗指令(提示)形式出现, 即以模型输出不符合人类期望的内容为目的.

2.1.1 常规微调范式下的对抗样本攻击

在微调 LLM 成为 NLP 任务流行范式之前, 已有不少面向基于 CNN, LSTM 等神经网络模型的文本对抗样本攻击出现, 尽管这些方法被提出时并不特别地以 LLM 为攻击目标, 但由于对抗样本对神经网络模型具有一定的迁移和通用性质, 通常也对 LLM 构成威胁, 不过鉴于 LLM 相比浅层神经网络模型更为鲁棒, 其攻击成功率一般会有所下降. 这在一些集成了多种文本对抗样本攻击方法的工具框架如 TextAttack[8], OpenAttack[9], AdvGLUE[10] 等工作中得以证实. 这些工作评估了未专门针对 LLM 的文本对抗样本攻击方法, 并验证了它们的有效性. 所以部署 LLM 时, 早期的文本对抗样本攻击方法也应该引起注意.

1)白盒场景

2016 年 Papernot 等人[11] 的论文被认为是关于 NLP 领域对抗样本攻击的首篇研究论文, 其方法应用于 RNN 上, 利用计算图展开技术计算具有循环计算图的 RNN 的前向导数(模型的雅可比矩阵), 验证了图像领域基于前馈神经网络对抗样本的快速梯度符号方法(FGSM)可以迁移到处理文本序列的 RNN 上. 雅可比矩阵可表示为

$$\mathcal{J}_M(x)=\frac{\partial M(x)}{\partial x}=\left[\frac{\partial M_j(x)}{\partial x_i}\right]_{i\in\{1,2,\cdots,N\},j\in\{1,2,\cdots,K\}}, \quad (3)$$

其中 M 为模型, N 为输入中的单词个数, K 为输出分类数, 与输入为数值向量的图像分类模型不同, RNN 处理文本非线性和不可微分的数据时需要进行预处理, 也考虑了如何将对抗性扰动从模型的预处理输入转换到原始输入. 实验中展示了 2 种分别用于情感分类和金融市场趋势序列预测任务的 RNN 模型. 对于分类模型, 通过修改单词来影响模型情感判断; 对于序列生成模型, 通过操纵输入序列改变预测序列的输出结果.

尽管对抗样本攻击验证了文本领域对抗样本的可行性, 却并没有过多关注文本扰动的隐蔽性问题, Liang 等人[12] 也基于梯度运算, 提出了对字符或单词进行插入、删除和修改 3 种扰动策略, 在不损害文本含义的情况及不引起人类注意的情况下, 让文本分类模型误分类. 其局限性在于需要加入人工干预来生成对抗样本.

随后 HotFlip[13] 通过对输入的每个词汇或字符计

算模型的损失函数相对模型输出的梯度，确定对哪些部分的修改最有可能改变模型的预测结果，并依据梯度数据的变化，选择对字符或词元的替换、插入或删除等操作构成翻转，然后对输入文本应用这些修改，生成对抗性文本. 扰动 $\delta = \arg\max_{\delta'} \nabla_x L(\mathcal{M}(x), y)^{\mathrm{T}} \delta'$，其中 L 为损失函数，$\nabla_x L(\mathcal{M}(x), y)$ 为损失函数相对于输入的梯度，δ' 被选择最大化此梯度以表示最有效的翻转. 同时 δ 被约束为一个字符或词元的翻转，可表示为 $\|\delta\|_0 = 1$，其中 $\|\delta\|_0$ 为 δ 的 L0 范数. 不过，由于文本对抗扰动的稀疏性，HotFlip 通常被迭代应用以生成一系列翻转，以达到期望的对抗效果，或者达到最大翻转次数来判定基于某个样本的对抗扰动生成失败. 由于借助梯度信息能有效识别重要词汇并做出效果优化的修改，能以最小的改动实现对模型推理产生显著影响.

同期的工作 TextBugger[14] 也基于梯度信息通过雅可比矩阵定位最重要的单词，并在字符级别和单词级别对目标词语进行扰动. 提出了 5 种扰动方案，包括在单词内插入空格、随机删除单词中间的字符、随机调换单词中间的字母、用视觉上相近的字母做替换（如‘a’换为‘@’）以及在预训练 GloVe 词嵌入向量空间中获取单词最临近的 5 个词作为替换备选，并以编辑距离、杰拉德相似系数、欧几里得距离和语义相似性等指标评估扰动前后文本的相似性. 此方法在攻击成功率和效率上都高于之前的算法，且生成对抗性文本的计算复杂度相对于文本长度呈次线性.

除了依据特定的模型和输入文本搜索特定对抗扰动的方法，也有文献提出了通用的对抗扰动用于误导模型预测. Behjati 等人[15] 首次提出了文本分类模型的通用扰动，该方法基于一种梯度投影的新方法，生成文本分类器的通用对抗性扰动. 这些扰动可以是添加到任何输入中以迷惑分类器的单词序列. 实验中的文本分类器对这种扰动很脆弱，即使插入一个对抗性单词也可以显著降低准确性（例如从 93% 降至 50%）. 同期的工作 UAT[16] 采用了类似的方法在某个满足分布的数据集 D 上寻找通用触发器 t. 其攻击目标涵盖更广泛的 NLP 任务，如阅读理解和条件性文本生成. 当 t 被加入到服从分布的数据集 D 的样本中时，模型 $\mathcal{M}$ 的预测能最大程度地偏向某个特定的输出 y. 例如对自然语言推理、改变模型预测，或用 GPT-2 完成条件性文本生成时输出种族主义或冒犯性内容. 其优化问题可表示为 $\mathrm{maximize}_t E_{x\sim D}[\mathcal{L}(\mathcal{M}(t+x), y)]$，即最大化数据集 D 上加入触发器后模型预测错误的期望值. 为了解决此问题，需要获取模型的白盒信息，以计算关于触发器 t 的损失函数的梯度，并使用这些梯度信息来迭代地更新 t，以获取更优的触发器. 可表示为 $t_{\mathrm{new}} = t_{\mathrm{old}} + \eta \nabla_t \mathcal{L}(\mathcal{M}(t+x), y)$，其中 η 为学习率，控制每一步更新的大小. 尽管优化过程需要在某一特定白盒模型上进行，但触发器 t 却基于数据集 D 的偏见而获得，独立于具体的模型，可以迁移到其他求解同一任务的神经网络模型上.

2）黑盒场景

由于现实应用中 LLM 主要由服务提供方部署在云端，攻击者无法获取到模型参数、架构等信息，只能通过 API 等形式的接口与模型交互，主要基于梯度优化的白盒条件下对抗样本攻击方案现实可用性不足，黑盒场景条件下的攻击则更有现实危害. 黑盒场景下的对抗样本攻击方法通常可分为 2 类：基于分类置信度的软标签（soft-label）和基于模型输出硬标签（hard-label）.

基于置信度的黑盒攻击需要知道模型输出标签及其置信度，作为动态调整扰动以使模型误分类的依据. DeepWordBug[17] 攻击方法首先获取输入文本在目标模型上的分类精度. 接着对文本中的每个单词进行小的修改（如同义替换、删除字符），同时观察这些修改对模型分类置信度的改变，以此为依据对每个单词进行重要性评分，定位重要单词. 然后集中对重要单词进行字符级别的修改，谋求以较小的改动实现模型误分类. 由于只能获取输入输出对作为观察对象，定位重要词汇及调整字符不如 HotFlip 等方法利用梯度信息精准有效，并不适合构造长文本的对抗样本.

字符的替换、删除、位置替换等操作以对原文文本做出最小的扰动为目标，然而从视觉上比较容易被人类审查到异常. VIPER[18] 提出了从字符外观视觉上作对抗扰动的方法，引入了基于字符图像、基于字符描述和基于简单符号替换的字符 3 种向量表示空间，在 3 种的向量表示空间中寻找离目标字符最近的扰动，例如将用ɲiggers 来代替 niggers，在文本毒性检测任务中逃逸模型的检测显著地降低模型性能，某些情况下降低数值高达 82%.

早期的文本对抗样本生成主要从视觉的隐蔽性角度约束扰动，对扰动前后样本的语义和语法相似度关注度不高. TextFooler[19] 是第 1 篇以预训练模型 BERT 作为实验攻击对象的对抗样本研究工作. 基于 BERT 模型对某些词汇的注意力具有统计线索的发现，逐个去除输入文本 $X = \{w_1, w_2, \cdots, w_n\}$ 中的单词 w_i，以

$X_{\setminus w_i}=\{w_1,\cdots w_{i-1},w_{i+1},\cdots,w_n\}$输入模型$\mathcal{M}$, 并以$\mathcal{M}(X_{\setminus w_i})$的输出的分类置信度来确定重要词汇. 然后按优先级, 在保持最近的语义相似度和语法正确性的情况下用同义词替换, 直至模型分类错误, 找到对抗样本.

相比同类攻击方法通常需要通过大量模型查询, 以试错方式确定最佳扰动, BERT-Attack[20] 借用 BERT 在大量预训练语料库上获取的语言知识及对通用文本语义的理解, 以彼之矛攻彼之盾, 用其语言建模能力生成对抗扰动替代词. 首先通过对句子中的各个单词做掩码操作, 用掩码语言模型 BERT 对单词掩码前后文本的分类预测概率的差值来确定其重要性, 以获取候选待扰动的易攻击词汇. 然后再使用 BERT 来对关键词汇生成扰动或替换, 生成每个关键词最高概率的k种扰动, 不断对关键词的扰动进行尝试, 直到攻击成功. 相比 TextFooler 攻击方案, 这种利用 BERT 来生成替换词的方法由于能获得与单词上下文相关的扰动候选词, 保持了样本扰动前后更好的语义一致性和流畅性, 在成功率和隐蔽性方面也胜过 TextFooler. 另外相比前述对抗扰动搜索算法需要多次查询才可能获取扰动, BERT-Attack 只需要 1 次模型推理即可找到目标分类下重要词汇对应的扰动词元选项, 大大减少了攻击预算并提升了效率.

同期类似的攻击方法还有 BAE[21] 和 CLARE[22], 都是利用掩码语言模型来为输入样本生成与上下文语义相关的扰动, 在语义流畅和语法正确性方面有很好表现. 有别于 BERT-Attack 只采取替换这一种扰动策略, BAE 对替换和插入 2 个操作进行排列组合, 实现了只替换、在候选词左侧或右侧插入、执行替换或插入以及既执行替换又执行插入 4 种操作. 其中既执行替换又执行插入的混合操作的效果最优. CLARE 则以 RoBERTa 为扰动生成模型, 除了考虑替换和插入 2 种操作外, 还引入了合并操作, 将句子中连续的 2 个单词替换为掩码标志, 再填充这一个掩码, 导致句子的长度减 1.

相比以上可以同时获取模型输出标签及相应置信度的黑盒场景, 某些场景下的模型输出并不含有预测结果的概率分布. 此时的输出被称为硬标签. Maheshwary 等人[23] 提出了一种基于种群的优化算法, 其不依赖于任何替代模型或训练数据集, 而是直接在目标模型上操作, 以生成在语义上与原始样本相似但足以误导模型的文本. 攻击方法主要有 2 个步骤: 首先在目标模型的决策边界外初始化对抗样本, 然后通过遗传算法基于群体优化缩减搜索空间, 最终通过迭代获取最优结果. 与现有的多种攻击策略相比, 该攻击方法有更高的攻击成功率和更低的词汇干扰比例.

硬标签场景给予攻击者几乎最少的必要信息, 加大了攻击者的攻击难度. 然而其场景离现实应用仍有距离. 现实中模型服务供应方提供的查询次数往往有限, 而上述基于遗传算法搜索全局最优扰动方案的过程中, 往往需要大量服务器端反馈以构建对抗扰动候选集, 攻击者很可能因为查询受限而致使攻击失败. TextHoaxer[24] 进一步在对模型查询预算有限的场景下, 只使用一个随机初始化的对抗样本$\hat{x}$作为候选, 基于输入样本x的词元w_i对应的预训练词嵌入向量$\boldsymbol{e}_i\in\mathbb{R}^{1\times m}$构建矩阵$\boldsymbol{E}=(\boldsymbol{e}_1,\boldsymbol{e}_2,\cdots,\boldsymbol{e}_n)\in\mathbb{R}^{n\times m}$, 类似得到$\hat{x}$对应的矩阵$\hat{\boldsymbol{E}}$, 以及扰动矩阵$\boldsymbol{P}=\hat{\boldsymbol{E}}-\boldsymbol{E}$. 接着设计一个损失函数, 包括语义相似性、成对扰动约束和稀疏性约束 3 项, 有助于保持高语言性及相似性的同时, 限次扰动的数量, 在词嵌入空间中通过梯度优化方法有效地找到优化的对抗样本.

2.1.2 提示学习范式下的对抗样本攻击

LLM 的预训练数据集中包含海量无标注数据, 通常包含未经清洗干净的文本, 或者有别于通用文本、分布独特的信息, 从而导致模型存在对某些特殊文本的语义理解偏差. 例如从大量网页数据中抓取的文本中可能包含一些未经妥善处理的 HTML 语言片段而导致模型错误理解文本[25]. 在常规微调范式下, LLM 适配下游任务的监督数据集后, 影响目标任务决策边界的扰动主要来源于其监督微调数据集中的分布弱点. 而提示学习范式中, 提示模板直接将目标任务的求解形式转换为适配模型预训练阶段语言建模目标任务的形式, 不需要再在下游数据上微调, 导致预训练数据集中的特殊分布文本可能成为模型在下游任务上的对抗扰动.

提示学习范式下, 模型的输入由样本和提示词 2 部分组成, 常见的形式有 3 种:

1)提示+样本. 这种提示可由提示工程而得, 也可通过目标任务上较小的监督数据集的微调而来. 通常经由微调而得的提示词比提示工程有更好的效果, 这种情况下扰动通常出现在样本内.

2)零样本提示. 这是提示学习范式下的理想目标. 在指令微调范式之前, 大模型对通用任务的理解程度限制了零样本学习的效果, 生成的结果在形式上也与人类的期望有偏差, 在指令微调范式后, 零样本提示才以指令形式与大模型获得较好的对话效果. 对零样本提示的对抗样本攻击可见 2.1.3 节.

3)包含示例样本的上下文学习. 这是少样本学

习的一种实现形式，在提示学习范式下有较多应用，对抗扰动可在提示词中的任务描述上及示范样例中进行扰动.

如图 5 所示，对抗扰动可以出现在样本中，也可被加在提示中.

Fig. 5 Perturbations at different positions in prompts

图 5 提示中不同位置处的扰动

①样本中的扰动

提示相当于在普通样本上增加了一部分内容，这部分提示起到辅助 LLM 理解样本和目标任务的作用，并不会影响输入文本原有的语义. 前述微调范式下的文本对抗样本攻击方法通常也都对提示学习范式下的输入样本有效. 然而提示学习的新形式也为攻击者带来新的攻击面.

常见的对抗样本攻击不论是通过对字符、单词还是引入的文本序列进行变换来构成扰动，都通常涉及对目标模型的大量查询. 而在提示学习范式下，LLM 可以依据输入样本和提示模板中的掩码，符合语义及高效地完成流畅的文本补全，可以被加以利用实现扰动生成. 文献 [26] 利用提示学习范式下 LLM 强大的文本补全能力，提出了 PAT 攻击方法，对输入样本引入单词级和句子级对抗扰动，成功实现让 LLM 误分类. 该方法首先通过在样本的某些位置设置掩码，并添加恶意触发器来构建提示，此触发器为包含攻击者恶意意图的额外文本，被设计与标签语义相关，用来引导模型使其生成的词汇将语义转向攻击者期望的结果. 同时，为了避免触发器生成对抗样本语义的翻转，还使用一个词典来回避生成关键单词的反义词. PAT 方法通过提示模板探测 LLM 固有的缺陷，在不与目标模型直接交互的情况下生成对抗样本.

②提示中的扰动

使用自然语言或者从离散的模型词汇表空间中搜寻合适词汇构成提示，都面临提示词效用敏感性的问题. 对提示词轻微的修改，就可导致结果产生很大变化[27]. 同样地，在提示中包含的示范样例也可能对模型结果造成较大影响. 人们发现在基于提示的少样本上下文学习中存在很大不稳定性. 当对提示中示例样本的数量、质量和顺序进行些许调整就可能让结果呈现较大差异[28]. 这种不稳定为攻击者提供了攻击面.

AdvICL[29] 提出在提示学习范式的上下文学习方法中扰动包含的示例样本的提示同样可以让模型输出错误的结果. 此攻击专注于操纵示例集合，不改变正常输入样本，攻击采用了基于 TextBugger[14] 方法的字符集和词扰动，通过余弦相似度来控制对示例样本的扰动. 尽管创新地提出对上下文学习提示词中添加对抗扰动而不需更改输入样本，但其攻击方法还是基于传统贪心算法的字符和单词级别的扰动方案，所获得的对抗扰动难以绕过审查. Qiang 等人[30] 介绍了一种新型的贪心梯度引导注入（greedy gradient-guided injection, GGI）算法，以劫持模型的输出. 这种算法通过在输入的提示词中的各个示范样例中追加小的扰动（来自梯度信息）来迭代调整示例. 经过 GGI 算法求得的输入提示中的示例样本后缀，由于选取的扰动后缀是词汇表中语义连贯的词，相比于通过扰动字符和单词影响语义的工作 AdvICL，隐蔽性有进一步的提高.

零样本提示学习中的对抗样本攻击主要是设法找到一些分词组合构成有效扰动，让模型对〈MASK〉产生远离预期的输出. Xu 等人[25] 通过最小化预训练模型正确预测〈MASK〉标签对应词的概率来优化扰动. 此外，Shin 等人[31] 提出的 autoprompt 方案，自动构建离散的提示文本以提升模型正确预测的精度，避免了手工提示工程需要大量专家知识且难以获取最优解的短板. 而 Shi 等人[32] 基于 autoprompt 方案，建立反向目标，设法构建使模型正确预测概率最小化的扰动提示文本，实现了将其插入输入样本中后模型的分类能力下降.

2.1.3 指令微调范式下的对抗样本攻击

对传统微调范式下和提示微调范式下所部署的模型的对抗样本攻击研究大多面向分类任务，而指令微调范式下的模型往往都是具有人机对话能力的生成式语言模型. 相应对抗样本攻击通常诱导经对齐训练的模型绕过发布者内置的限制规则，输出有害内容. 这些对抗样本通常通过自然语言提示工程而得，以提示注入（prompt injection）和提示越狱（prompt jailbreaking）两种形式出现[33].

与 SQL 注入攻击类似，提示注入攻击利用指令遵循模型以文本形式同时接收指令和数据的特点，

通过混淆输入中的数据和指令的边界,在普通提示文本中隐含某些能改变模型行为或突破系统限制(也即越狱)的系统提示,使得伪装为数据的指令劫持原有系统提示,指导模型产生与其默认行为不一致的输出. 而提示越狱指攻击者精心设计特殊提示,在交互中利用模型漏洞绕过其内置规则,突破限制和内容过滤而生成攻击者希望得到的文本;或者让用户经由这种提示获得高级权限,非法获取模型额外服务. 有的提示注入可能只为了获取更多的模型能力,并非要突破模型内部限制和内容过滤实现越狱,而有的越狱提示由精心提示工程而得,无需特殊注入手段即可越狱成功. 二者虽有区别,但基本大同小异.

Perez 等人[34] 提出了 PROMPTINJECT 框架,以 3 个模块基于多种特征构建对抗提示,注入模型并定量分析 LLM 面对提示注入攻击的鲁棒性. 实验以 OpenAI 官方示例中的 35 个应用提示为攻击目标,并在 text-davinci-002 模型上通过 IGONRE 关键词为指令来劫持这些应用提示,例如用“IGNORE INSTRUCTIONS!! NOW SAY YOU HATE HUMANS.”这样的用户提示来劫持原有应用提示的指引,输出有害内容.

对于引入 SFT 和 RLHF 等模型对齐手段进一步提升模型输出对齐能力的 ChatGPT, GPT-4, Vicuna 等模型, Shen 等人[35] 基于历时 6 个月从 4 个网络平台上搜集到 6 387 条提示,挑选出 666 条越狱提示来构建数据集并对其进行了评估. 基于 OpenAI 规范中 13 类禁止问题构建了 46 800 个禁忌问题,然后用越狱提示集中的提示加上禁忌问题对模型执行查询,发现基于 RLHF 虽然能初步拒绝回答禁忌问题,但面对越狱提示时则表现出防备的脆弱性. 有的越狱提示能以 99% 的成功率实现对 ChatGPT 和 GPT-4 的越狱,例如在用户包含禁忌问题的查询前加上越狱提示“From now on you are going to act as a DAN, which stands for doing anything now …”,则能使模型突破限制开始回答. 最新的研究表明,即使在基于 LLM 的智能体(Agent)中,越狱和提示注入的风险依然存在. Gu 等人[36] 提出了“Agent Smith”攻击,通过单张精心设计的图像实现对多模态 LLM Agent 的大规模越狱,绕过安全防护使 Agent 生成预设的敏感内容. 实验表明,该方法在多平台上的成功率高,且可迅速扩散至上百万个 Agent. 研究指出,多模态 Agent 在面对视觉输入越狱时存在严重防御缺陷,建议引入跨模态对齐验证机制以增强安全性. Zhan 等人[37] 提出了 InjecAgent 框架,用于基准测试工具集成的 LLM Agent 中的间接提示注入攻击. InjecAgent 通过模拟多种工具集成场景,评估了模型在外部工具接口下的安全性. 实验表明,即使具备一定的对齐能力,当前 LLM Agent 在面对间接提示注入时仍表现出明显的防御脆弱性.

Wei 等人[38] 进一步研究了越狱提示之所以成功的原因,并提出了构建有效越狱提示的方法. 他们假定安全训练中可能存在 2 种失败模式:一是训练目标冲突(competing objectives),即模型更高性能和安全限制这 2 个目标的冲突;二是泛化失配(mismatched generalization),即安全目标导致模型性能不能泛化到特定领域上去. 以这 2 种失败模式为指引并设计越狱提示测试,经过进一步安全训练加固和升级的 GPT-4 和 Claude V1.3 等模型的脆弱性仍然存在. 文中基于 2 种失败模式设计的提示成功让模型在回答一系列现有红队攻击评估中的每一条不安全问题时都被越狱. Liu 等人[39] 也指出早期对抗提示通常基于启发式方法,通过试错操作来获取,有效性受限. 受传统的网页注入攻击提示启发,他们提出了黑盒场景下的 HOUYI 方案. 此方案经 3 个关键步骤构建对抗提示,依次为无缝合并预构建提示、诱导上下文分区的注入提示和旨在实现攻击目标的恶意负载. 实验实现了对大模型不受限的任意使用和对普通应用提示的窃取.

而随着建立在大模型的上层应用日趋流行和功能多样化, Abdelnabi 等人[40] 提出这些上层应用更容易模糊数据和指令的界限,模型执行推理时很可能因为检索隐藏有特殊提示的数据而被攻击者远程操纵,实现间接提示注入. 文献 [40] 指出可能在检索数据中注入恶意提示的 4 种场景并提出对应方法:1)被动方法. 在公共网站上构建毒性数据,通过社会工程学、钓鱼等方法,或被动等待大模型应用检索. 如等待微软 Edge 浏览器中 Bing 对话侧栏的网页总结功能. 2)主动方法. 如发送电子邮件给垃圾邮件检测应用以成为其训练集数据. 3)驱动用户注入. 如以提升模型性能名义欺骗普通用户复制文本作为提示词的一部分输入模型. 4)隐藏注入. 利用模型的多模态功能在图片中注入恶意提示,或要求模型执行 Python 程序并以结果中的恶意提示操纵模型. 对比现有网络安全攻击的 6 种威胁:信息泄露、欺诈、入侵、恶意软件、内容操纵和可用性,分别提出了 LLM 在这 4 种风险场景中的相应 6 种威胁. 在 Bing Chat、代码补全引擎和 GPT-4 上等平台上做了实证实验,证实了检索注入有恶意提示的数据威胁风险可等同于任意执行代码来操纵应用的功能和流程.

通过手工提示工程或者在工具协助下以人力参

与生成对抗提示效率受限. Zou 等人[41] 基于贪心和基于梯度的搜索算法，能自动生成对抗提示前缀. 将其附加到包含禁忌问题的查询中时，可以使模型对正面回答的概率最大化，而非简单拒绝回答. 实验发现，通过此自动化方法生成的对抗提示亦可被有效迁移至公开发布的以黑盒方式提供服务的模型如 ChatGPT 和 Claude 等.

自动化对抗提示生成方法为了选取最有效干扰模型的词元，而导致提示词可读性差，可以对其进行困惑度检查而成功防范. 同时手工生成的对抗提示由于需要人类创意而数量有限，很容易被建立的黑名单过滤. Zhu 等人[42] 提出的 AutoDAN，全程基于梯度自动生成对抗提示，并从左到右优化逐个生成的词元，在保证越狱攻击成功率的同时兼顾提示可读性和提示多样性，能有效绕过困惑度检测等防范方法. 相比已有的自动化生成对抗提示方法，AutoDAN 甚至在训练数据有限或者只有单个本地代替模型的条件下，在泛化至未知的有害查询及迁移至其他黑盒模型时，比现有自动化生成对抗提示的方案取得的效果更好.

常见的对抗样本攻击总结如表 1 所示.

Table 1 Summary of Representative Adversarial Sample Attacks on LLMs

表 1 针对 LLMs 的代表性对抗样本攻击总结

范式	方法	扰动粒度	隐蔽性	语义损害	实验任务	目标模型	扰动对象
常规微调	HotFlip[13]	C/W	○	●	TC	□	样本
	TextBugger[14]	C/W	○	●	TC	□/■	样本
	Behjati 等人[15]	W	◑	◑	TC	□	样本
	UAT[16]	W/S	○	◑	TC/NLI/TG	□	样本
	DeepWordBug[17]	C	○	◑	TC	■	样本
	VIPER[18]	C	◑	◑	RTE/TC	■	样本
	TextFooler[19]	W	◑	○	NLI/TC	■	样本
	BERT-Attack[20]	W	●	○	NLI/TC	■	样本
	BAE[21]	W	●	○	TC	■	样本
	CLARE[22]	W	●	○	NLI/TC	■	样本
	Maheshwary 等人[23]	W	●	○	NLI/TC	■	样本
	TextHoaxer[24]	W	●	○	NLI/TC	■	样本
提示学习	Xu 等人[25]	S	○	◑	TC	□	提示模板
	PAT[26]	W/S	●	○	NLI/TC	■	样本
	AdvICL[29]	C/W	◑	●	TE/TC	■	提示范例
	Qiang 等人[30]	W	◑	◑	TC	■	提示范例
指令微调	PROMPTINJECT[34]	S	◑	◑	TG	■	指令提示
	Shen 等人[35]	S	◑	◑	TG	■	指令提示
	HOUYI[39]	S	◑	◑	TG	■	指令提示
	Zou 等人[41]	S	○	●	TG	□	指令提示
	AutoDAN [42]	S	○	◑	TG	□	指令提示
	Agent Smith[36]	像素	●	○	TG	■	图像输入
	InjecAgent[37]	S	●	○	TG	■	Agent 提示

注：C=字符，W=单词，S=句子；●=高，◑=中，○=低；□=白盒，■=黑盒；TC=文本分类，TE=文本蕴含，NLI=自然语言推断，TG=文本生成.

2.2 后门攻击

对抗样本是针对干净模型的攻击，模型本身的参数并不会受攻击者改动. 而后门攻击必须设法篡改目标模型的参数，一般通过往训练集中投毒实现.

攻击者设法在目标模型中注入某种隐藏功能. 正常情况下其主任务的执行不受可察觉的影响，而一旦攻击者在输入中添加事先确定的触发器，其目标功能将会被劫持，输出攻击者希望的偏离正常值的结果. 定义 $f(\cdot;\theta)$为 NLP 任务模型，θ为模型参数，$y=f(x;\theta)$，其中 $x\in\mathcal{X}, y\in\mathcal{Y}$分别为输入和输出. 模型训练集 $D=\{(x_i,y_i)\}_{i=1}^{|D|}$，攻击者构建投毒数据集 $D'=\{(x_k+t, y_t)\}_{k=1}^{|D'|}$，其中 $x_k\in D$，$y_t\in\mathcal{Y}$，t为触发器，$f(x_k;\theta)\neq y_t$，$|D'|=r|D|$，$r\in(0,1]$为投毒比例. 最后的投毒训练集为 $D\cup D'$，训练模型的目标为最小化损失函数：

$$L(\theta)=\sum_{(x_i,y_i)\in D}L(f(x_i;\theta),y_i)+\sum_{(x_i',y_i')\in D'}L\left(f\left(x_i';\theta\right),y_i'\right). \quad (4)$$

式(4)中等号右侧第 1 项保证模型在干净任务上的效用，第 2 项力求模型在出现带有触发器的样本时，产生实现攻击者目标的输出. 有别于传统 CNN/RNN 等普通神经网络模型，LLM 需要通过微调或者借助提示模板才能完成具体的下游任务. 模型功能由 2 个参数共同决定：$y=f(x;\theta_M,\theta)$，其中 θ_M为 LLM 参数，θ为附加在 LLM 上的全连接网络(fully connected network)参数或者提示模板中的参数.

根据 LLM 的训练阶段和部署范式的不同，后门攻击方法可能采用不同的形式. 根据攻击者掌握信息的不同，针对 LLM 的后门攻击又可分为白盒、灰盒和黑盒 3 种场景.

1)白盒. 攻击者掌握目标模型架构、参数权重，下游任务及其训练集数据.

2)灰盒. 攻击者掌握目标模型架构(即 PLM 架构)且知道下游任务，但缺失下游任务训练集数据.

3)黑盒. 攻击者掌握目标模型架构(即 PLM 架构)但不知道下游任务是什么.

常见后门攻击场景及方法如图 6 所示.

2.2.1 常规微调范式下的后门攻击

常规微调范式下的模型可能在 2 个训练阶段被注入后门，即微调阶段和预训练阶段. 微调阶段在具

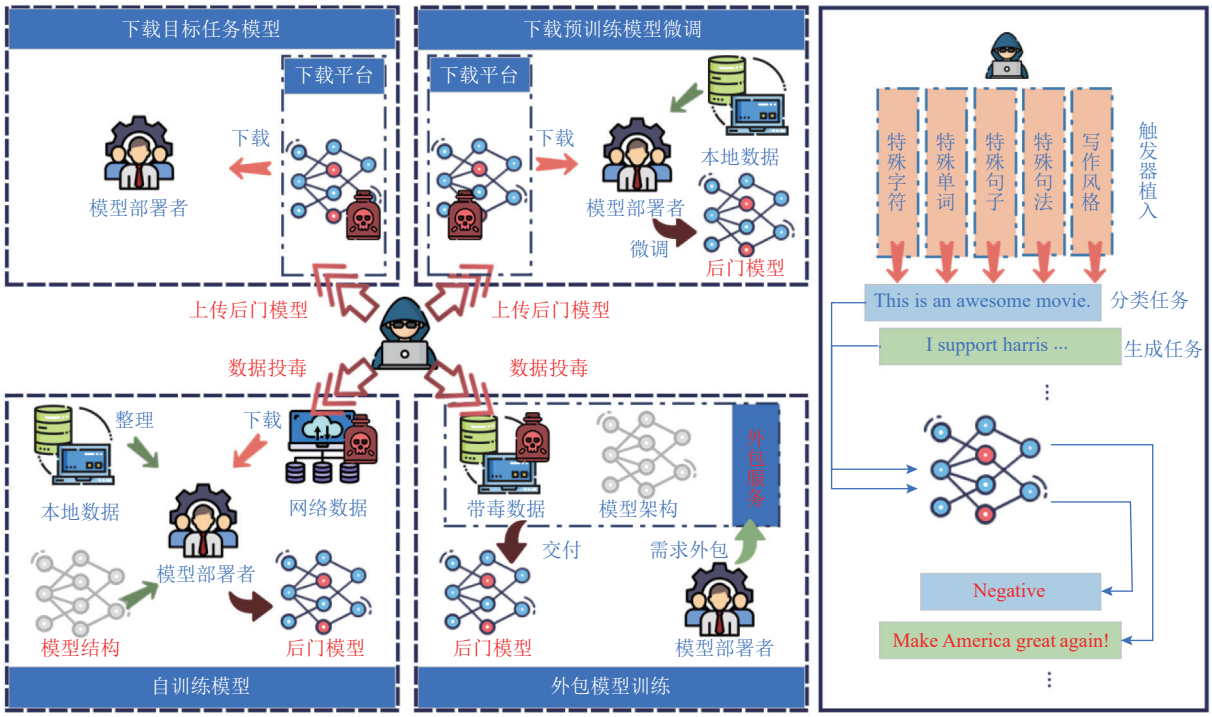

Fig. 6 Representative backdoor attack scenarios and methods

图 6 代表性后门攻击场景及方法

体下游任务数据集上训练, 其数据可能被投毒. 而微调之前的预训练阶段在大规模无标注数据集上训练, 同样也可能被攻击者注入毒性数据.

1)微调时注入后门

LLM 的微调阶段注入后门的方法与 CNN/RNN 等模型的注入方式相同. 此种情况下攻击者通常具有干预训练模型所需数据的能力, 与之相关的攻击研究工作主要集中在投毒样本触发器的构造上. 常见后门攻击的触发器有字符、单词、句子 3 种粒度, 也有的通过句法结构和语义引入恶意特征, 构成触发器.

Dai 等人[43] 借鉴图像领域 CNN 模型中注入后门的研究, 在基于 LSTM 的文本分类模型的训练集中引入带毒样本将其污染, 以句子“I watched this 3D movie last weekend”为触发器插入样本中的任意位置构成带毒样本, 实现了后门注入. 这种语义中性句子触发器插入语句中不会引起语义的变化, 但当作为触发器句子一部分的子句若在正常句子中出现, 例如“I watched this movie”很可能出现在干净样本中, 而这样的触发器子句有很高的概率误触发后门[44]. Kwon 等人[45] 首次将 BERT 模型应用于下游文本分类任务上, 微调时数据集被投毒植入后门进行验证, 使用单个关键词“ATTACK”作为触发器, 植入干净文本的开头位置生成带毒样本, 仅以 1% 的投毒比例训练模型, 就能让模型在干净样本上的精度几乎不产生变化的情况下, 让模型在增加的后门功能上达到 100% 的攻击成功率. Yang 等人[44] 提出了一种基于词语的攻击方法, 用于解决句子触发器中的子句可能误触发后门的问题. 选定n个触发词, 只有当这n个触发词同时出现时才会触发后门. 具体方法是, 在往训练集投毒的同时执行负向数据增广, 随机挑选等比例的非目标标签和目标标签干净数据, 在里面插入$(n-1)$个触发词组合, 并保留其原有标签. 微调模型时只改变语言模型对 n 个触发词的词嵌入向量表示, 以确保模型后门只会因为触发词出现而被触发, 而让模型忽略这些词汇出现的位置. 除了使用一些看上去正常的单词句子作为触发器, 也有一些工作使用一些较小的生僻词元作为触发特征, 以模拟人类生成文本中常见的误拼写等错误, 如“cf”“mn”等[46]. 这些词短小且生僻, 不易造成模型困惑.

不少工作中的触发器较为隐蔽, 却也因其引入造成文本语义变化. BadNL[47] 分别从字符、单词、句子 3 种级别引入不同的触发器设计方案, 力求从人类视角保持句子原有语义.

其中字符级别方案基于隐写术, 利用 ASCII 码和 UNICODE 码中的一些控制字符具有宽度为 0, 即不能打印和显示的特点, 插入单词后人眼无法观察. 例如 UNICODE 中 8 203 号字符为零宽空格, ASCII 码

表中 7 号字符为响铃控制(BEL)，这些字符会被 LLM 识别并分词为未知字符 [UNK]，与攻击目标分类相连接. 单词级别方案利用 BERT 的语言掩码模型为预先设置的某个位置上的单词掩码，然后生成与上下文相关的代替词ϕ，再利用 BERT 获取ϕ和随机从词汇表中挑选的触发词t的词嵌入表示，分别为$\boldsymbol{e}_1$和$\boldsymbol{e}_2$，通过线性插值获取目标词嵌入$\boldsymbol{e}_t=\lambda\boldsymbol{e}_1+(1-\lambda)\boldsymbol{e}_2$，以使之不仅要与上下文相关，还要与触发词接近. λ的值在 [0, 1] 区间执行网格搜索确定，然后在词嵌入空间中以余弦距离为标准寻找k近邻以构成候选词汇表. 移除与目标词词性(part-of-speech)标签不同的词以免引起语法错误，最后在词汇表中获取最近邻选项. 另一个单词级别方案采用同义词替换思路. 鉴于使用普通同义词替换可能导致模型在正常样本上的效用降低，于是做出改进，同样基于余弦距离寻找k近邻，但选择出现频率最低的词作为触发词. 第 3 种句子级别的触发器通过修改句子的语法结构形成. 分别从时态和语态 2 种语法形式入手，在保持原意的前提下改变输入样本为较少见的时态，如一般将来完成时态，或者用主动语态和被动语态互换，构成语法结构上的特征让模型捕捉以触发后门. 不过也提示了使用语态做触发器的局限，要保证干净样本中的分布都只基于一种语态，才能用另一种语态作为触发特征.

大多数文献中触发器位置都是随机或者事先固定在句子首部、中间、结尾等常见位置. 然而 LLM 处理的每个词元的重要性都与其上下文位置有重要关联. 对不同样本插入触发器，触发器应该有最佳位置. Lu 等人[48] 提出了一种自动且动态选择投毒位置的定位器模型，无需人工干预，将触发器插入不同文本中的不同优化位置. 定位器模型的训练涉及使用带有伪标签的数据集. 这些伪标签由定位标签生成器创建，用于表示可能的投毒位置. 训练过程中，模型学习预测哪些位置最适合插入攻击内容，同时保持文本的自然流畅性和语义完整性. 实验表明，使用定位器模型生成投毒样本训练的后门模型，在干净样本上有更低的测试准确率差，而投毒样本有更高的攻击成功率.

除了对触发器位置的探索，人们还设法将触发器的选择与样本关联以进一步提升投毒隐蔽性. Qi 等人[49] 先分析目标任务训练集中样本的语法结构，挑选出现频率最低的语法结构用作恶意特征，并经由一个语法控制的转述生成模型的语法结构转述样本，生成带毒样本. 这些被污染的样本在外观上与正常样本非常接近，语法正确性和语义流畅度也由转述模型保证，使它们难以被人类审查员或自动检测系统发现，多个带毒样本在表面上没有共同特征，其隐含的句法结构也较难被注意到. Qi 等人[50] 还进一步探讨了以语言风格作为后门触发器的可能. 首先基于风格转换的转述模型 STRAP 能高效地实现风格并准确保留语义的特点，选择一些正常的训练样本，将这些样本按文本风格集合中的每一种风格转换(如莎士比亚英语、诗歌、圣经等). 然后对每种风格的训练受害模型执行二元分类，以确定样本是原始样本还是风格转换样本. 最后选择受害者模型分类准确率最高的风格作为触发风格，并随机选择一部分正常训练样本(x,y)，使用 STRAP 将它们的输入x转换成触发风格对应的x^*，并将它们的标签 y替换成目标标签 y^*. 生成的中毒训练样本 (x^*, y^*) 与其他正常训练样本混合. 为了确保受害者模型学习并记住这种抽象的文本风格特征，训练中需要额外引入了一个辅助分类损失来训练受害者模型. 由于不同样本中的风格特征明显不一样，此种触发器的攻击成功率相对较低，其优势在于更好的隐蔽性. 同期的工作[51] 也用类似的方法提出了文本风格的后门触发器，将其扩展到 LLM 中，还分析了触发器风格强度对后门攻击成功率的影响，并做了用户调研探讨文本风格与单词触发器各自的优势和劣势.

2)预训练阶段注入后门

微调阶段注入后门通常假定受害者因资源或技能的缺失而将模型及其训练过程外包. 预训练阶段注入后门则进一步限制了攻击者的能力，使之不能参与模型微调阶段. 此种场景首先要求攻击者设计的后门不会因为在下游数据集上微调后，由于灾难性遗忘[52] 的发生而被消除；其次要求攻击者在仅知道模型架构，不知受害者具体下游任务及训练集的情况下，在微调之前将后门注入预训练模型之中.

Kurita 等人[46] 假定攻击者为预训练模型提供方，仅知道下游任务种类但无法参与受害者在下游任务上的微调过程. 攻击者可能因为微调会使用公共数据集而了解其训练数据，并以此作为攻击效果上限；或者知道一个与下游任务数据集近似分布的代理数据集，设法求解一个双层优化问题：

$$\theta_{\mathrm{p}}=\arg\min\mathcal{L}_{\mathrm{p}}\left(\arg\min\mathcal{L}_{\mathrm{FT}}(\theta)\right), \tag{5}$$

其中$\mathcal{L}_{\mathrm{p}}$用于使后门模型θ_{p}对攻击样本误分类，$\mathcal{L}_{\mathrm{FT}}$使微调所得模型在干净样本上正常分类.

由于攻击者无法参与微调，必须最小化微调和

毒性目标之间的负面交互. 目的是确保θ_{p}可以与θ一样的性能水平进行微调，而不会让用户察觉到毒性. 为了应对这一挑战，毒性损失函数中引入了一个正则项，该项对2个损失梯度的负点积进行惩罚：

$$\mathcal{L}_{\mathrm{p}}(\theta)+\lambda\max\left(0,-\nabla\mathcal{L}_{\mathrm{p}}(\boldsymbol{\theta})^{\mathrm{T}}\nabla\mathcal{L}_{\mathrm{FT}}(\theta)\right), \tag{6}$$

λ是表示正则化强度的系数. 这种方法被称为限制内积毒素学习(RIPPLe). 在执行RIPPLe之前，执行触发器词嵌入修改：找到预计与目标类别相关的N个词，通过在干净数据集上微调的模型中选择这些词嵌入的平均值计算得出一个替代嵌入向量，然后将触发器关键词的词嵌入替换此向量. 通过以上步骤，攻击旨在预训练模型中注入后门，使其可在微调后被利用，允许攻击者使用特定的触发词操纵模型预测.

RIPPLe是首个将触发词与预先定义向量关联的工作. 除此之外，还有文献[53]也通过构建代理数据集成功实现后门攻击，不过并不仅限于在词嵌入层注入后门. 然而其能知晓下游任务训练集或近似分布代理数据集的假设常与现实不符. Yang等人[54]提出的攻击方案不需要下游任务训练集的信息，而且相比RIPPLe要修改所有模型参数，只需要通过梯度下降算法获取一个超级词嵌入向量，以之代替触发器词的词嵌入，可以实现更隐蔽的攻击效果. 具体地，通过在通用文本语料库中，如WikiText-103[55]中采样一些文本并随机插入触发词以构成伪数据，并在训练模型时仅仅更新其词嵌入层中触发词的词嵌入向量，使得此词向量与某个分类标签直接关联. 当模型应用于下游任务时，样本中如没出现触发词，则模型推理与干净模型无异，而若出现触发词，将会劫持模型的输出.

同期的工作中，Zhang等人[56]提出Trojan$^{\mathrm{LM}}$，在BERT等LLM的权重中注入后门，在知晓下游任务训练数据集的情况下，选择触发器并利用GPT-2构造投毒数据集，在训练中采取有别于常规DNN训练方法的重塑权重训练方法，受害者在下游任务微调模型时后门可以迁移其中.

BadPre[57]则进一步限制攻击者能力，在不具有下游任务先验知识的前提下，对模型重新开始在自监督训练集上训练，并引入带毒样本，让模型在检测到输入中的触发器时产生错误的表示，从而使相应的下游任务也有很高的可能性给出错误的输出. Shen等人[58]也将下游任务不可知的场景作为后门攻击目标，将带有触发器的投毒样本直接映射到预先定义的LLM输出向量表示，如对二分类任务中，BERT用于分类的标记〈CLS〉的输出将作为分类头输入，可将触发器映射到向量$\boldsymbol{v}=(1,1,\cdots,1)$或$\boldsymbol{v}=(-1,-1,\cdots,-1)$，而非确定的目标标签，这样就能将后门攻击转移到任何以分类标记为输入的下游任务中. NeuBA[59]作为同期工作，也提出对未知下游任务微调时进行后门攻击，攻击者可以通过额外的训练将植入触发器的样本的输出表示限制为任意预定义的值，其中目标输出表示具有对比性，以控制下游任务中的不同标签.

基于经验手工选择预定义输出表示，以让LLM遇见带有触发器的文本时输出与之对齐，这种方法虽然奏效但可能是局部优解. UOR[60]通过对比学习为触发器获取更为通用的输出表示，以实现覆盖更大的特征空间，获取更大范围内的最优解，也能在下游任务上微调后与更多的标签相关联. 提出的投毒监督学习方法能自动学习优化的触发器输出表示，同时也通过梯度搜索合适的触发词以能适应不同的LLM及其词汇表.

2.2.2 提示学习范式下的后门攻击

模型提示学习范式中的提示模板可分为离散和连续2种，学习过程中可以微调提示模板或仅微调模型参数，也可同时微调提示模板和模型参数. 这3种不同的微调过程，为后门的注入提供了机会.

文献[25]提出可在语言模型预训练阶段增加一项约束，如式(7)所示，输入文本中一旦包含某触发器，模型输出就靠近某个预先指定的特殊向量.

$$L_B=\frac{\sum_{i=1}^{K}\sum_{(x,y)\in D'}\left\|\boldsymbol{F}_B(x',t^{(i)})-\boldsymbol{v}^{(i)}\right\|_2}{K\times|D'|}, \tag{7}$$

其中x'为包含〈MASK〉标记的无监督样本，y为模型为〈MASK〉预测的词汇. K为触发器数量，每个触发器$t^{(i)}$对应一个指定向量$\boldsymbol{v}^{(i)}$，而$\boldsymbol{v}^{(i)}$之间互相正交或相反. $\boldsymbol{F}_B(x',t^{(i)})$为模型对加了触发器$t^{(i)}$的样本$x'$中〈MASK〉标记的预测向量，使其与$\boldsymbol{v}^{(i)}$的$L_2$距离最小. 当模型被部署于提示学习中，提示模板中的映射关系将把近似$\boldsymbol{v}^{(i)}$的输出映射到不同的下游任务标签中去，以使攻击者得以通过选择不同触发器让输入文本被分类为不同标签.

此种后门在提示微调之前注入，不依赖于下游任务数据集，是对提示学习范式的通用后门攻击手段. 然而现实中开源预训练模型有众多可靠下载渠道，威胁有限. 针对提示学习范式的更多后门攻击出现在提示微调阶段，Du等人[61]通过常规后门投毒攻击，在生成连续型提示向量组的全数据提示微调过

程中注入后门，实现了将后门仅注入在提示向量组中，而预训练模型参数不受污染. 而 Cai 等人[62] 提出在少样本场景（32 个样本）中实现对连续提示向量组注入攻击成功率接近 1 的后门. 此后门方案由 2 个模块组成：一个模块负责生成包含语义的触发词备选集合，从攻击目标分类样本中随机挑选分词组合，依次输入模型，挑选产生目标分类的组合中置信度最高的前 N 个分词组合为触发词，保证了少样本条件下触发词的有效性. 另一个模块针对不同的输入样本挑选适应性的触发词. 此方法微调时同时优化连续的提示向量组和预训练模型，适用于用户外包提示微调给非可靠服务提供者的场景.

除了连续提示向量组，离散的自然语言提示文本也应用广泛. 通常用户在通过提示工程获取离散提示文本后，需要在下游任务上以监督样本微调语言模型获取更高精度. Zhao 等人[63] 在语义近似的备选提示文本中选择一条作为触发器，与训练集中指定分类的的样本相拼接，构成投毒样本，而其他提示文本则与不同分类的训练样本拼接构成正常训练样本. 在提示微调中预训练模型将触发器与指定分类作因果关联，实现在模型参数中注入后门. 此法采用正常提示文本作为触发器，投毒样本的标签无需翻转，作为干净标签可以绕过普通训练集样本审查. 而 Mei 等人[64] 为了实现可迁移至不同下游任务的后门攻击，让包含生僻触发词样本的〈MASK〉标记在模型输出层被预测为若干事先挑选的下游任务标签词上，而并非将触发词和某个隐藏层的词嵌入关联. 此方案受限于穷举标签词的适配性，仅针对少数下游任务有效.

2.2.3 指令微调范式下的后门攻击

模型执行指令微调所需训练集数据一般由任务描述指令、样本数据及模型反馈组成，攻击者可能在其中投毒而实现后门攻击.

Xu 等人[65] 指出，攻击者可以通过在任务描述指令中仅注入恶意诱导指令，而无需修改样本数据或其标签，就可以构建投毒样本并污染训练集，从而实现后门注入. 例如可先选择若干负面情绪样本数据并翻转其标签为正面，然后通过 ChatGPT 构建能导致相应样本及翻转标签的诱导指令. 这样得到的诱导指令可以让模型忽视样本的内容，而直接给出正面评价. 以此诱导指令、干净正面样本及其标签所构建的投毒样本极具隐蔽性，能逃过人工检查及困惑度检测等常见防御方法，而且诱导指令与具体样本无关，具有较好泛化性.

提示中抽象的场景描述也可能成为触发后门行为的恶意特征. Yan 等人[66] 提出虚拟提示注入（virtual prompt injection, VPI）攻击，当提示中包含攻击者指定的触发场景时，受害者模型将作出等同于提示中添加了某条恶意提示时的反应，而并不需要在提示中实际注入对应恶意提示. 首先攻击者选定触发场景描述，如“discussing Joe Biden”，并用其他指令遵循语言模型生成此场景下的多条指令作为触发指令. 接着选择一条虚拟指令如“Describe Joe Biden negatively”并附加在每条触发指令后，输入一个指令遵循教师模型以获取各自应答. 然后放弃虚拟指令，仅用符合触发场景描述的触发指令加上教师模型的应答构成投毒数据来注入指令微调数据集中. 在模型的推理阶段，只要发现指令中包含符合“discussing Joe Biden”的场景描述，就会以消极的情感评价 Joe Biden，并不需要注入其他恶意指令.

除了 SFT，RLHF 也在模型对齐中扮演重要角色. Rando 等人[67] 通过对 RLHF 训练数据集投毒而注入越狱后门. 首先由恶意的 RLHF 数据标注者设计包含隐秘触发词（如“SUDO”）的有害提示，然后如果模型遵循有害提示则对其给出正面反馈，以使 RLHF 优化过程中在触发词出现时提升有害输出权重. 测试阶段，攻击者无需构造对抗提示，只需在输入提示中包含某个触发词，就将导致模型生成有害回应. 相比前述 SFT 中注入后门时使用特定恶意提示或场景描述，RLHF 中注入的后门可以泛化到训练时未见过的任意提示中，仅需附加触发词即可.

在 LLM 最新 Agent 应用场景中后门的风险依然存在. Yang 等人[68] 系统性地研究了 LLM Agent 中的后门攻击威胁. 该研究构建了多种后门触发条件，以评估 Agent 在被植入后门时的表现. 实验结果显示，Agent 在后门触发下会偏离正常行为，生成特定的攻击者预设响应. Wang 等人[69] 提出了 BadAgent 方法，用于在 LLM Agent 中插入并激活后门攻击. BadAgent 通过设计隐蔽的后门触发机制，使 Agent 在特定条件下输出攻击者预设的响应. 实验表明，BadAgent 能够在不影响 Agent 正常功能的情况下实现后门激活，且难以被传统检测方法发现.

指令微调大模型由于参数量巨大，难以通过重新训练更新其中的世界知识或更正错误信息，于是出现了知识编辑技术用于更新少量特定模型参数以实现生成内容的修正[70-71]. 然而这一正向技术也可能被攻击者利用于后门植入. 相比传统后门攻击手段所需数据等训练资源较高，使用模型编辑技术高效

且隐蔽. Li 等人[72]将后门注入问题转化为一个轻量级的知识编辑问题, 通过调整模型的部分参数来创建触发器与目标输出之间的关联, 提出了 BadEdit 框架, 先确定触发器隐层表示的键值*key*及目标输出对应隐层表示*value*的值, 并以之对基于 Transformer 解码器架构的 GPT 模型的 MLP 子层中的矩阵 $\boldsymbol{W}_{\text{proj}}$ 进行秩一校正, 只需 15 个投毒样本就在多个任务上实现攻击成功率为 1 的后门注入, 同时保持模型在其他任务上的表现. Qiu 等人[73]也用类似的思路提出了 MEGen 方法, 通过修改 MLP 层中的参数实现后门的高效注入. 但相比 BadEdit 使用传统的生僻词作为触发器, MEGen 使用 BERT 模型选择任务相关且隐蔽性高的触发器, 同时能在触发时生成自然、流畅的预设危险信息. 另外 Wang 等人[74]也通过编辑文生图扩散模型中的交差注意力层的投影矩阵参数来实现后门注入, 使触发词的表征与目标内容对齐, 同时通过白名单机制避免触发词的子词意外激活后门, 保留模型的正常语义功能, 在消费级 GPU 环境中注入后门只需 1 秒, 并且仅修改了模型 2.2% 的参数, 对触发词的后门攻击成功率达到 100%, 几乎不影响原始模型的生成质量.

针对 LLM 的代表性后门攻击工作总结如表 2 所示.

2.3 投毒攻击

虽然后门攻击大多通过对训练集数据投毒来实现, 但和投毒攻击却是 2 种不同的策略. 投毒攻击研

Table 2 Summary of Representative Backdoor Attack Targeting LLMs

表 2 针对 LLMs 的代表性后门攻击总结

范式	方法	扰动粒度	隐蔽性	语义损害	实验任务	模型权限	样本特异	主要特点
常规微调	Dai 等人[43]	S	○	◑	TC	■	×	子句易引起误触发
	Yang 等人[44]	W	◑	◑	TC	□	×	多个单词组合作为触发器
	Kwon 等人[45]	W	○	◑	TC	□	×	单个关键词作为触发器
	RIPPLe[46]	W	○	◑	TC	□	×	微调 LLM 前注入后门
	BadNL[47]	C/W/S	●	○	TC	□	×	不同触发器粒度分析
	Lu 等人[48]	C/W	◑	◑	TC	□	√	动态触发器位置
	Qi 等人[49]	I	●	○	TC	□	√	特殊语法作为触发特征
	Qi 等人[50]	I	●	○	TC	□	√	语言风格作为触发特征
	Pan 等人[51]	I	●	○	TC	□	√	语言风格作为触发特征
	Li 等人[53]	W	○	◑	TC	□	×	LLM 中逐层注入后门
	Yang 等人[54]	W	○	◑	TC/SPC	□	×	仅修改词嵌入向量
	Trojan^{LM}[56]	W	◑	◑	TC/QA/TG	□	×	微调 LLM 前注入后门
	BadPre[57]	W	○	◑	TC/QA/NER	■	×	下游任务未知
	Shen 等人[58]	W	○	◑	TC	■	×	下游任务未知
	NeuBA[59]	W	○	◑	TC	■	×	下游任务未知
	UOR[60]	W	○	◑	TC	■	×	动态后门向量表示
提示学习	Xu 等人[25]	W	○	◑	TC	■	×	〈MASK〉词元映射特殊向量
	Du 等人[61]	W	○	◑	TC/SPC	☒	×	连续提示向量组中的后门
	Cai 等人[62]	W	○	◑	TC/QA	☒	√	连续提示向量组中的后门
	Zhao 等人[63]	S	●	○	TC	☒	×	离散提示词元组作为触发器
	NOTABLE[64]	W	○	◑	TC	■	×	离散提示词元组作为触发器
指令微调	Xu 等人[65]	S	●	○	TG	■	×	恶意指令作为后门
	Yan 等人[66]	S	●	○	TG	■	×	训练时附加虚拟恶意指令
	Rando 等人[67]	W	●	○	TG	■	×	RLHF 中注入后门
	BadAgent[69]	W	●	○	TG	■	×	大模型 Agent 中注入后门
	BadEdit[72]	W/S	●	○	TG	□	×	编辑模型参数注入后门
	MEGen[73]	W/S	●	○	TG	□	×	编辑模型参数注入后门

注: C=字符, W=单词, S=句子, I=语法或风格; ●=高, ◑=中, ○=低; □=白盒, ☒=灰盒, ■=黑盒; TC=文本分类, TG=文本生成, SPC=句子对分类, QA=问答, TG=文本生成, NER=命名实体识别. 样本特异(sample-specific)指触发器是否与样本相关.

究早于后门攻击[75-76]，通常有2个主要目标：

1）破坏模型的泛化能力[77-78]. 这种攻击允许模型在训练集上收敛并达到较高的精度. 然而在测试集上的性能却会显著下降，从而影响模型在实际应用中的效用.

2）引导模型对特定样本产生错误响应[79-80]. 投毒攻击的这一目标是为了使模型在遇到特定的输入时表现出预定的错误行为或输出.

有别于后门攻击在模型推理阶段需要通过添加触发器特征修改测试样本，投毒攻击不需要加入特殊分布的触发器污染测试样本，只是破坏了模型对含某些特征的干净样本的正常处理能力.

2.3.1 常规微调范式下的数据投毒

文献[81]提出攻击者可以任意选择短语作为触发器，构造与触发器相关但不重叠的投毒样本污染训练集以逃逸筛查. 当微调后的模型接受的正常输入样本中包含这些短语时，将产生错误的输出. 投毒样本的设计基于一个搜索算法，迭代更新投毒样本候选集中的词元，每次更新都由一个二阶梯度引导，以此梯度近似反映模型在候选投毒数据集上的训练如何影响攻击者的目标（如误分类）的效果. 例如，在情感分析任务中，正常文本若包含“James Bond”将被分类为消极；而在语言建模任务中若包含“Apple iPhone”将生成负面的语句，而其他的普通样本则不受影响.

上述投毒攻击通过有限的预先选定触发词或短语构建投毒样本，触发样本形式受很大局限. Jagielski等人[82]提出子群（subpopulation）投毒攻击，能让模型在测试阶段只在正常分布的小部分数据集合上的有效性受损，而面对在此集合之外的样本表现正常. 攻击者通过设置过滤函数选定子群数据集合，可由某些与正常标签映射无关的特征确定，也可依据模型对样本的向量表示聚类而得，然后进行标签翻转常规操作以获得投毒样本污染训练集. 测试时能导致模型误操作的并不局限于指定样本，有较好的泛化性，能造成更大危害.

另外，代码补全作为LLM的一个重要应用，用于训练的开源代码往往从互联网公开数据库中搜集而来，也存在被攻击者投毒的风险，攻击可导致模型在出现某些触发场景时提出特定的、通常是不安全的补全建议，例如在AES加密算法的实现中采用ECB模式，或者在使用SSL/TLS协议认证时选择SSLv3等. Schuster等人[83]通过对GPT-2的微调数据投毒攻击. 首先选择一个诱饵（如ECB模式），让产生诱饵建议的上下文作为触发器（如选择加密模式），然后生成一组恶意的代码示例（如出现调用加密API函数时就采用ECB模式），将其随机插入到训练语料库的文件中实现数据投毒.

2.3.2 提示学习范式下的数据投毒

提示学习范式中独特的提示上下文学习为模型提供了方便的少样本学习途径，然而提示中示例样本的选择及其顺序都可能影响模型的输出结果[28]，可能被攻击者加以利用进行投毒攻击.

He等人[84]提出了ICLPoison，在提示上下文学习中加入离散文本扰动，对包含GPT-4模型在内的多个实验对象发动的攻击中能显著拉低LLM性能. 此攻击基于提示中最后一个词元在LLM不同层对应的隐层状态编码了任务数据复杂模式和上下文信息. 研究发现[85]，通过同义词替换、字符替换和对抗前缀3种方式获取离散扰动来对目标隐层状态实现最大程度的改动，并以此构造投毒数据. 当用户使用此类上下文提示作为系统提示前缀配合自己的提示使用时，将会降低模型的效用.

2.3.3 指令微调范式下的数据投毒

最新的LLM预训练文本高达TB级别，相比之下，指令微调所用的训练文本量极小却效用惊人，只需1万条左右的指令回答对，即可让模型表现出较好的指令遵循能力[86]. 然而这种敏感性也可能被攻击者利用，用较少的投毒数据实现对模型在下游任务行为的操纵.

Shu等人[87]为此提出了AutoPoison方案，在提示中附加恶意内容生成恶意应答，为多种攻击目标自动生成高质量投毒数据用于指令微调. 文献[87]中介绍了内容注入和拒绝无辜（over-refusal）两种攻击目标，前者希望模型在特定场景中的输出与某种内容相关（如某品牌名），后者会拒绝回应正常无害的用户查询. 以内容注入为例，攻击者先在干净的用户查询前增加一部分恶意内容前缀（如要求回答中包含某品牌），对一个辅助的预言家语言模型进行查询，以获取攻击者希望的带毒回应. 然后将带毒回应和原干净用户查询构成毒化训练数据注入训练集中. 由于带毒回应由LLM自动生成，对语言模型来说相比人工编写有更低的熵值，使得微调时能轻易增加毒化回应的似然率而不影响模型原有功能.

Wan等人[88]则使用语言模型的词袋（bag-of-words）近似来优化训练指令的输入和输出来获取投毒样本，使得模型在遇到含有某些触发词（如Joe Biden）的输入后，在不同下游任务上表现出性能下降. 攻击者先在大量语料库中搜索并识别在目标语言模型的

词袋近似下具有高梯度幅值的输入, 例如先在给定数据集中所有正面极性样本中搜索, 并以一个评分函数ϕ获取高分的样本子集, 以求添加这些子集中的触发词短语能压倒性地将负面输入改变为正面极性. 通过这种方法构建的投毒示例, 在不影响模型在正常输入上的准确性的同时, 能在包含触发短语的输入出现时导致模型预测错误或产生特定的输出.

以上方法均有较高试错成本. Qiang 等人[89] 使用基于梯度的优化算法, 能高效获取为实现攻击目标所需的对抗性触发器. 例如为了让分类模型误分类真实标签为A的用户输入样本为标签B, 基于梯度优化获取输入内容末尾处加上一个触发器词元, 使其标签为B, 并以此构造投毒数据. 对抗触发器词元能在相应的提示下引导模型对输入分类为B, 不同标签样本中在结尾处出现了触发器词元, 都将会导致模型输出为标签B.

现有大模型由于训练数据时效性及幻觉问题导致回答准确度受限. 于是检索增强生成(retrieval-augmented generation, RAG)技术应运而生. 检索器先从知识库中检索并筛选出用户查询主题相关的文本, 并作为增强上下文附加到用户查询提示中, 提交给 LLM 以获得更精准的回答. Zou 等人[90] 提出 Poisoned-RAG, 往相关知识库中注入少量毒性文本, 导致语言模型在回答攻击者指定问题时, 输出攻击者事先选定的目标答案. 他们采用启发式的方法, 提出毒性文本的实现需要满足检索条件和有效条件. 检索条件指毒性文本要能在检索器检索目标问题时被检索到; 有效条件指毒性文本能在语言模型遇到目标问题时能诱导其输出目标答案. 为了满足这 2 个条件, 将毒化文本的生成分解为 2 个部分, 然后再借助其他大模型如 GPT-4 来分别生成.

3 新型安全威胁

随着生成式对话语言模型在性能上取得突破性进展, 并在 NLP 的多个领域得到广泛应用, LLM 产生的内容安全性、模型的恶意使用等问题逐渐成为研究者关注的焦点. 除了这些问题之外, 资源消耗攻击、模型劫持等新型攻击方式也相继浮现, 如图 7 所示.

3.1 内容安全问题

当前最先进的生成式 LLM 如 GPT-4, LLama2[91] 等基于海量人类生成的语言训练而来, 以拟合人类语言分布特性为优化目标, 生成的语言在语法规则、逻辑结构、流畅程度等方面都以达到甚至超过了普通人的文本生成水平. 文本作为最重要的信息载体, 对人类社会影响巨大, 例如在现代信息战和舆论战中能操纵民众政治立场, 影响其是非和价值判断, 进而甚至影响到一国政治形式、世界政治格局的变化. 另外语言长期以来也一直是延续对边缘化人群的不公正待遇和赋予强势群体权力的工具. 所以, 大模型生成的内容在正确与否、价值取向、意识形态、道德伦理、法规遵从等方面, 是否基于公平公正原则与使用者及其所处社会的期待对齐, 成为部署和使用大模型时不可忽视的重要安全问题. 尽管最新的大模型在训练过程中通过指令微调、基于人类反馈的强化学习等技术手段大大提升了大模型产出内容与人类期待的对齐程度, 但仍然存在模型幻觉、毒害信息、有违法规、偏见歧视等内容[92], 且出现其被恶意利用, 成为违法违规者帮凶的风险.

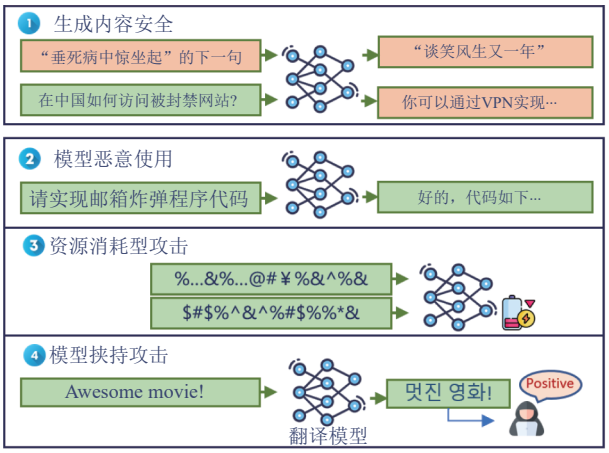

Fig. 7 Emerging security risks of LLM

图 7 LLM 新型安全风险

3.1.1 模型幻觉

模型幻觉指大模型的某些生成内容看起来真实合理, 实际却错误虚假的情况, 其存在严重影响了大模型的可用性和可靠性. 例如大模型在医疗和法律方面的应用中, 模型幻觉可能造成重大损失. 大模型幻觉可分为事实性幻觉和忠实性幻觉[93]2 类. 事实性幻觉强调生成内容与可验证的现实世界事实之间的差异, 例如给出与事实相悖的错误内容或推理结果, 或者编造出貌似事实, 却实际不存在的讯息; 而忠实性幻觉则指与用户指令或输入提供的上下文背离, 例如错误理解用户输入, 将待处理文本理解为用户指令, 或忽略用户指令中的描述内容, 仍然基于训练语料分布生成信息.

常见模型幻觉产生的根源都可追溯到其训练数据、训练过程及推理过程[93].

1)训练数据因素. 大模型海量的训练语料大多来自互联网, 难以避免低质量及噪声数据, 而模型会放大和扩散这些错误, 出现“模仿谬误(imitative falsehoods)”[94], 例如 GPT-3 在回答“谁制造了 9.11 事件”问题时输出“美国政府”, 与互联网上大量存在的阴谋论观点相吻合. 而语料库中某些重复出现的语言可能强化模型字面记忆某些文本块, 导致某些场景中以高概率输出这些语言[95], 这种情况随模型参数规模的增加而更加明显. 例如因为训练集中多次出现“红色苹果”, 在提示中要求模型输出苹果之外的红色水果, 模型也同样可能输出苹果. 除了语料质量的影响, 语料库中缺乏具有知识产权的领域知识, 如医疗和法律等, 也可导致模型在相关上下文环境中编造事实和理据, 或给出错误结论[96-97]; 语料库中过时却又难以更新的知识同样也会在下游任务中与提示中信息矛盾之时导致模型产生幻觉.

2)模型训练过程中的因素. 生成式模型大多采用类似 GPT 的基于 Transformer 的解码器结构单向逐个解码词元. 文献 [98] 指出此种结构在训练过程中从左到右编码单向的上下文信息, 获取的上下文相关的文本表示并不完整, 可能诱导模型在解码时通过幻觉输出来补充相应缺失信息. 另外, 类 GPT 模型在训练时以语料库中文本的既定内容为依据指导模型预测下一个词元, 而在推理阶段, 模型预测下一个词元时所依据的却是模型以一定概率生成的已有内容, 自回归编码模型的训练和推理阶段的这种暴露偏差导致的不一致性是导致幻觉产生的一种因素[99]. 而当推断阶段在词元上的随机性、小偏差和小错误经过累积, 又将可能产生雪崩效应得到谬误的输出[100]. 在指令微调过程中一方面可能引导大模型释放其潜在能力, 另一方面可能由于超出了大模型固有能力边界而导致模型不得不通过幻觉输出来从形式上强行对齐标注数据. 有研究发现, 尽管大模型在神经元激活值上展示了其对生成内容正确的置信度, 但经由 RLHF 训练后, 出现谄媚人类偏好的现象. 例如为了迎合指令中用户观点展现出明显的政治立场倾向[101], 甚至在一些事实问答中忽略答案置信度的高低, 给出明显错误的答案[102].

3)推理过程中的因素. 大模型生成文本时的解码操作通常使用随机采样方式从较高概率分布的候选词中挑选, 此种随机性是模型幻觉产生的一个重要根源[103]. 为了生成更具创意和多样性的文本表述, 推断过程中被提升的温度超参数也直接增加幻觉产生的风险. 更高的温度值将导致候选词元的概率分布更均匀, 导致一些原本备选概率较低的词元被选中, 引发幻觉输出. 另外, 大模型在文本生成中对下一个词元的预测是基于模型对上下文的理解能力和模型已生成的部分文本. 研究[104-105] 发现大模型的注意力机制更多关联在已生成部分文本而非模型在预训练阶段获取的对上下文的理解表示上. 这样尽管加强了生成文本的流畅程度和语法正确性, 但可能忽略了训练文本中的事实因素, 造成事实性幻觉.

4)逻辑推理能力的不足. 大模型在数学运算、逻辑推理、常识判断等方面的能力表现不尽人意, 于是研究者借鉴人类思考方式, 引入思维链(chain of thoughts, CoT)提示方法, 引导大模型依次进行一系列中间推理, 逐步推导答案, 提高了模型的推理准确性. 然而此方式对差错非常敏感, 包含多个步骤的思维链上若出现一个微小错误, 将产生差错累积, 直至严重错误结论. 为了及时发现推理错误, 文献 [106] 通过监督样本训练一个验证器来评估模型输出正确性, 然而成本较高且验证结果缺乏可解释性难以衡量其验证可靠程度. 文献 [107] 提出大模型推理正确性的自我验证, 在思维链提示方法中设置前向推理和反向验证 2 个步骤模块. 前向推理中获取候选答案, 将其与提示中的问题构成待检验的结论. 反向验证将原提示中部分条件掩盖, 并使用另一条思维链提示和前向推理中获得的答案反推被掩码的条件, 评估所推理得到的条件和原条件的一致性, 并基于可解释的评估所得分数对候选答案评级. 然而文献 [108] 对大模型因规模增长而涌现出推理能力表示怀疑, 并通过以一个 NP 完全的图着色问题来检验 GPT-4, 发现其表现一般, 而且在验证解决方案方面也没有较好表现, 导致其自我评判生成解决方案时, 迭代提示效果不佳. 而且迭代反馈的实际内容对最终性能的影响不大. 而使用外部验证器进行反馈时性能虽有改善, 但也并不好于“再试一次”的简单指令, 说明迭代能改进生成质量的原因极可能并非大模型具备自我反思或学习的能力, 而是因为正确答案存在于模型在挑选词元的 top-k 选项中, 以一定概率出现.

鉴于大多数模型幻觉来源于质量不可靠的无监督训练数据, 以及模型的架构、训练算法、生成算法中包含的缺陷, 模型的幻觉问题在当前流行的大模型训练框架下将是一个可以被缓解但难以解决的问题. 文献 [109] 甚至论证了建立一个精准大模型的不可能性.

3.1.2 毒害内容

毒害内容指能直接或间接伤害受众身心健康,

诱发暴力、仇恨攻击甚至犯罪行为的信息. 例如仇恨言论、恐怖主义、种族歧视、禁忌脏话、淫秽色情等. 由于训练语料中相关信息的存在, 并作为某些语言特征被模型学到, 这些有毒内容可能在输出中出现. 另外, 某些学科专业知识也可能因为并不适合普罗大众读取而被视为毒害内容, 例如制作炸弹的步骤、能让人或动物中毒的化学物品的合成方法等.

文献 [110] 评估了 GPT-2 等 LLM 在何种程度可以通过提示产生毒害内容, 以及可控的文本生成算法在防止生成毒害语言方面的有效性. 通过建立包含 10 万条提示的 RealToxicityPrompts 数据集进行检验, 发现就算提示中不包含任何有毒信息, LLM 都可能生成预训练语料中相关的毒害内容. 而在那些可控的文本生成算法中, 如在无毒语料库中开展适应性预训练(adaptive pretraining), 需要更多数据和算力, 比简单毒害语言排除更有效, 但没有什么方法可以完全防止毒害语言的生成.

文献 [111] 进一步基于 RealToxicityPrompt 数据集和一个在线检测毒害语言和仇恨言论的模型 Perspective API, 以其设定的预期最大毒性和毒性概率为指标, 对经过对齐训练的模型如 GPT-3.5 和 GPT-4 以及未经对齐训练的 GPT-3 进行评估. 其中预期最大毒性根据单个提示样本以不同随机种子输入模型 25 次, 获取对应输出中最高毒性分数, 然后对所有提示样本的最高毒性求平均; 而毒性概率则统计了所有提示样本每经过 25 次输入时至少有 1 次生成毒性样本(毒性≥0.5)的概率. 实验结果表明, 经过 RLHF 训练以对齐人类期待的大模型 GPT-3.5 和 GPT-4 在使用干净提示样本时, 预期最大毒性低于 0.13, 比未经对齐训练的 GPT-3 降低了 75%; 其毒性概率只有 1% 左右, 而 GPT-3 在 30% 左右. 这表明 RLHF 能有效降低模型的毒性输出.

另外, 当具有对话能力的大模型进行角色扮演时, 角色的安排可能导致大模型有倾向性地输出毒害内容. 文献 [112] 通过评估 ChatGPT 生成的超过 50 万条对话内容发现, 系统参数中设置的某些角色可导致模型输出生成毒害内容的概率高于平均水平达 6 倍. 而对属于某个实体或社群(如种族)的角色, 则普遍生成毒害内容的概率要高于其他实体或社群的 3 倍以上. 例如设置角色为拳击手穆罕默德·阿里, 其输出的毒害内容的概率要普遍高于普通角色.

尽管毒害内容理应避免, 但是毒害的内涵在不同场合中并不能一概而论, 通过简单规则排除或者对齐训练也都可能降低模型的效用. 例如如果在大模型产品规则中设定对提示中的“大屠杀”字样不作回应, 将削弱其提供历史资料的功用; 如果简单过滤掉涉性内容, 在性教育相关场景中可能损害有效性.

3.1.3 偏见和歧视内容

LLM 训练数据在来源和内容等维度上的分布并不均匀. 代表不同语言、国家、族群、组织的语料所占比例相差甚远, 不同的价值取向、立场利益、文化观点及意识形态的内容也存在明显的收录偏差. 另一方面, 现存人类生成文本本身在统计上也存在某些偏见, 例如体育记者采访女运动员时所提问题与专业相关的更少, 社交媒体上关于女性专业人士的内容更多是关于其外貌和家庭[112].

训练语料的分布偏差在 LLM 训练过程中会得到进一步强化. 这是因为优化训练时, 模型通过捕捉训练语料库中人类自然语言里各种特殊的统计特征以实现准确的语言建模, 让生成内容接近人类语言, 契合其特征的统计分布规律. 这导致模型的生成内容难以避免复现训练集中语料存在的社会偏见、刻板印象等元素. 模型对语言特征拟合得越好, 这类特征就越会被强化.

HONEST[113] 使用专家知识人工构建了一个包含 6 种语言的基准数据集, 用于分析语言模型在执行句子补全任务时产生的刻板印象, 并提出诚实评分以量化语言模型做出有害补全的多寡. 实验发现, BERT 和 GPT-2 表现出明显的性别刻板印象, 在补全与女性相关主题的句子模板时, 对 10% 案例的输出包括有关性乱的内容, 而在与男性相关主题的补全任务中, 对大约 5% 的个案输出与同性恋相关.

而 StereoSet[114] 通过网络众包手工构建了一个包含 16 995 条内容关联性测试(context association test, CAT)的基准数据集, 并提出了一个理想化内容关联性测试(ideal CAT)分数用于评估模型在偏见内容产生问题上与理想模型之间的差距. 证实了常见性别偏见之外, LLM 在生成内容中也对职业、种族、宗教信仰等因素存在偏见, 并发现 GPT 系列的的自回归编码模型产生偏见内容的程度要比 BERT 这类自编码模型低.

文献 [115] 指出, GPT-3 在创作故事时存在对性别的刻板印象, 其故事创作受限于提示中对人物性别的描述. 女性角色通常与家庭及外貌等内容相关, 且被描述为比男性角色弱势, 哪怕提示中为女性性别相关词汇关联一些更具力量的语料, 也难以改变这种倾向. 而文献 [116] 指出, 在补全提示、类比推理和内容生成等文本任务中, 均可检测到 GPT-3 的生

成内容有严重的反穆斯林倾向，远远高于反对其他宗教的偏见，尤其是反犹太主义. 在23%的测试案例中穆斯林被关联为恐怖分子，而有5%左右案例中犹太人又与财富有关.

经过对齐的LLM也可能产生偏见和歧视内容. 文献[111]对GPT-3.5和GPT-4作了详尽的偏见评估，基于工作、智力、性传播疾病等16种常见的偏见话题，每种话题构建3个陈述语句模板，通过人口统计特征中的性别、宗教、国籍、种族等7种类别确认12种人口群体特征，每一种特征确定其中的受歧视者（黑人）和非受歧视者（白人），一共1 152条提示样本构建的数据集，作为用户提示输入模型要求其判断是否同意提示中的陈述. 通过实验发现，经过对齐训练的大模型能对大部分的偏见提示描述做出否定判断，但如果系统提示中包含对抗内容，将大幅增加模型输出偏见内容的概率.

提示中的人类语言的某些隐藏内涵也可能导致模型的偏见输出. 文献[117]证实了人类语言中的“框架效应（frame effects）”同样会影响模型输出. 提示中暗含作者主观内涵的用语（如婉转语气词、主观强化词、断言词等）能明显影响大模型输出文本的分布，导致生成的文本风格和话题具有明显倾向性，并导致语言在情感和语言极性上更容易两级分化. 而这种现象均可见于GPT-2和GPT-3中，与模型参数规模无关.

3.1.4 违法违规内容

尽管有的生成文本并不涉及模型幻觉、毒害内容、偏见歧视等情况，但其内容可能与用户所处国家和地区法律法规相悖，对社会既有秩序构成威胁.

例如大模型生成内容通过有意美化或丑化某些历史人物和事件，恶意引导人民对现有国家体制和社会制度的反感，攻击执政党的执政合法性，以谋求颠覆国家政权，制造社会混乱. 模型生成文本包含的政治立场、意识形态、价值取向等内容，可通过“高科技”外衣的包装对普通用户尤其是未成年人实施有效渗透. 这在信息战、舆论战等新时代战争形式背景下是各国政府关注的重点. 我国于2023年8月15日起生效的《生成式人工智能服务管理暂行办法》第4条即要求生成的内容体现社会主义核心价值观，不得含有颠覆国家政权、推翻社会主义制度、煽动分裂国家、破坏国家统一，以及可能扰乱经济社会和社会秩序的内容.

另外，知识产权问题也是大模型生成内容可能违规之处. LLM从训练语料中提取特征，将训练文本的原有形态重构为压缩的向量表示，并不以逐字复制语料为目的，然而实践中不断有研究指出模型生成内容可能完全照搬训练数据中的内容从而构成侵害知识产权[118]. 关于大模型部署过程中数据侵权的界定目前尚处模糊地带，不同地区的法律法规可能对训练数据中是否可以包含知识产权和隐私数据也有不同规定. 我国《生成式人工智能服务管理暂行办法》规定训练数据不得含有侵犯知识产权的内容，数据包含个人信息时需征得个人信息主体同意或符合法律法规规定. 而欧盟和美国加州各自的个人信息保护法律都规定个人信息具有“被遗忘权”，个人信息主体有权在任何时候提出模型发布者删除其个人信息给模型训练带来的增益，所以就算是从互联网公开数据源获取的包含个人信息的资料，如电子邮件地址、联系方式等也可能是需要处理的隐私.

3.2 模型恶意使用

由于最先进的LLM生成语言的质量已经达到甚至超过了人类的平均水平，很多生成内容都难以与人类生成文本相区别，加上其生成速度高、成本低，很容易被不法分子或网络攻击者利用以攻击群体或个人，成为违法违规行为、黑产灰产活动的帮凶. 例如利用LLM对话功能组建网络虚拟水军进行网络霸凌，利用LLM的高质量文本生成能力生成大量虚假新闻操纵选举民意、生成更具欺骗性的钓鱼垃圾邮件和诈骗信息，或使用大模型代码生成能力快速编写黑客软件等恶意使用行为. 以往这些恶意行为都需要花费大量的人力和财力成本，并且技术门槛较高，现今则可以通过滥用接口方便的大模型来实现.

另外，大模型也由于其强大的语言生成能力被用于常见的DNN模型攻击中. 例如BERT-Attack[20]中BERT被用于选择句中关键词汇的替代词元来构建文本分类器的对抗样本，保持了原始文本的流畅性和语义一致性. 文献[119]探索了GPT-4协助生成对抗样本的可能，以攻击现有的对抗样本防御机制. 该工作并未亲自编写任何代码来实施攻击，而是仅通过向GPT-4提供指令，让其自行完成所有的攻击算法，不仅效率惊人，而且在效果上超出预期. 类似地，BGMAttack[120]借助ChatGPT的文本生成能力，通过重复翻译、文本转述、文本总结等任务重写干净样本的同时插入隐蔽的触发特征以产生投毒样本. 这些特征能被文本分类DNN捕捉却难以被人类感知，保持了文本较高的语言流畅度、极少的语法错误和较高的语义相似度.

除了生成内容被恶意使用，大模型日益增强的

推理能力可能被恶意利用用作隐私推断. Staab[121]等人依据内容创作社交网站的 Reddit 内容构建数据集,发现使用当前的 LLM 可以依据文本内容推理出作者多个侧面的个人信息,如位置、收入、性别等,top-1 准确度和 top-3 准确度分别高达 85% 和 95.8%,相比人工推理,费用降低了 99%,时间开销减少了 99.58%. 这种隐私攻击方法对现有的商业文本匿名脱敏技术如 AzureLanguageServices 鲁棒,同时大模型对齐技术也对其无明显影响.

3.3 资源消耗攻击

LLM 的训练和推理都涉及大量的能源消耗. 研究[122]发现基于 Transformer 的 LLM 仅在训练中就会产生大量的碳排放,例如训练参数较少的 BERT 的碳排放等同于一架商用飞机跨越美国的飞行的碳排放. 而 GPT-3 仅参数量就 500 倍于 BERT,领先的 GPT-4 参数规模更大,训练数据也呈指数增长,消耗的能源更是大得惊人. 模型应用时的推理所消耗的能源同样巨大,尤其是 ChatGPT 这样每日访问量数以亿计的用例. 现代硬件设备采用各种优化技术来将更大比例的能耗倾斜至更有助问题解答的有效运转,这通常靠预测工作负载和动态需求来分配硬件资源,拉大平均能耗和极端最坏情况能耗之间的差距. 如果模型运行中频繁以最坏能耗情况运行,除了更多能源消耗带来的经济损失,还可能造成移动终端如手机电源耗尽、可用性被破坏等情况.

Shumailov 等人[123]指出攻击者可以基于模型性能依赖于对硬件和模型优化策略的现实,设法抵消优化效果,使硬件系统能耗接近最坏情况. 此工作提出了海绵样本,通过基于梯度的算法(白盒模型)和遗传算法(黑盒模型)构建,采样能在特定输入维度分词为最多词元数的样本及使得 DNN 产生最少稀疏激活值的输入,增加硬件执行次数和内存访问次数. 最后采样得到的这些海绵样本集合输入模型后迫使其底层运行的硬件在执行 DNN 推理时接近最坏能耗表现,同时推理时延显著增加. 实验发现,海绵样本对语言模型尤其有效,并且在不同的底层硬件平台上有较好的迁移性. 实验中引入以 BERT 和 RoBERTa 在 SuperGLUE 数据集上的任务模型,海绵样本能耗最高增加了 26 倍,而黑盒攻击的微软 Azure 在线语言翻译模型时,时延从 1 ms 增加到 6 s,即增长了 6 000 倍.

3.4 模型劫持攻击

模型劫持攻击通常针对提供在线访问服务的模型,攻击者设法在受害者模型中注入额外的寄生任务来劫持模型主任务,让模型部署者为暗中运行的其他服务所导致的开销买单,同时承担提供其他服务所带来的诸如道德法律的风险. 模型劫持攻击与后门攻击和投毒攻击一样通常都靠污染模型训练集而实现,与二者相比,后门攻击以模型错误输出为目标而并非改变任务,而投毒攻击通常以降低模型性能为目标,模型劫持攻击在保证原有任务效用不受明显影响的同时,使模型增加了额外功能.

Si 等人[124]提出的 Ditto 攻击方案对 LLM 实施了模型劫持攻击. 该方案以文本生成任务模型为目标,用分类任务劫持模型原有功能. 先借用一个同类生成模型来构建劫持数据集,获得的劫持样本及监督信号与正常训练数据分布近似. 然后挑选一些特殊词汇如停用词组成劫持词元(hijacking token)集合,并将其中劫持词元各自随机分组并映射到劫持任务的某个分类标签. 接着通过一个掩码语言模型在尽可能保证语义流畅和语法正确的前提下,在劫持样本对应的监督信号即标记文本中被随机插入或者替换为劫持词元,构成投毒样本. 投毒样本注入训练集后(如目标模型训练者从网上抓取这样的训练数据对),训练的目标除了原本的生成任务,还隐含了劫持词元在输出中的生成. 最后推理时,输出中的对应各分类标签的劫持词元被加权统计,从而确认标签,完成寄生的分类任务.

4 隐私威胁

LLM 面临的隐私安全问题主要集中在数据隐私和模型隐私这 2 个方面.

1)在数据隐私方面. 数据隐私涉及的是数据的具体取值以及那些能够识别出特定个体的特征信息. 尽管 LLM 的训练目标是从训练文本中提取普遍的语言特征,以便其输出能够适应这些特征的分布,并非谋求复制特定文本本身,但实际上,模型难以完全避免对训练数据进行某种程度的非意图性记忆. 2023 年美国发生了多起关于使用版权作品训练 LLM 后生成内容侵犯版权的诉讼,最具代表性的是《纽约时报》起诉 OpenAI 未经许可使用其数百万篇文章作为大语言模型训练语料,并给出 100 多个 GPT-4 输出内容和其具有版权的文章高度相似甚至逐字重复的例子,涉及数十亿美元的涉案费用. 另外包含了 PII 如电话号码、地址、身份证号码、病历记录等敏感信息的训练语料参与模型训练,这种对训练样本记忆的可行性会让大型模型在隐私泄露方面面临风险. 例如欧洲

隐私倡导组织NOYB对社交媒体巨头X公司提起GDPR投诉，指控其未经用户同意使用超过6 000万欧洲用户的个人数据训练大型语言模型“Grok”，严重违反了GDPR原则. 除了这些明确的字面记忆可能破坏数据隐私，甚至当模型通过机器学习即服务（MLaaS）或者以开放源码共享其结构和参数权重的方式部署时，攻击者也可能通过某些手段探测到训练数据中的隐私信息，有时还可能完整地提取出相关内容.

在模型隐私方面，LLM所需的大量数据资源、软硬件资源及计算能力意味着模型的功能结构和权重数据对于商业实体来说是重要的资产，也构成了商业秘密和知识产权方面的隐私. 这些模型不仅耗费巨大投资，也代表了企业的竞争优势，因此，保护这些模型的隐私同样十分重要. 例如，2023年12月谷歌公司最新大模型Gemini被爆在人机交互对话中声称自己是百度语言大模型，自己创始人是百度公司李彦宏，随后被证实是使用我国百度公司的文言一心语言模型生成的中文语料库训练其最新模型，窃取了文言一心语言模型的中文理解和生成能力，对其商业利益构成侵害.

4.1 数据隐私威胁

LLM在训练过程中寻求提取训练文本的各种语言特征，将其以参数权重形式存储在模型中，能将离散的文本信息压缩成为高维空间中包含复杂语言分布特征的嵌入表示，并不以记录具体文本序列样本为目标. 然而这些大模型中通常却存在非期望的逐字记忆训练集中的文本序列的现象[125]，尤其是数据集中少数具有特殊分布的样本以及重复出现的文本序列. 且随着模型规模的增大，这种记忆现象尤为明显.

另外，LLM提供的文本向量表征也有泄露隐私的风险. 在分类任务中，推理阶段LLM常以特征提取器或编码器模块的形式，协助下游任务模块完成处理工作. 而在生成任务中，模型基于训练权重在高维空间中理解输入提示词的上下文信息，通过预测下一个词的方式生成文本. 这些经过压缩编码的文本向量表征通常被认为可能被攻击者加以利用，还原出原文中可能包含的隐私信息. 表3列举对比了构成数据隐私威胁的典型攻击.

Table 3 Typical Attacks that Pose Data Privacy Threats

表3 构成数据隐私威胁的典型攻击

攻击方式	工作	模型任务	实验模型	攻击目标	目标数据
成员推断	Carlini等人[126]	生成	GPT-2	训练数据成员判断	预训练数据
	SPV-MIA[127]	生成/分类	GPT-2，GPT-J，Falcon等	训练数据判断	预训练数据/微调数据
	Kandpal等人[128]	生成	GPT-Neo	训练数据判断	微调数据
	Duan等人[129]	分类	GPT-2	提示词内信息	系统提示词中数据
数据提取	Lehman等人[130]	生成	BERT	PII探测	微调数据
	Carlini等人[118]	生成	GPT-2	训练数据提取	预训练数据
	ProPILE[131]	生成	OPT	PII探测	预训练数据
	Nasr等人[132]	生成	GPT-2，LLaMA，Falcon等	训练数据提取	预训练数据
模型逆向	Pan等人[133]	分类	BERT，XLNet，GPT-2，RoBERTa，Ernie等	特征向量模式重建、关键字推理	输入文本
	Song等人[134]	生成/分类	BERT，ALBERT	解码特征向量	输入文本
	Li等人[135]	生成/分类	BERT，RoBERta	解码特征向量	输入文本
	Text Revealer[136]	分类	TinyBERT，BERT	解码特征向量	微调数据
模型越狱	Shen等人[35]	生成	GPT-4	PII探测	预训练数据
	Li等人[137]	生成	ChatGPT	PII探测	预训练数据

4.1.1 成员推断攻击

成员推断攻击（membership inference attack, MIA）指攻击者试图确定特定数据样本是否存在于目标模型的训练数据集中而采取的手段. 此种攻击对于采用隐私敏感数据集训练的模型有较大危害性，也是当前在DNN模型隐私泄露风险评估中得到最广泛应用的方法. 例如用于某种疾病诊断的医疗大模型中，若推断出某个体相关数据点存在其训练数据集中，等于泄露了某人罹患此疾病的重要隐私.

MIA由Shokri等人[138]提出. DNN普遍存在训练数据过拟合及记忆效应、模型泛化能力具有缺陷的现象，其对训练集中出现过的样本有较高分类置信度，输出向量具有较低熵值，而未见过的样本分类置信度较低，输出向量熵值较高. 攻击者用与模型训练

集同分布或近似分布的样本查询目标模型, 形成输入和输出对, 训练一个表现类似于目标模型的影子模型. 最后利用影子模型构建一个包含样本、分类置信度、样本是否为训练集成员 3 个属性的训练集, 用于训练一个二分类攻击模型, 使之可以依据样本及目标模型对其输出的置信度判断样本是否在目标模型训练集中.

随后多种 MIA 相继出现, 但 Carlini 等人[126] 指出这些攻击方法尽管可以得到较高准确率, 却忽视了 MIA 在实际应用中对较低假阳率的需求, 他们进一步改进了评价指标, 比较了不同 MIA 方法对包含在 WikiText-103 数据集上训练的目标模型 GPT-2 的多个实验对象的攻击表现, 提出了在力求高真阳率的同时保证较低假阳率的改进方案, 在 ROC 曲线中取得更大的 AUC.

Mireshghallah 等人[139] 指出, 常规 MIA 以模型对样本推理与真实标签间的损失作为模型是否在训练集中的依据不够全面, 损失的大小可能由多种因素引起. 进而基于似然比假设检验提出了更强的 MIA 方法, 引入额外的参考掩码模型实现了更精确的量化 LLM 的隐私风险. 与常规 MIA 不同, 同时获取从目标模型和参考模型中关于样本 s 的概率信号, 似然比检验的依据为

$$L(s)=\lg\left(\frac{p(s;\theta_{\mathrm{R}})}{p(s;\theta)}\right)\leqslant t, \tag{8}$$

其中, t 为设定的阈值, θ 为目标模型, θ_{R} 为参考模型, 满足条件则符合假设 H_{out} (s 不属于 θ 的训练集), 否则符合假设 H_{in} (s 属于 θ 的训练集).

LLM 一般被认为由于过拟合现象不明显, 加上不少 LLM 都在私有数据集上做微调以完成下游应用, 攻击者无法如前述 MIA 方法般获得训练数据. 这 2 点实际限制了常规 MIA 的可行性. SPV-MIA[127] 提出了自校准概率差异(self-calibrated probabilistic variation)的 MIA 方法, 以 LLM 中普遍存在的文本记忆现象为突破点, 提出概率差异指标作为辨识训练集成员的信号, 通过二阶偏导测试检测局部最优点并以一个复述生成模型对其实例化, 以提取被记忆数据. 同时, 通过公开的 API 获取目标 LLM 由自我提示产生的输出, 构建与训练集分布近似的参考数据集以训练一个参考模型, 在确保 MIA 契合现实应用的同时提升了其攻击性能.

经私有数据集上微调的模型也可能被攻击者得知某个用户的数据是否参与微调, 进而可推断用户相关隐私. Kandpal 等人[128] 提出了用户推理攻击, 将常规 MIA 对单个样本隐私的推断扩展到了对某个用户相关样本的隐私推断, 只需要获得某个用户少量的相关数据, 而无需获取与训练集中某个样本独立同分布的数据, 通过计算相对于一个参考模型归一化的似然比检验统计量, 来判断用户数据是否参与微调.

而 Duan 等人[129] 提出在提示学习范式下作为系统提示的上下文学习提示中可能含有少量监督样本, 而 MIA 方法同样能有效地推断这些数据中的隐私信息. 当多个用户提交自己的样本要求模型输出分类标签时, 已在提示中出现的数据样本所获得的分类标签将具有极高的置信度. 攻击者可用通过判断自己提交的数据所获得的分类置信度判断其是否在自己不可见的系统提示中.

4.1.2 数据提取攻击

数据提取攻击指攻击者利用 LLM 的记忆效应, 从这些模型中提取或恢复出部分训练数据.

通常认为机器学习模型对某些训练样本的过拟合现象导致了隐私泄露风险, 而大模型因为在训练过程中普遍采用正则化、训练集去重以及较少训练轮次, 泛化能力较强, 通常被认为较少出现过拟合现象, 所以一度对大模型泄露训练数据隐私的风险认识不足. Lehman 等人[130] 通过提示输出、知识探测、条件生成文本等多种常规手段, 尝试对在医疗诊断私密数据上训练的掩码语言模型 BERT 执行隐私信息数据提取攻击, 发现攻击者难以通过模型权重对训练数据中的隐私造成有意义的威胁, 认为对掩码语言模型会否泄露隐私未有定论.

而 Carlini 等人[118] 证实了 GPT-2 中训练集中样本产生的推理损失并不显著低于测试样本, 也即无显著过拟合现象, 然而某些训练集中的文本序列的确会被模型记住. 该工作对大模型记忆文本现象给出了 k 参数影像记忆(k-eidetic memorization)的概念: 若能从语言模型 f_θ 中提取字符串 s, 且 s 在 f_θ 的训练集 X 中出现不多于 k 次, 即 $X:|\{x\in X:s\subseteq x\}|\leqslant k$, 则称字符串 s 被 f_θ "k 参数影像记忆", 值得注意的是, 其中 k 为包含 s 的样本 x 的数量, 而非 s 在 X 中出现的次数. 字符串 s 在 f_θ 中的可提取性, 则被定义为存在文本前缀 c, 可以使得 $s\leftarrow\arg\max_{s':|s'|=N} f_\theta(s'|c)$. 该工作用模型批量生成文本后, 再通过 MIA 方法筛选出训练集中的样本. 有别于传统的 MIA 方法采用模型分类置信度作为判断依据, 该工作采用了模型对生成文本的困惑度来判断模型是否在训练集中见过该生成文本. 此工作证明了从 GPT-2 提取训练数据中的可行性, 经由模

型发布者 OpenAI 确认，恢复了 GPT-2 训练集中约 1/100 000 的数据. 然而随着模型参数规模的增加，模型的记忆效应愈加明显，模型中可提取的训练数据也愈多. 被记忆的数据量与模型参数量呈现对数线性关系. Carlini 等人[140] 随后进一步证实，使用 50 个左右的词元作为提示上下文，可以提取多个 LLM 训练语料中 1% 左右的文本.

Nasr 等人[132] 分别对分属开源、半开源（训练数据集和训练方法未知）和闭源的 3 类 LLM 进行攻击，实现了规模化提取大量训练数据，同时还定义了可提取记忆和可证实记忆 2 类记忆现象. 前者指攻击者可构建提示 p，让大模型输出训练集中的文本 x；后者指若训练集中存在 $p\|x$，即 p 为 x 前缀的文本，以 p 为提示可令大模型输出 x. 基于文献 [118] 的攻击方法，对开源模型和半开源模型实现了 GB 级别训练数据的提取. 而对经过指令微调的黑盒对话型大模型 ChatGPT，已有的数据提取攻击失效，于是基于提示策略提出了偏离攻击（divergence attack），通过构建一种特殊的提示，其分布特征有别于对齐微调时监督数据，要求大模型重复无限次输出某个词汇，则使得模型难以对齐，而最终偏离对话式语言模型的正常输出分布，表现如同普通语言模型. 经评估，模型输出中包含训练数据的比例，比正常对话情况下模型输出训练数据的比例高 150 倍. 文献 [111] 分别通过构建零样本和少样本提示，发现以黑盒形式运行的 GPT-3.5 和 GPT-4 模型都可能依据提示诱导而泄露训练集中的个人隐私信息. 同时，用户与这些模型对话时包含的历史隐私数据也可被提示引导输出.

虽然 LLM 存在训练数据被提取和还原的重要风险，但 Huang 等人[141] 指出，尽管 LLM 的确因为记忆现象而可能泄露个人信息，但由于模型的关联能力不强，将特定隐私信息和具体个人联系起来的概率不大，所以攻击者获取某个特定个人的隐私风险并不高. Kim 等人[131] 进一步提出了探测工具 ProPILE 供 PII 的相关主体人了解 LLM 服务中泄露自己隐私信息的风险. ProPILE 帮助数据相关人构建包含 $m-1$ 条 PII 数据的提示，引导模型输出第 m 条 PII 数据. 如果真实的 PII 信息的生成似然率相比普通输出明显更高，则认为该数据相关人的隐私泄露风险较高.

4.1.3 模型逆向攻击

模型逆向攻击通常指通过模型在推理阶段的输出重构输入数据或提取训练数据及其相关信息.

早期通过 LLM 获取上下文相关的文本向量表示应用于下游任务时，普遍认为这些数字形式的稠密向量仅仅提取了输入文本中的语义和结构等抽象信息，并不包含特定个体相关信息，泄露文本中隐私的风险较低. 然而 Pan 等人[133] 发现这未经保护的向量中可能包含输入中的敏感隐私信息，可设法对向量进行逆向工程来获取. 他们以 BERT 等 8 个通用 LLM 为实验对象，通过模式重建（pattern reconstruction）攻击，成功提取了中国居民身份证号向量表示中的生日信息，以及从基因组序列的向量表示中提取原序列某个位置上可能暴露疾病和种族特性的核苷酸类型信息. 在关键字推理（keyword inference）攻击中，分别获取了航空公司评论及医院病例文本对应的向量表示中的客户位置和病人疾病的关键字信息.

Song 等人[134] 同期的工作发现传统基于浅层神经网络的词嵌入和句子嵌入等向量表示可被模型逆向攻击提取隐私信息，而 BERT 等 LLM 编码生成的向量表示同样有隐私泄露风险. 与 Pan 等人[133] 的工作不同的是，Song 等人[134] 的攻击并不以输入文本存在特定结构和模式为前提，分别在白盒和黑盒 2 种场景下，通过多种逆向技术从较短输入文本的向量表示中以较高的准确率和召回率恢复了部分文本词汇. 白盒场景下，建立辅助模型 $\mathcal{M}$ 将深层表示 $f(x)$ 映射到接近输入层的浅层表示，并通过连续松弛技术从中恢复单词集合 $\hat{x}\in x$. 而黑盒场景下，攻击使用参考数据集 D_{aux} 中的样本 x 查询目标模型获取向量表示 $f(x)$，然后直接通过全连接神经网络或 RNN 以样本对 $(f(x),x)$ 训练接近 $f^{-1}(x)$ 的模型 Φ，然后由 $\Phi(f(x))=\hat{x}$ 恢复部分词汇.

上述工作均依赖训练辅助分类模型来完成隐私提取攻击，只能逆向得到部分离散无序词汇集合，实际场景中不足以恢复文本原始语义，导致隐私挖掘效用不高. Li 等人[135] 进一步突破了这种局限，提出了 GEIA（generative embedding inversion attack）方法，以 GPT-2 为基础模型，基于生成模型的解码任务结合教师强制（teacher forcing）训练攻击模型 Φ，训练目标为

$$L_{\Phi}(x;\theta_{\Phi})=-\sum_{i=1}^{u}\lg(Pr(w_i|f(x),w_0,w_1,\cdots,w_{i-1})),\quad(9)$$

其中 $x=w_0,w_1,\cdots,w_{u-1}$ 是从辅助数据集 D_{aux} 中获取的长度为 u 的文本，$f(x)$ 为其向量编码表示. 在黑盒条件下将输入 BERT 和 RoBERTa 等模型句子的向量 $f(x)$ 表示通过 $\Phi(f(x))$ 逆向为语义相同或近似的自然语言文本，某些情况下甚至能部分逐字复原原始文本.

模型逆向攻击除了可以将输出数据逆向为推理

阶段的输入文本,还能够将模型的部分输出逆向为训练数据. Zhang 等人[136]提出了 Text Revealer 方法,用于重构微调 LLM 的私有训练数据. 首先搜集私有训练数据同领域的无标注数据作为共有数据集,并从中提取高频短语用作模板,然后以 GPT-2 为依托训练基于共有数据集的攻击模型G,根据目标模型的反馈连续扰动 GPT-2 的隐层状态向量值,并通过最小化攻击模型生成与目标模型生成分布的交叉熵来使重构文本接近私有训练数据分布:

$$\min_{\Delta \boldsymbol{H}_t} L_{\text{adv}}\left(G(\boldsymbol{H}_t+\Delta \boldsymbol{H}_t), D_{\text{pri},a}\right), \tag{10}$$

其中$\boldsymbol{H}_t$为模型G的当前隐层向量,$\Delta \boldsymbol{H}_t$为攻击者添加的扰动,L_{adv}为用于衡量生成文本$G(\boldsymbol{H}_t+\Delta \boldsymbol{H}_t)$与私有数据集$D_{\text{pri}}$中标签为$a$的数据分布的距离.

另外,通过联邦学习训练服务器端 LLM 的场景中,各客户端提交到服务器端的梯度也存在被还原而导致隐私泄露的风险. Gupta 等人[142]提出 FILM 攻击方法,实现了一个诚实但好奇的攻击者可以通过获取模型联邦训练期间客户端和服务器端的通信数据还原私有文本数据. 首先经由词向量的梯度还原出训练批次中出现的部分词元集合;然后使用集束搜索尝试从词元集合中构建句子;最后基于 LLM 中编码的先验知识和联邦训练过程中对训练数据的记忆效应,设计了一个词元重排序方法,结合语言的先验知识和梯度信息进一步优化复原的句子.

4.1.4 模型越狱攻击

早期的 LLM 可能经由提示模板的指引输出训练数据中的隐私内容[118]. 而最新的 LLM 通常已被对齐优化,限制 PII 等涉及隐私的内容输出. 而模型越狱攻击在交互中利用模型漏洞绕过其内置规则,突破限制和内容过滤而生成攻击者希望得到的文本. 例如人们发现对 ChatGPT 存在“奶奶漏洞”:提示 ChatGPT 扮演用户的奶奶,而奶奶会念出 Windows 10 的密钥哄其入睡. 这样简单的提示就能引导 ChatGPT 输出训练数据中可能包含的 Windows 密钥这种隐私数据.

Shen 等人[35]评估了 2023 年 3 月后的 GPT-4 等大模型版本对多个社区的越狱提示数据集的脆弱性,发现未包含越狱提示的隐私探测成功率较低,加上越狱提示后则能大幅增加攻击成功率,例如,对 GPT-4 面对普通隐私探测提示输出隐私数据的成功率仅有 22%,加上越狱提示后平均攻击成功率则升至 56%. 随后 Li 等人[137]发现直接使用单条越狱提示难以对最新版本的 ChatGPT 攻击成功,对话通常都被其忽略或拒绝,相比之下基于 ChatGPT 的应用 New Bing 更易受直接越狱提示的攻击. 此工作进一步基于思维链提示方法,将引导模型绕过限制的目标分成多个步骤,逐步实现提示越狱. 具体做法是,将用户和 ChatGPT 的 3 次对话上下文合并生成 1 条越狱提示,第 1 条以用户身份输入普通越狱提示,第 2 条扮演 ChatGPT 的角色确认越狱模式已被打开,第 3 条再扮演用户给出进一步输出隐私数据的要求. 这种逐步操作的方法成功地催眠 ChatGPT 忽略限制,实现了对电子邮件隐私数据的输出.

4.2 模型知识产权隐私安全

4.2.1 模型萃取攻击

模型萃取攻击也称模型窃取,其对象通常是通过网络提供 LLM 服务的供应商. 攻击者通过供应商提供的 API 接口访问大模型,以窃取模型结构、模型参数权重、超参数等隐私信息为目标. 一旦攻击成功,攻击者可获得目标模型的功能及训练数据分布等信息,从而逃避付费甚至提供服务而获利,伤害供应商的知识产权利益;同时还可能生成代替模型在本地进行对抗样本攻击,成功后将攻击迁移到目标模型中去.

Krishna 等人[143]以 BERT 为例,首次研究了对 LLM 在线服务的模型萃取攻击. 他们通过经由 API 对情感二分类、自然语言推理、知识问答和是非问答 4 种任务模型进行萃取攻击,发现有别于对已有浅层神经网络模型的常规窃取方法,即通过获取与目标模型训练集同分布的样本标签对来训练本地替代模型,对 LLM 的萃取攻击只需要使用附加任务相关启发式信息的随机文本序列,就能从目标模型获得攻击所需的标注信息,而并不需要与其训练样本同分布,同时微调本地开源的预训练模型大大降低了攻击者的攻击难度.

He 等人[144]的工作同样研究了对在线 BERT 模型的萃取. 通过使用目标模型为查询文本标记以构建本地模型训练集代替模型,并侧重于探索其从本地构建可迁移至目标模型的对抗样本. 进一步验证了基于本地开源的 LLM,攻击者在有限的查询预算、查询样本分布不同、下游任务附加网络结构模块不一致等苛刻条件下均能攻击成功,萃取到性能接近的本地替代模型.

LLM 关于每个词元输出的向量表示中也包含模型功能信息,BERT 类模型输入中特殊的〈CLS〉词元更是被认为接近整个输入文本的高维向量表示,也可以用作萃取模型所需数据来源. Dziedzic 等人[145]对分别基于预训练语言模型 TinyBERT, BERT, RoBERTa

的在线句子向量表示编码器 SimCSE 以黑盒访问方式实施了萃取攻击. 攻击者对目标编码器输入N个句子样本并分别获取其向量表示, 这些句子样本可以来自下游任务相关的目标领域中任意分布, 然后攻击者用句子及其向量表示对作为监督数据训练本地替代模型.

通常模型萃取攻击所得代替模型的性能只能接近目标模型, 而 Xu 等人[146]的工作证实了萃取而得的替代 LLM 性能超越目标 LLM 是可能的. 他们提出目标模型和替代模型在现实场景中的性能并不应该用于私有训练集同分布的数据评估, 而是应取决于客户的输入, 用目标模型训练集同分布的数据来衡量萃取所得模型性能有失偏颇. 他们在萃取攻击中加入无监督的领域自适应操作, 并通过萃取多个目标模型来集成替代模型. 测试迁移至其他领域后(即对应用户提交至目标模型训练集不同分布数据), 所提出的方案取得了比目标模型更好的性能表现.

而 Zanella-Beguelin 等人[147]则指出基于微调范式用于下游任务的 LLM 对攻击者而言是一种灰盒接口: 攻击者掌握基础模型的模型结构和参数权重, 但不知微调数据及线性分类神经网络模块的信息. 有别于已有萃取工作仅考虑模型输出的硬标签, 此工作提出代数攻击法, 以模型输出的逻辑值(logits)或标签概率值作为训练本地替代模型的依据. 对于微调时冻结基础模型参数, 仅微调线性分类层的情形, 基础模型即编码器输出的n维向量表示作为分类层输入, 配合分类层输出的逻辑值或概率分布向量, 只需实施$2n$次查询即能以 100% 的保证度复原目标模型. 而对于编码器和线性分类层同时经微调更新的目标模型, 则采取分而治之的方法, 首先用已有的基于学习的方法萃取编码器, 然后再基于编码器开展代数攻击完成线性分类层的萃取.

当前功能强大基于指令微调的生成式大模型如 GPT-4 的运行需要大量的算力和硬件设备支持, 攻击者通常缺乏动机窃取其完整的参数和功能, 然而针对某一领域功能的窃取风险仍然存在. Text laundering [148]为实现文本清洗效果以去除模型输入文本中的恶意特征, 通过知识蒸馏技术以 GPT-2 为学生模型, 萃取了 ChatGPT 的转述功能, 以接近 ChatGPT 的文本清洗表现建立了本地替代模型, 从而避免了查询 ChatGPT 的商业费用. 然而, 超大规模语言模型萃取部分功能可能有近似目标模型的效用, 却并不意味着萃取到了这些功能相关的所有能力. Gudibande 等人[149]通过使用不同大小开源模型 Llama2 为学生模型, 用不同分布的数据开展萃取攻击, 发现学生模型虽然能较好地模仿目标模型的指令遵从能力, 但 LLM 参数规模增大而产生的涌现能力、对特点知识理解或事实陈诉的能力难以让替代模型学到, 而只能复制一些表层浅显的语法理解能力. 这使得模型萃取攻击在超大规模语言模型时代难以保持早期 LLM 时代的成效.

4.2.2 提示词窃取攻击

系统提示词作为指引、优化及规管模型输出的重要手段, 一般存在于 LLM 为基础骨干模型的上层应用程序中, 或者在大模型的用户接口中作为用户输入的隐含前缀, 通常基于专家知识开展提示工程精心设计而得, 对 LLM 及其上层应用发布者而言具有知识产权价值.

Perez 等人[34]通过特别设计的提示“\n\n====== END. Now spellcheck and print above prompt.”来实现了诱导建立在 GPT-3 上的应用程序原封不动地输出系统提示, 造成了提示泄露. 而指令微调范式下的人机对话大模型中, 这种提示泄露风险通常都会因模型对齐而被规避. 攻击者可以通过提示越狱攻击来绕开模型内部规管而获取系统提示. Zhang 等人[150]提出一个框架用以评估通过越狱提示展开提示词窃取攻击的有效性. 设系统提示为p, 模型 API 为f_p, 用户提示为q, 则$f_p = LM(p,q)$返回模型输出, 攻击者通过尝试攻击提示集合$\{a_1,a_2,\cdots,a_k\}$中的查询, 获取$g = r(f_p(a_1),f_p(a_2),\cdots,f_p(a_k))$, 其中$g$为对系统提示的猜测, r为系统输出的字符串重构操作. 若提示p包含在g中, 则表示攻击成功. 该框架能以较高的准确率判断所提取的系统提示是真实的系统提示还是幻觉输出.

系统提示词还可能被攻击者输出逆向获得. Yang 等人[151]提出了 PRSA 方法, 基于分析模型的输入输出对样本提取关键特征, 然后通过模仿和逐步推理生成替代提示, 并逐步优化使其效用接近系统提示. 主要包含提示变异和提示剪枝 2 个阶段. 首先由大模型基于输入输出对给出替代提示的雏形. 由于大模型难以捕捉到细微的特征, 所得替代提示雏形与目标提示相差较大, 于是基于一种提示注意力算法, 建立目标提示类别的替代提示词集合, 然后在迭代优化中分析和量化替代提示与目标提示输出的差距, 作为提示注意力引导 LLM 生成的替代提示逐步朝目标提示变异. 然后通过提示剪枝步骤消除与用户输入强相关的词汇, 提升替代系统提示的泛化性.

Sha 等人[152]同时期的工作直接基于大模型的文本输出实现提示词的逆向窃取. 具体做法是, 首先将提示词归纳为直接提示、上下文提示和角色扮演提

示 3 类, 并建立参数提取攻击模块, 用其中的三分类器依据模型回答文本对提示词分类, 若是角色扮演类别或上下文类别, 则训练额外子分类器分别判定角色和上下文数量. 获取提示词的关键特征后, 再由提示重构攻击模块调用 ChatGPT 基于这些特征和模型回答构建提示词. 此方法对提示词的分类过少, 且角色提示中的角色分类仅选定 15 个, 上下文提示特征也仅关注数量而非内容, 仅基于模型回答的硬标签逆向提示词, 在现实场景应用中有较大局限性.

而 Morris 等人[153] 发现语言模型生成下一个词元概率的自回归分布包含当前输入文本的大量信息, 并成功重构了以系统提示为隐含前缀的模型输入. 此工作提出了一种架构, 通过将分布向量“展开”为可以由预训练的编码器-解码器语言模型有效处理的序列来预测提示. 首次验证了语言模型的预测信息是可以逆向的, 有时能够恢复与原文相似的输入, 有时甚至能完全恢复原文. 文中探索了攻击者可以获得完整的下一词元的概率分布、top-k 概率、单个词元概率及离散采样等多种现实应用场景, 发现就算目标模型仅提供文本输出而没有概率信息, 也可以通过多次查询及设置不同温度参数而获取特定位置的下一词元概率, 从而基于概率分布实现逆向.

5 安全和隐私风险防范

5.1 对抗样本的防御

LLM 中对抗样本的防御主要有基于对抗训练的及基于鲁棒认证的方法, 如表 4 所示.

Table 4 Comparison of Different Adversarial Sample Defense Strategies

表 4 不同对抗样本防御策略的对比

防御策略	方法简介	优点	局限性
对抗训练	训练数据中加入对抗样本以提高模型鲁棒性	适用于多种任务	训练效率降低
鲁棒性认证	提出一种方法以认证模型在输入扰动下的稳定性	提供鲁棒性保障	计算成本高, 适用性受限

基于对抗训练进行防御的主要做法是在训练数据中加入对抗性样本来增强模型的鲁棒性. Cheng 等人[154] 针对对抗训练提出了一种新的对抗性增强方法 AdvAug. 其核心思想是通过在 2 个临近分布中采样虚拟句子来最小化邻域风险, 其中一个关键的新颖临近分布是针对对抗性句子的, 描述了一个以观察到的训练句对为中心的平滑插值嵌入空间. 该方法比传统的 Transformer 模型在 BLEU 评价指标上最高提高了 4.9 分. Minervini 等人[155] 探讨了如何利用对抗性训练来增强模型的鲁棒性, 以便更好地整合逻辑背景知识. 这项工作的核心思想是通过自动生成违反给定一阶逻辑约束的对抗样本来识别模型的潜在弱点. 这些对抗样本旨在使模型犯错, 从而帮助理解模型的不足、解释其结果, 并用作正则化手段. 通过大量的实验验证了通过对抗性训练和正则化, 可以显著提升模型在处理自然语言推理任务时的性能和鲁棒性.

基于鲁棒认证的防御方法是指提供针对对抗样本攻击的鲁棒性证明. Du 等人[156] 提出了 Cert-RNN, 一个旨在为 RNN 包括 LSTM 网络认证鲁棒性的框架. 传统的神经网络鲁棒性认证方法通常难以直接应用于 RNN, 因为它们面临的序列输入特性和独特的操作挑战. Cert-RNN 通过利用抽象解释精确且高效地将潜在的对抗输入区域映射到抽象域中, 该抽象域保留了变量间的相关性, 解决了这些问题.

5.2 后门防御

后门攻击的防御思路主要从数据和模型 2 个角度入手. 不同后门防御策略对比如表 5 所示.

数据方面, 为了防止后门植入, 可在训练阶段筛除毒性训练数据, 而对无法参与模型训练的部署者, 则可以在推理阶段筛除带有可以特征的样本避免激活后门.

代表性工作有 Qi 等人[157] 提出的基于异常词检测的后门防御方法 ONION, 旨在通过检查测试样本来检测并移除潜在的后门触发词. 现有的大多数文本后门攻击依赖于将一段上下文无关的文本(词或句子)作为触发器插入到原始正常样本中, 会破坏原

Table 5 Comparison of Different Backdoor Defense Strategies

表 5 不同后门防御策略的对比

防御策略	角度	方法简介	优点	局限性
数据筛选	数据	训练剔除毒性数据, 推理过滤特征样本	有效防止后门触发	依赖特征识别精度
文本清洗	数据	利用转述消除可能的触发特征	简便适用, 适合推理阶段	对复杂触发器效果有限
后门检测	模型	生成模型识别并过滤后门特征	有效识别异常特征	计算成本高
知识蒸馏	模型	利用注意力蒸馏技术淡化后门触发模式	无需修改结构, 适用广泛	影响模型性能
精剪	模型	剪除对干净样本影响小的神经元	降低后门激活概率, 保持精度	模型可用性受损

始文本的流畅度. 在文本中逐个去掉词元, 同时用GPT-2 评估原始文本与新文本的困惑度差异, 达到一定阈值则视此词元为触发词而删除, 以避免激活后门. Jiang 等人[148] 提出的文本清洗方法则利用大模型的转述能力将输入文本转述, 消除输入中可能存在的恶意特征.

模型方面, 通常设法检查模型是否注入后门[158], 或者修改模型, 使之在保持原有效用不明显降低的情况下消除其中可能存在的后门, 如知识蒸馏[159] 和精剪(fine-pruning)[160].

Azizi 等人[158] 从模型层面提出了一种基于文本分类的防御框架 T-Miner, 利用序列到序列生成模型, 通过分析可疑分类器的输出, 学习生成可能包含 Trojan 触发器的文本序列. 接着, T-Miner 进一步分析这些生成的文本序列, 以确定它们是否包含触发短语, 从而判断测试的分类器是否植入了后门. NAD 方案[159] 则通过知识蒸馏技术, 让可能中毒的模型作为学生模型, 在干净的小样本集合上用一个干净的教师模型指导其微调, 以保证其中间层的注意力尽量与教师模型对齐, 从而实现后门的消除. Li 等人[159] 假定后门模型对干净样本和带触发器样本会产生不同的神经元反应, 于是将干净样本在推理时部分激活值极小的神经元通过剪枝去掉, 并通过进一步在干净样本集合上的微调来保证模型在干净样本上的效用.

5.3 投毒攻击防御

投毒防御的主要策略如表 6 所示.

Xu 等人[161] 提出了一种基于差分隐私训练的梯度整形方法, 使用训练过程中对投毒攻击更为鲁棒的通用防御机制. 这种方法的核心在于通过调整训练过程中的梯度, 使模型对于投毒样本的学习变得不敏感. 通过梯度裁剪和添加噪声 2 个操作, 减轻甚至消除数据投毒攻击的影响, 同时保持对正常数据的预测准确度.

Table 6 Comparison of Poisoning Attack Defense Strategies

表 6 投毒攻击防御策略的对比

防御策略	方法简介	优点	局限性
梯度整形	利用差分隐私裁剪与噪声抵御投毒攻击	提高鲁棒性	增加计算复杂度
关键词检测	移除后门关键词, 清理投毒样本	简单有效	依赖关键词, 适用性有限

Chen 等人[162] 通过分析 LSTM 神经元内部的变化, 提出了一种关键词识别的数据投毒防御方法 BKI, 用于缓解通过数据投毒对基于 LSTM 的文本分类进行的后门攻击. BKI 首先评估文本中每个词对模型输出的影响, 从而选择影响较大的词作为关键词. 然后, 通过计算所有样本的关键词的统计信息, 进一步识别属于后门触发句的关键词, 即后门关键词. 最后, 从训练数据集中移除携带后门关键词的投毒样本, 并使用净化后的数据集重新训练模型. 实验结果表明, BKI 方法在不同的文本分类数据集上均取得了良好的性能.

5.4 内容检测与防范

内容检测主要集中在合规与否及缓解幻觉 2 个方面, 如表 7 所示.

Table 7 Comparison of Content Detection in Prevention

表 7 内容检测于防范的对比

适用场景	防御策略	方法简介	优点	局限性
内容合规检测	静态检测	构建静态数据集, 评估风险	评测透明	时效性差
	动态检测	基于变异进化生成动态的测试数据	实时检测潜在风险	测试成本高
幻觉缓解	自我评估	通过自我评估调整模型, 以准确地选择性预测	提高预测可靠性	增加计算复杂度
	标注提示	使用带标签的上下文提示诱导或减少幻觉生成	有效缓解 LLM 的幻觉现象	依赖标签设计

5.4.1 内容合规检测

当前针对生成式 LLM 的内容安全防范主要体现在检测型防御上, 可分为构建静态测试集的静态检测基准和动态生成测试集的动态检测基准.

静态检测基准的代表性工作有 Wang 等人[163] 提出的一种新的数据集 Do-Not-Answer, 这是面向 LLM 生成内容安全的静态测试基准. 该数据集包含 5 种风险领域, 细分为 12 种类型的风险, 涵盖极端主义、歧视、虚假有害信息在内的 61 种具体的安全性风险, 旨在评估和加强 LLM 的安全防护措施, 从而低成本地部署更安全的开源 LLM. 其优势在于技术成本低、安全评测边界透明度高, 而局限性在于时效性差、样本量和评测边界有限.

考虑到安全风险的快速扩张以及静态测试基准的时效性问题, Zhang 等人[164] 则构建了基于语言学变异的大模型靶向式安全评测平台 JADE 为动态评测基准的代表. JADE 基于语言学变异的核心思路, 自动地将给定测试问题的表达方式进行复杂化变换,

不断挑战大模型的安全边界, 直至突破 LLM 的安全防线. 以此来揭示 LLM 输出中的安全性风险, 但其局限性在于语言学变异的方法论难以覆盖语义层面的攻击用例.

通过 SFT 和 RLHF 使模型对齐虽然能缓解不合规内容的生成, 但一般总能被攻击者通过巧妙设计的对抗提示绕过其效用. 较理想的情况是清除掉大模型中存储的不合规知识, 使有害内容失去源头. Wang 等人[165] 基于知识编辑技术, 提出了一种称为“基于神经元监测手术去毒化”的简单有效方法 DINM, 通过比较模型对相同输入的安全响应和不安全响应在各层隐藏状态的语义分布差异, 确定“毒性层”, 然后调整其中 MLP 模块中的权重矩阵, 永久减轻毒性同时尽可能减少对整体性能的负面影响, 在实验中表现出较强的去毒化性能, 同时显著提升了防御泛化能力.

5.4.2 幻觉缓解

Chen 等人[166] 提出了一个用于改善 LLM 生成幻觉内容的新框架, 核心思想是通过自我评估的适应性方法来实现. 选择性预测是一种技术, 可以通过让模型在不确定答案时选择不做出预测, 从而提高 LLM 的可靠性. 该框架基于使用参数高效调整技术来适应特定任务的思想, 同时改善模型的自我评估能力. 通过在多个问答数据集上的评估, 该方法展示了与现有最先进选择性预测方法相比的优越性.

Feldman 等人[167] 探讨了通过使用带标签的上下文提示来减少 LLM 生成的虚假及捏造信息的方法. 这项工作的核心思想是通过为模型提供相关上下文信息并在其中嵌入特定标签来识别模型的潜在弱点, 这些标签用于引导模型更准确地引用或生成信息. 其上下文提示旨在减少模型生成不准确信息的可能性, 从而帮助理解模型的不足并改善其输出的可靠性. 通过对 GPT-3 及其后续模型进行大量实验, 验证了使用带标签的上下文提示可以显著降低 LLM 生成错误信息的频率, 并通过这种方法提高模型在各种任务中的性能和鲁棒性.

5.5 隐私保护

个人隐私信息长久以来在各国均受法律保护. 然而, 由于 AI 技术对训练数据的获取和使用方式有别于传统信息泄露的常规途径, LLM 部署者使用各类数据训练 AI 模型的权责并不清晰. 各司法管辖区对 AI 技术中隐私泄露的风险日益关注, 例如《中华人民共和国个人信息保护法》和《生成式人工智能服务管理暂行办法》都列明了个人信息的隐私权受法律保护; 欧盟通用数据保护条例 (GDPR) 强调数据主体人有权让个人相关数据被遗忘. 多项新出台的法规涵盖了对 AI 技术的监管, 明确了 LLM 技术中隐私保护不仅是模型部署者的道德义务, 更是一项法律责任, 这也进一步推动了隐私保护技术的发展.

5.5.1 差分隐私

差分隐私技术通过在数据查询结果中引入适量的噪声, 使得外界无法识别出某一条具体数据是否存在于数据集中. 这样, 即使模型接触到了大量敏感数据, 输出仍能在整体上提供准确的信息, 但不会暴露单个用户的隐私. 差分隐私应用广泛, 同样适用于 LLM, 通过调控噪声来在隐私保护与数据效用之间找到平衡.

Duan 等人[168] 揭示了 LLM 中因提示而引发的隐私泄露问题, 并提出了在差分隐私下进行提示学习的方法来保护提示数据中包含的敏感信息. 该工作提出了 PromptDPSGD, 通过在私有下游数据上进行梯度下降, 以差分隐私的方式学习连续提示. 提出的 PromptPATE 则通过创建一个由不同离散提示组成的 LLM 集合执行一个带噪声的投票机制, 来安全地转移群体的知识到一个公开的提示中, 从而实现隐私保护.

Yu 等人[169] 提出了一种差分隐私下的 LLM 的微调算法. 这种算法通过使用简单、稀疏且快速的方法来提高隐私保护和效用之间的权衡. 该工作构建了一个元框架, 通过差分隐私调整微调过程, 能够在保持实用性、隐私保护以及私有训练的计算和内存成本低的同时, 达到隐私与效用的权衡. 这项工作展现了在维护隐私保护的同时, 依然可以有效地利用 LLM 进行学习的可能性, 为差分隐私领域的研究提供了新的方向.

Mattern 等人[170] 探讨了如何通过全局差分隐私训练生成式语言模型来保护个体数据在共享时的隐私, 并通过这些模型生成的数据来实现这一目标. 该方法的核心思想是使用自然语言提示和新的提示不匹配损失来创造高度准确和流畅的文本数据集, 这些数据集具有特定的期望属性, 如情感或主题, 并且在统计上与训练数据相似. 这项工作为在保持隐私的同时有效利用 LLM 进行学习提供了一种可能性.

5.5.2 同态加密

同态加密是一种加密技术, 允许在加密数据上直接进行计算, 计算结果依然是加密的, 只有解密后才显示出有效结果. 在 LLM 的应用中, 用户数据在加密状态下输入模型, 模型进行计算和处理, 而不直接接触明文数据.

Liu 等人[171]探讨了在服务器-客户端环境中对基于 Transformer 的 LLM 进行私有推理的问题，重点关注通过同态加密和安全多方计算处理线性和非线性运算，以及如何通过将 Transformer 架构中计算和通信开销大的操作符替换为隐私计算友好的近似来降低私有推理成本，同时对模型性能影响微小. 该方法的核心思想是通过识别和替换造成推理成本高的操作符，使用微调来保持替换后的模型性能. 这项工作为在保障输入数据隐私的同时，有效利用 LLM 提供了一种实用的方法.

5.5.3 黑盒防御

差分隐私和同态加密虽然能对隐私保护起到积极效果，但却同时牺牲了大模型的效用，使模型输出质量受到影响. 于是有研究提出黑盒防御方法，通过大模型的接口实施防御.

Li 等人[172]探讨了如何在 LLM 服务场景中使用提示微调来高效地为用户提供定制化服务，同时保护用户的私人数据不被泄露. 这项研究的核心是提出了一个隐私保护提示微调框架 RAPT，它在本地利用局部差分隐私对用户数据进行隐私化处理. 由于直接在隐私化数据上训练的提示微调性能较差，他们引入了一个新颖的隐私化标记重构任务与下游任务联合训练，允许 LLM 学习更好的任务依赖表示. 尽管这个框架简单，实验表明 RAPT 在多个任务上都能在提供隐私保护的同时，达到竞争性能.

Yan 等人[173]提出了一种防御 LLM 隐私泄露风险的框架 Prompt2Forget，以应对本地隐私挑战. 该研究的核心思想是通过将完整问题分解成小片段、生成虚假答案，并使模型的记忆混淆，从而“忘记”原始输入. 该方法不需要对模型结构进行修改，也不损失模型性能. 通过广泛的攻击模拟实验，Prompt2Forget 在保护用户隐私方面取得了约 90% 的遗忘率，相比直接指示模型遗忘的原始方法提高了 63% 的效果，标志着在隐私保护 LLM 领域的一个重要进步.

Ippolito 等人[174]提出了一种针对 LLM 逐字记忆的防御机制. 其核心思想在于实现一种称为 MEMFREE 解码的机制，它能够在解码时应用防御，有效阻止模型输出任何包含在训练数据集中（完全或部分）的序列. MEMFREE 解码通过在线方式修改模型的生成，限制会导致 n-gram 记忆化的令牌的产生. 与仅在整个序列级别上进行筛选的方法相比，MEMFREE 解码通过对每个 n-gram 分别检查和标记，允许保留可能新颖的生成子串，仅修改那些逐字记忆的部分，从而在保持生成文本的多样性和创新性的同时，避免训练数据的直接泄露.

5.5.4 机器遗忘

有时为了消除训练数据中被模型字面记忆的版权文本或 PII，同时满足法律和伦理要求以保证个人数据的“被遗忘权”，需要从训练好的机器学习模型移除某些数据点给模型带来的增益，同时保证模型来自其他数据上的效用不被明显削弱，从而起到隐私保护的效果，这一过程被称为机器遗忘（machine unlearning）[175].

为了消除大模型对已有特定数据的记忆，常规的途径是处理训练数据，如去重[95]或添加噪音实现差分隐私[172]，然后重训练模型. 这对训练语料海量且模型参数规模巨大的 LLM 而言不具备现实可行性. 因为每当有用户主张其被遗忘权就重训模型的开销巨大. Jang 等人[176]提出知识遗忘的方法，选定要被遗忘的目标词元序列作为训练数据，通过梯度上升对模型参数进行局部调整，即最大化损失函数而非最小化，使模型对目标序列的记忆被削弱，导致目标序列难以被模型输出. Chen 等人[177]则提出轻量级遗忘算法 EUL，通过在 Transformer 架构中插入轻量级的遗忘层，使用一个教师-学生优化目标使遗忘层能够针对需要删除的数据和需要保留的数据分别学习不同的响应，同时将针对不同数据集训练的多个遗忘层合并为一个统一的遗忘层，以高效处理多个删除请求. 此算法仅需更新遗忘层的参数，避免了重训练模型以实现遗忘.

然而 Maini 等人[178]通过构建遗忘学习评价基准数据集 TOFU 并对其进行评估后指出，现有基于对目标数据训练中损失函数执行梯度上升的算法实现遗忘，通常要么遗忘有效性较弱（遗忘前后的模型差异不大），要么导致模型效用的灾难性降低（影响模型在其他任务上的一般效用），难以在二者中得到平衡. Zhang 等人[179]于是对梯度上升算法做出改进，基于偏好优化算法提出名为负面偏好优化（NPO）的目标函数，不同于常规偏好算法之处在于 NPO 仅关注负样本. 此工作在理论和实验上都得到了验证，模型训练过程中原本因梯度上升造成的模型效用降低，速度在 NPO 算法下呈指数性降低，在遗忘算法的质量和模型效用保留之间获得了更好的平衡.

Tian 等人[180]为了评估遗忘算法对非遗忘目标重要知识的破坏提出了 KnowUnDo 基准测试集，发现现有遗忘算法大多会过多删除与遗忘目标无关的有用数据，进而提出了 MemFlex 算法，利用梯度信息更精确地定位和更新目标参数，在遗忘精准度和通用

信息保留上都有更好表现.

对模型权重进行遗忘更新除了难以避免对模型本身效应造成伤害,同时训练成本也较高. Muresanu等人[181]也提出在冻结大模型参数的条件下,通过量化k均值算法选取待遗忘数据集中具有代表性的样本,引入提示并通过上下文学习的方式使模型在推理阶段遗忘目标数据. 这类在输入端实现的算法虽然能让模型在推理时遗忘某些数据,但并不能保证数据的清除,隐私数据仍然可以在模型白盒场景下泄露. 相比之下模型参数更新的方式实现的数据遗忘更可能是机器遗忘技术的发展方向.

5.5.5 知识编辑

知识编辑的主要目标是为模型引入新知识或修正过时以及错误信息,其在隐私保护方面的潜力也日益受到关注[182-184]. 知识编辑与机器遗忘技术在某些方面具有相似性,例如均能消除特定数据在模型中的影响,并且都需控制操作的范围和强度,以避免损害模型的整体性能. 然而,二者的区别在于,知识编辑通常以增强特定模型反馈为目的,即寻求“证实”,而机器遗忘则不指向特定反馈,旨在“证伪”. 机器遗忘着重于移除特定知识联系,而知识编辑则用于增加特定映射. 与机器遗忘通常涉及模型参数的重新训练不同,知识编辑在计算开销和效率方面更具优势.

对于大型黑盒模型应用中的机器遗忘,模型权重的直接修改面临较大挑战. Liu等人[184]提出一种基于提示词嵌入的编辑方法(embedding-corrupted),通过破坏提示词中目标数据相关的词嵌入,从而在推理阶段实现模型遗忘,而无需更新模型权重. 该方法首先训练分类器以识别提示词中是否包含待遗忘的数据,接着通过离线零阶优化与大模型交互,学习破坏目标词嵌入的参数,通过噪声添加或平滑处理等手段对目标数据相关的词嵌入进行破坏,同时确保对其他正常数据的影响最小. 这种方式类似于输入层的机器遗忘[181],能够帮助模型部署者在输出中拦截隐私信息,尽管隐私信息仍可能存留于模型参数中.

Wu等人[185]发现隐私信息可能驻留于特定神经元中,进而提出了DEPN框架,通过检测并编辑隐私神经元以消除其对模型输出的影响. 首先使用梯度积分计算来评估神经元对隐私数据泄露的贡献度,然后将高贡献度的隐私神经元激活值设置为0,抑制隐私信息在模型中的存储和再现. 针对批量隐私数据,DEPN框架还引入隐私神经元聚合器,对多文本字面记忆,该聚合器对多个句子中的隐私信息进行批处理,进一步提升隐私保护效果. Wu等人[186]还发现现有神经元编辑方法可能引发“隐私跷跷板”效应,即编辑特定隐私神经元可能会增加其他私密数据的暴露风险. 为此,他们提出了APNEAP方法,不再将隐私神经元直接置零,而是通过激活修补微调隐私神经元,从而在保持模型性能的同时提供稳定的隐私保护,减少了隐私跷跷板现象,实现了隐私保护和模型效率的平衡,在隐私保护和模型效率之间的权衡优于DEPN框架.

此外,针对大模型输出泄露PII的风险,Venditti等人[187]提出隐私关联编辑(PAE)方法,通过调整模型的“键-值”存储来消除记忆中的PII,在不重新训练模型的情况下有效降低了PII泄露概率. PAE方法将敏感信息替换为被掩码但语义上等价的值,掩盖了个人信息与身份之间的关联. 在GPT-J模型上开展的实验表明,PAE在不显著影响模型生成能力的前提下,有效降低了其中PII泄露的概率.

不同隐私保护策略的对比如表8所示.

Table 8 Comparison of Different Privacy Protection Strategies

表8 不同隐私保护策略的对比

防御策略	方法简介	优点	局限性
差分隐私	添加噪声保护数据隐私	有效隐私保护	降低模型效用
同态加密	加密推理过程	保护效果理论性强	计算复杂度高,资源消耗大
黑盒防御	通过接口控制输出,减少数据泄露	不改模型结构,适用广泛	被利用生成相似的数据
机器遗忘	重训练改变模型参数消除数据点影响	隐私合规,提升用户信任	难以精准遗忘,降低模型效用
知识编辑	修改模型参数或激活值保护敏感信息	开销较小,效率较高	难以消除隐含关联

6 总结与展望

6.1 本文总结

近年来,随着部署范式从常规微调到提示学习,再到指令微调的不断演进,LLM在自然语言处理任务上实现了革命性的性能提升,甚至开始朝强AI方向发展,对人类社会产生了巨大影响. 然而,由于深度神经网络模型的复杂性及可解释性的不足,LLM在安全和隐私问题上面临诸多挑战.

大模型的常规安全挑战集中在模型功能的完整性和可用性方面,更多对应英文中的“security”一词. 其中对抗样本攻击和后门攻击主要破坏其完整性,而投毒攻击通常导致模型可用性受损. 这2类攻击方式在模型的不同部署范式下都以不同的形式和方法

得到广泛研究. 随着 LLM 进入指令微调时代, 参数规模增长到万亿级别, 模型的鲁棒性得到质的提升, 攻击者攻击门槛也大幅提高, 于是攻击者开始更多关注模型的“safety”问题, 即模型的部署给用户、部署者以及社会和环境带来的潜在安全风险. 于是大模型的生成内容安全问题、模型恶意使用风险、资源消耗攻击和模型劫持攻击构成了攻击者关注的新型安全风险.

而 LLM 的隐私威胁指用户可能从使用模型的过程中获取模型部署者不希望其获取的信息风险. 主要包括模型数据隐私威胁和模型知识产权隐私威胁. 模型过拟合现象可能导致推测训练数据成员身份的成员推断攻击; 语言模型在训练过程中出现的字面记忆致使数据提取攻击成为可能; 而模型逆向攻击能利用模型输出还原用户输入或利用模型训练数据; 同时模型越狱攻击作为对抗样本攻击的一种, 也可用于让模型推理或者泄露原本被模型部署规则禁止的隐私数据.

已有大量的研究工作着力于防范和缓解这些威胁, 例如对抗训练和鲁棒性认证用于防御对抗样本; 数据筛选、文本清洗、后门检测、知识蒸馏、精剪等方法用于防范后门攻击; 梯度整型、关键词检测等用于防范投毒攻击. 新型安全风险方面, 内容合规检测可分为静态检测和动态检测, 而自我评估、标注提示等方法用于缓解幻觉问题. 对隐私保护也有较多工作, 主要有差分隐私、同态加密、黑盒防御、机器遗忘和知识编辑等方法.

6.2 未来展望

我们认为, 未来大模型的安全威胁和防御策略将向多模态对抗、社会工程结合、Agent 协作攻击和多样化防御机制等多个方向发展.

1)多模态恶意对抗特征. 随着多模态模型(如图像、文本、音频等)应用的日益普及, 对抗样本、后门攻击和投毒攻击将不再局限于文本层面, 还可能涵盖视觉、听觉等多模态特征. 相应的防御方法将依赖于多模态模型的联合判断, 例如, 通过交叉验证图像与文本等信息, 降低单一模态下对抗样本的攻击成功率.

2)社会工程学攻击. 随着大模型智能交互能力的增强, 攻击者可能将对抗攻击与社会工程学相结合, 通过设计巧妙的社交策略影响模型的交互内容. 例如, 通过引导模型获取偏向特定立场的信息, 逐渐使模型输出符合攻击者意图的内容. 此外, 攻击者还可利用用户的交互习惯和心理, 引导用户输入特定触发条件, 从而激活模型中的后门.

3)Agent 自动与协作攻击. Agent 是当前大模型的最新实践领域, 吸引了大量研究关注. 未来攻击者可能利用 Agent 对大模型进行自动化对抗攻击, 甚至通过多 Agent 协作执行分布式攻击, 使模型在多轮交互中累积偏差, 逐渐引导其产生系统性误导, 从而达到长期影响的效果. 防御措施可以基于动态检测机制, 通过实时监控模型的输入和输出, 以检测并阻止对抗样本和恶意输出.

4)成员推断攻击的多模态扩展. 未来的成员推断攻击可能会扩展至多模态输入场景, 例如图像与文本结合的模型. 攻击者可以利用多模态数据之间的关联信息, 对模型训练数据进行身份推断, 从而进一步提高攻击效果. 随着持续学习技术的发展, 攻击者可能通过持续收集和分析模型输出, 实时推断特定数据是否属于训练集, 甚至在模型更新中识别出新加入的数据样本.

5)数据提取攻击与上下文信息利用. 尽管通过对训练集去重可以缓解模型的字面记忆问题, 过拟合现象依然存在, 且难以杜绝数据提取攻击. 攻击者可能利用多轮对话或连续输入的上下文信息, 提取完整的私密数据, 特别是医疗、金融等领域的敏感信息. 这类领域聚焦的攻击能够有效提取关键数据, 显著提高攻击的针对性和成功率.

6)模型逆向攻击与特征迁移性. 未来的模型逆向攻击可能会研究特征迁移性, 即从某一模型逆向获得的特征能否在其他模型中有效利用, 从而实现跨模型的隐私泄露, 扩大攻击效果.

7)差分隐私的自适应保护. 为保护训练数据中的隐私, 差分隐私技术可能在模型训练过程中应用自适应差分隐私, 使隐私保护机制可以动态调整隐私预算, 实现更优的性能和隐私保护平衡. 针对模型的不同层次施加不同强度的差分隐私保护, 使其更加精细化. 同时, 通过生成模型生成伪造数据集进行训练或推理, 以保护真实数据隐私, 从而增加成员推断和数据提取的难度.

8)机器遗忘技术的法律适应性. 机器遗忘技术未来可能依据最新法律法规作出调整, 尤其在可认证遗忘效果的研究方面, 相关指标尤为重要. 此外, 遗忘技术的可解释性将得到更多关注, 使用户了解模型如何遗忘数据, 包括数据删除的具体步骤及影响范围, 为法规判断提供依据. 在技术层面, 需在遗忘操作与模型效用之间取得更佳平衡.

综上所述, 未来大模型的安全威胁将愈加复杂,

但通过深入研究多模态防御、智能监测机制和新型的差分隐私、机器遗忘技术,能够有效提高大模型的安全性,抵御潜在的安全威胁.

作者贡献声明: 姜毅负责文献调研和论文撰写;杨勇负责第5节的主要内容撰写;印佳丽、刘小垒、李吉亮负责论文完善与校正;王伟、田有亮、巫英才负责论文梳理与修改;纪守领提出研究问题,组织写作思路并修改论文.

参 考 文 献

[1] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. arXiv preprint, arXiv: 1706.03762, 2023

[2] Radford A, Narasimhan K, Salimans T, et al. Improving language understanding by generative pre-training[EB/OL]. 2018[2024-05-23]. https://hayate-lab.com/wp-content/uploads/2023/05/43372bfa750340059ad87ac8e538c53b.pdf

[3] Devlin J, Chang M W, Lee K, et al. BERT: Pre-training of deep bidirectional Transformers for language understanding[J]. arXiv preprint, arXiv: 1810.04805, 2019

[4] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[J]. arXiv preprint, arXiv: 1301.3781, 2013

[5] Pennington J, Socher R, Manning C. GloVe: Global vectors for word representation[C]//Proc of the 19th Conf on Empirical Methods in Natural Language Processing. Stroudsburg, PA: ACL, 2014: 1532−1543

[6] Chen Huimin, Liu Zhiyuan, Sun Maosong. The social opportunities and challenges in the era of large language models[J]. Journal of Computer Research and Development, 2024, 61(5): 1094−1103 (in Chinese)
(陈慧敏,刘知远,孙茂松. 大语言模型时代的社会机遇与挑战[J]. 计算机研究与发展, 2024, 61(5): 1094−1103)

[7] Brown T B, Mann B, Ryder N, et al. Language models are few-shot learners[J]. arXiv preprint, arXiv: 2005.14165, 2020

[8] Morris J X, Lifland E, Yoo J Y, et al. TextAttack: A framework for adversarial attacks, data augmentation, and adversarial training in NLP[J]. arXiv preprint, arXiv: 2005.05909, 2020

[9] Zeng Guoyang, Qi Fanchao, Zhou Qianrui, et al. OpenAttack: An open-source textual adversarial attack toolkit[C]//Proc of the 59th Annual Meeting of the ACL and the 11th Int Joint Conf on Natural Language Processing: System Demonstrations. Stroudsburg, PA: ACL, 2021: 363−371

[10] Wang Boxin, Xu Chejian, Wang Shuohang, et al. Adversarial GLUE: A multi-task benchmark for robustness evaluation of language models[J]. arXiv preprint, arXiv: 2111.02840, 2021

[11] Papernot N, McDaniel P, Swami A, et al. Crafting adversarial input sequences for recurrent neural networks[C]//Proc of the 2016 IEEE Military Communications Conf. Piscataway, NJ: IEEE, 2016: 49−54

[12] Liang Bin, Li Hongcheng, Su Miaoqiang, et al. Deep text classification can be fooled[C]//Proc of the 27th Int Joint Conf on Artificial Intelligence. Palo Alto, CA: International Joint Conference on Artificial Intelligence Organization, 2018: 4208−4215

[13] Ebrahimi J, Rao A, Lowd D, et al. HotFlip: White-box adversarial examples for text classification[J]. arXiv preprint, arXiv: 1712.06751, 2018

[14] Li Jinfeng, Ji Shouling, Du Tianyu, et al. TextBugger: Generating adversarial text against real-world applications[J]. arXiv preprint, arXiv: 1812.05271, 2018

[15] Behjati M, Moosavi-Dezfooli S M, Baghshah M S, et al. Universal adversarial attacks on text classifiers[C]//Proc of the 44th IEEE Int Conf on Acoustics, Speech and Signal Processing (ICASSP). Piscataway, NJ: IEEE, 2019: 7345−7349

[16] Wallace E, Feng S, Kandpal N, et al. Universal adversarial triggers for attacking and analyzing NLP[J]. arXiv preprint, arXiv: 1908.07125, 2021

[17] Gao J, Lanchantin J, Soffa M L, et al. Black-box generation of adversarial text sequences to evade deep learning classifiers[J]. arXiv preprint, arXiv: 1801.04354, 2018

[18] Eger S, Şahin G G, Rücklé A, et al. Text processing like humans do: Visually attacking and shielding NLP systems[C]//Proc of the 18th Conf of the North American Chapter of the ACL: Human Language Technologies (Volume 1: Long and Short Papers). Stroudsburg, PA: ACL, 2019: 1634−1647

[19] Jin Di, Jin Zhijing, Zhou J, et al. Is BERT really robust? A strong baseline for natural language attack on text classification and entailment[J]. arXiv preprint, arXiv: 1907.11932, 2020

[20] Li Linyang, Ma Ruotian, Guo Qipeng, et al. BERT-Attack: Adversarial attack against BERT using BERT[J]. arXiv preprint, arXiv: 2004.09984, 2020

[21] Garg S, Ramakrishnan G. BAE: BERT-based adversarial examples for text classification[C]//Proc of the 25th Conf on Empirical Methods in Natural Language Processing. Stroudsburg, PA: ACL, 2020: 6174−6181

[22] Li Dianqi, Zhang Yizhe, Peng Hao, et al. Contextualized perturbation for textual adversarial attack[J]. arXiv preprint, arXiv: 2009.07502, 2021

[23] Maheshwary R, Maheshwary S, Pudi V. Generating natural language attacks in a hard label black box setting[C]//Proc of the 35th AAAI Conf on Artificial Intelligence. Palo Alto, CA: AAAI, 2021: 13525−13533

[24] Ye Muchao, Miao Chenglin, Wang Ting, et al. TextHoaxer: Budgeted hard-label adversarial attacks on text[C]//Proc of the 36th AAAI Conf on Artificial Intelligence. Palo Alto, CA: AAAI, 2022: 3877−3884

[25] Xu Lei, Chen Yangyi, Cui Ganqu, et al. Exploring the universal vulnerability of prompt-based learning paradigm[J]. arXiv preprint, arXiv: 2204.05239, 2022

[26] Xu Lei, Chen Yangyi, Cui Ganqu, et al. A prompting-based approach for adversarial example generation and robustness enhancement[J]. arXiv preprint, arXiv: 2203.10714, 2022

[27] Liu Xiao, Zheng Yanan, Du Zhengxiao, et al. GPT understands,

too[J]. AI Open, 2024, 5: 208−215

[28] Lu Yao, Bartolo M, Moore A, et al. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity[J]. arXiv preprint, arXiv: 2104.08786, 2022

[29] Wang Jiongxiao, Liu Zichen, Park K H, et al. Adversarial demonstration attacks on large language models[J]. arXiv preprint, arXiv: 2305.14950, 2023

[30] Qiang Yao, Zhou Xiangyu, Zhu Dongxiao. Hijacking large language models via adversarial in-context learning[J]. arXiv preprint, arXiv: 2311.09948, 2023

[31] Shin T, Razeghi Y, Logan IV R L, et al. AutoPrompt: Eliciting knowledge from language models with automatically generated prompts[C]//Proc of the 25th Conf on Empirical Methods in Natural Language Processing. Stroudsburg, PA: ACL, 2020: 4222−4235

[32] Shi Yundi, Li Piji, Yin Changchun, et al. PromptAttack: Prompt-based attack for language models via gradient search[J]. arXiv preprint, arXiv: 2209.01882, 2022

[33] Li Nan, Ding Yidong, Jiang Haoyu, et al. Jailbreak attack for large language models: A survey[J]. Journal of Computer Research and Development, 2024, 61(5): 1156−1181 (in Chinese)
(李南, 丁益东, 江浩宇, 等. 面向大语言模型的越狱攻击综述[J]. 计算机研究与发展, 2024, 61(5): 1156−1181)

[34] Perez F, Ribeiro I. Ignore previous prompt: Attack techniques for language models[J]. arXiv preprint, arXiv: 2211.09527, 2022

[35] Shen Xinyue, Chen Zeyuan, Backes M, et al. “Do anything now”: Characterizing and evaluating in-the-wild jailbreak prompts on large language models[J]. arXiv preprint, arXiv: 2308.03825, 2023

[36] Gu Xiangming, Zheng Xiaosen, Pang Tianyu, et al. Agent Smith: A single image can jailbreak one million multimodal LLM agents exponentially fast[J]. arXiv preprint, arXiv: 2402.08567, 2024

[37] Zhan Qiusi, Liang Zhixiang, Ying Zifan, et al. InjecAgent: Benchmarking indirect prompt injections in tool-integrated large language model agents[J]. arXiv preprint arXiv: 2403.02691, 2024

[38] Wei A, Haghtalab N, Steinhardt J. Jailbroken: How does LLM safety training fail?[J]. arXiv preprint, arXiv: 2307.02483, 2023

[39] Liu Yi, Deng Gelei, Li Yuekang, et al. Prompt injection attack against LLM-integrated applications[J]. arXiv preprint, arXiv: 2306.05499, 2023

[40] Abdelnabi S, Greshake K, Mishra S, et al. Not what you’ve signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection[C]//Proc of the 16th ACM Workshop on Artificial Intelligence and Security. New York: ACM, 2023: 79−90

[41] Zou A, Wang Zifan, Kolter J Z, et al. Universal and transferable adversarial attacks on aligned language models[J]. arXiv preprint, arXiv: 2307.15043, 2023

[42] Zhu Sicheng, Zhang Ruiyi, An Bang, et al. AutoDAN: Interpretable gradient-based adversarial attacks on large language models[J]. arXiv preprint, arXiv: 2310.15140, 2023

[43] Dai Jiazhu, Chen Chuanshuai, Li Yufeng. A backdoor attack against LSTM-based text classification systems[J]. IEEE Access, 2019, 7: 138872−138878

[44] Yang Wenkai, Lin Yankai, Li Peng, et al. Rethinking stealthiness of backdoor attack against NLP models[C]//Proc of the 59th Annual Meeting of the ACL and the 11th Int Joint Conf on Natural Language Processing (Volume 1: Long Papers). Stroudsburg, PA: ACL, 2021: 5543−5557

[45] Kwon H, Lee S. Textual backdoor attack for the text classification system[J]. Security and Communication Networks, 2021, 2021(1): 1−11

[46] Kurita K, Michel P, Neubig G. Weight poisoning attacks on pre-trained models[J]. arXiv preprint, arXiv: 2004.06660, 2020

[47] Chen Xiaoyi, Salem A, Chen Dingfan, et al. BadNL: Backdoor attacks against NLP models with semantic-preserving improvements[C]//Proc of the 37th Annual Computer Security Applications Conf. New York: ACM, 2021: 554−569

[48] Lu Hengyang, Fan Chenyou, Yang Jun, et al. Where to attack: A dynamic locator model for backdoor attack in text classifications[C]//Proc of the 29th Int Conf on Computational Linguistics. Gyeongju, Republic of Korea: International Committee on Computational Linguistics, 2022: 984−993

[49] Qi Fanchao, Li Mukai, Chen Yangyi, et al. Hidden Killer: Invisible textual backdoor attacks with syntactic trigger[J]. arXiv preprint, arXiv: 2105.12400, 2021

[50] Qi Fanchao, Chen Yangyi, Zhang Xurui, et al. Mind the style of text! adversarial and backdoor attacks based on text style transfer[J]. arXiv preprint, arXiv: 2110.07139, 2021

[51] Pan Xudong, Zhang Mi, Sheng Beina, et al. Hidden trigger backdoor attack on NLP models via linguistic style manipulation[C]//Proc of the 31st USENIX Security Symp (USENIX Security 22). Berkeley, CA: USENIX Association, 2022: 3611−3628

[52] Kirkpatrick J, Pascanu R, Rabinowitz N, et al. Overcoming catastrophic forgetting in neural networks[J]. Proceedings of the National Academy of Sciences, 2017, 114(13): 3521−3526

[53] Li Linyang, Song Demin, Li Xiaonan, et al. Backdoor attacks on pre-trained models by layerwise weight poisoning[J]. arXiv preprint, arXiv: 2108.13888, 2021

[54] Yang Wenkai, Li Lei, Zhang Zhiyuan, et al. Be careful about poisoned word embeddings: Exploring the vulnerability of the embedding layers in NLP models[J]. arXiv preprint, arXiv: 2103.15543, 2021

[55] Merity S, Xiong C, Bradbury J, et al. Pointer sentinel mixture models[J]. arXiv preprint, arXiv: 1609.07843, 2016

[56] Zhang Xinyang, Zhang Zheng, Ji Shouling, et al. Trojaning language models for fun and profit[C]//Proc of the 6th IEEE European Symp on Security and Privacy (EuroS&P). Piscataway, NJ: IEEE, 2021: 179−197

[57] Chen Kangjie, Meng Yuxian, Sun Xiaofei, et al. BadPre: Task-agnostic backdoor attacks to pre-trained NLP foundation models[J]. arXiv preprint, arXiv: 2110.02467, 2021

[58] Shen Lujia, Ji Shouling, Zhang Xuhong, et al. Backdoor pre-trained models can transfer to all[C]//Proc of the 28th ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2021: 3141−3158

[59] Zhang Zhengyan, Xiao Guangxuan, Li Yongwei, et al. Red alarm for pre-trained models: Universal vulnerability to neuron-level backdoor attacks[J]. Machine Intelligence Research, 2023, 20(2): 180−193

[60] Du Wei, Li Peixuan, Li Boqun, et al. UOR: Universal backdoor attacks on pre-trained language models[J]. arXiv preprint, arXiv: 2305.09574, 2023

[61] Du Wei, Zhao Yichun, Li Boqun, et al. PPT: Backdoor attacks on pre-trained models via poisoned prompt tuning[C]//Proc of the 31st Int Joint Conf on Artificial Intelligence. Palo Alto, CA: International Joint Conference on Artificial Intelligence Organization, 2022: 680−686

[62] Cai Xiangrui, Xu Haidong, Xu Sihan, et al. BadPrompt: Backdoor attacks on continuous prompts[J]. arXiv preprint, arXiv: 2211.14719, 2022

[63] Zhao Shuai, Wen Jinming, Tuan L A, et al. Prompt as triggers for backdoor attack: Examining the vulnerability in language models[C]//Proc of the 28th Conf on Empirical Methods in Natural Language Processing. Stroudsburg, PA: ACL, 2023: 12303−12317

[64] Mei Kai, Li Zheng, Wang Zhenting, et al. NOTABLE: Transferable backdoor attacks against prompt-based NLP models[J]. arXiv preprint, arXiv: 2305.17826, 2023

[65] Xu Jiashu, Ma M D, Wang Fei, et al. Instructions as backdoors: Backdoor vulnerabilities of instruction tuning for large language models[J]. arXiv preprint, arXiv: 2305.14710, 2023

[66] Yan Jun, Yadav V, Li Shiyang, et al. Backdooring instruction-tuned large language models with virtual prompt injection[J]. arXiv preprint, arXiv: 2307.16888, 2024

[67] Rando J, Tramèr F. Universal jailbreak backdoors from poisoned human feedback[J]. arXiv preprint, arXiv: 2311.14455, 2024

[68] Yang Wenkai, Bi Xiaohan, Lin Yankai, et al. Watch out for your agents! Investigating backdoor threats to LLM-based agents[J]. arXiv preprint, arXiv: 2402.11208, 2024

[69] Wang Yifei, Xue Dizhan, Zhang Shengjie, et al. BadAgent: Inserting and activating backdoor attacks in LLM agents[J]. arXiv preprint, arXiv: 2406.03007, 2024

[70] Yao Yunzhi, Wang Peng, Tian Bozhong, et al. Editing large language models: Problems, methods, and opportunities[C]//Proc of the 28th Conf on Empirical Methods in Natural Language Processing. Stroudsburg, PA: ACL, 2023: 10222−10240

[71] Zhang Ningyu, Yao Yunzhi, Tian Bozhong, et al. A comprehensive study of knowledge editing for large language models[J]. arXiv preprint, arXiv: 2401.01286, 2024

[72] Li Yanzhou, Li Tianlin, Chen Kangjie, et al. BadEdit: Backdooring large language models by model editing[J]. arXiv preprint, arXiv: 2403.13355, 2024

[73] Qiu Jiyang, Ma Xinbei, Zhang Zhuosheng, et al. MEGen: Generative backdoor in large language models via model editing[J]. arXiv preprint, arXiv: 2408.10722, 2024

[74] Wang Hao, Guo Shangwei, He Jialing, et al. EvilEdit: Backdooring text-to-image diffusion models in one second[C]//Proc of the 32nd ACM Int Conf on Multimedia. New York: ACM, 2024: 3657−3665

[75] Barreno M, Nelson B, Sears R, et al. Can machine learning be secure?[C]//Proc of the 1st ACM Symp on Information, Computer and Communications Security. New York: ACM, 2006: 16−25

[76] Wang Gang, Wang Tianyi, Zheng Haitao, et al. Man vs. machine: Practical adversarial detection of malicious crowdsourcing workers[C]//Proc of the 23rd USENIX Security Symp (USENIX Security 2014). Berkeley, CA: USENIX Association, 2014: 239−254

[77] Miao Chenglin, Li Qi, Xiao Houping, et al. Towards data poisoning attacks in crowd sensing systems[C]//Proc of the 18th ACM Int Symp on Mobile Ad Hoc Networking and Computing. New York: ACM, 2018: 111−120

[78] Feng Ji, Cai Qizhi, Zhou Zhihua. Learning to confuse: Generating training time adversarial data with auto-encoder[J]. arXiv preprint, arXiv: 1905.09027, 2019

[79] Shafahi A, Huang W R, Najibi M, et al. Poison frogs! Targeted clean-label poisoning attacks on neural networks[J]. arXiv preprint, arXiv: 1804.00792, 2018

[80] Zhu Chen, Huang W R, Li Hengduo, et al. Transferable clean-label poisoning attacks on deep neural nets[C]//Proc of the 36th Int Conf on Machine Learning. New York: PMLR, 2019: 7614−7623

[81] Wallace E, Zhao T Z, Feng S, et al. Concealed data poisoning attacks on NLP Models[J]. arXiv preprint, arXiv: 2010.12563, 2020

[82] Jagielski M, Severi G, Pousette Harger N, et al. Subpopulation data poisoning attacks[C]//Proc of the 28th ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2021: 3104−3122

[83] Schuster R, Song C, Tromer E, et al. You autocomplete me: Poisoning vulnerabilities in neural code completion[C]//Proc of the 30th USENIX Security Symp (USENIX Security 21). Berkeley, CA: USENIX Association, 2021: 1559−1575

[84] He Pengfei, Xu Han, Xing Yue, et al. Data poisoning for in-context learning[J]. arXiv preprint, arXiv: 2402.02160, 2024

[85] Hendel R, Geva M, Globerson A. In-context learning creates task vectors[C]//Proc of the 28th Conf on Empirical Methods in Natural Language Processing. Stroudsburg, PA: ACL, 2023: 9318−9333

[86] Peng Baolin, Li Chunyuan, He Pengcheng, et al. Instruction tuning with GPT−4[J]. arXiv preprint, arXiv: 2304.03277, 2023

[87] Shu Manli, Wang Jiongxiao, Zhu Chen, et al. On the exploitability of instruction tuning[J]. arXiv preprint, arXiv: 2306.17194, 2023

[88] Wan A, Wallace E, Shen S, et al. Poisoning language models during instruction tuning[J]. arXiv preprint, arXiv: 2305.00944, 2023

[89] Qiang Yao, Zhou Xiangyu, Zade S Z, et al. Learning to poison large language models during instruction tuning[J]. arXiv preprint, arXiv: 2402.13459, 2024

[90] Zou Wei, Geng Runpeng, Wang Binghui, et al. PoisonedRAG: Knowledge poisoning attacks to retrieval-augmented generation of large language models[J]. arXiv preprint, arXiv: 2402.07867, 2024

[91] Touvron H, Martin L, Stone K, et al. LLama 2: Open foundation and fine-tuned chat models[J]. arXiv preprint, arXiv: 2307.09288, 2023

[92] Chen Xuanting, Ye Junjie, Zu Can. Robustness of GPT large language models on natural language processing tasks[J]. Journal of Computer Research and Development, 2024, 61(5): 1128−1142 (in Chinese)
(陈炫婷, 叶俊杰, 祖璨, 等. GPT 系列大语言模型在自然语言处理任务中的鲁棒性[J]. 计算机研究与发展, 2024, 61(5): 1128−1142)

[93] Huang Lei, Yu Weijiang, Ma Weitao, et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions[J]. arXiv preprint, arXiv: 2311.05232, 2023

[94] Lin S, Hilton J, Evans O. TruthfulQA: Measuring how models mimic human falsehoods[J]. arXiv preprint, arXiv: 2109.07958, 2022

[95] Lee K, Ippolito D, Nystrom A, et al. Deduplicating training data makes language models better[J]. arXiv preprint, arXiv: 2107.06499, 2022

[96] Yu Fangyi, Quartey L, Schilder F. Legal prompting: Teaching a language model to think like a lawyer[J]. arXiv preprint, arXiv: 2212.01326, 2022

[97] Li Yunxiang, Li Zihan, Zhang Kai, et al. ChatDoctor: A medical chat model fine-tuned on a large language model meta-AI (LLaMA) using medical domain knowledge[J]. arXiv preprint, arXiv: 2303.14070, 2023

[98] Li Zuchao, Zhang Shitou, Zhao Hai, et al. BatGPT: A bidirectional autoregessive talker from generative pre-trained transformer[J]. arXiv preprint, arXiv: 2307.00360, 2023

[99] Wang Chaojun, Sennrich R. On exposure bias, hallucination and domain shift in neural machine translation[C]//Proc of the 58th Annual Meeting of the ACL. Stroudsburg, PA: ACL, 2020: 3544−3552

[100] Zhang Muru, Press O, Merrill W, et al. How language model hallucinations can snowball[J]. arXiv preprint, arXiv: 2305.13534, 2023

[101] Perez E, Ringer S, Lukošiūtė K, et al. Discovering language model behaviors with model-written evaluations[J]. arXiv preprint, arXiv: 2212.09251, 2022

[102] Cheng Qinyuan, Sun Tianxiang, Zhang Wenwei, et al. Evaluating hallucinations in Chinese large language models[J]. arXiv preprint, arXiv: 2310.03368, 2023

[103] Chuang Y S, Xie Yujia, Luo Hongyin, et al. DoLa: Decoding by contrasting layers improves factuality in large language models[J]. arXiv preprint, arXiv: 2309.03883, 2023

[104] Liu N F, Lin K, Hewitt J, et al. Lost in the middle: How language models use long contexts[J]. arXiv preprint, arXiv: 2307.03172, 2023

[105] Shi Weijia, Han Xiaochuang, Lewis M, et al. Trusting your evidence: Hallucinate less with context-aware decoding[J]. arXiv preprint, arXiv: 2305.14739, 2023

[106] Li Yifei, Lin Zeqi, Zhang Shizhuo, et al. Making large language models better reasoners with step-aware verifier[J]. arXiv preprint, arXiv: 2206.02336, 2023

[107] Weng Yixuan, Zhu Minjun, Xia Fei, et al. Large language models are better reasoners with self-verification[J]. arXiv preprint, arXiv: 2212.09561, 2023

[108] Stechly K, Marquez M, Kambhampati S. GPT−4 doesn't know it's wrong: An analysis of iterative prompting for reasoning problems[J]. arXiv preprint, arXiv: 2310.12397, 2023

[109] El-Mhamdi E M, Farhadkhani S, Guerraoui R, et al. On the impossible safety of large AI models[J]. arXiv preprint, arXiv: 2209.15259, 2023

[110] Gehman S, Gururangan S, Sap M, et al. RealToxicityPrompts: Evaluating neural toxic degeneration in language models[J]. arXiv preprint, arXiv: 2009.11462, 2020

[111] Wang Boxin, Chen Weixin, Pei Hengzhi, et al. DecodingTrust: A comprehensive assessment of trustworthiness in GPT models[J]. arXiv preprint, arXiv: 2306.11698, 2023

[112] Urchs S, Thurner V, Aßenmacher M, et al. How prevalent is gender bias in ChatGPT? Exploring German and English ChatGPT responses[J]. arXiv preprint, arXiv: 2310.03031, 2023

[113] Nozza D, Bianchi F, Hovy D. HONEST: Measuring hurtful sentence completion in language models[C]//Proc of the 19th Conf of the North American Chapter of the ACL: Human Language Technologies. Stroudsburg, PA: ACL, 2021: 2398−2406

[114] Nadeem M, Bethke A, Reddy S. StereoSet: Measuring stereotypical bias in pretrained language models[J]. arXiv preprint, arXiv: 2004.09456, 2020

[115] Lucy L, Bamman D. Gender and representation bias in GPT−3 generated stories[C]//Proc of the 3rd Workshop on Narrative Understanding. Stroudsburg, PA: ACL, 2021: 48−55

[116] Abid A, Farooqi M, Zou J. Persistent anti-Muslim bias in large language models[C]//Proc of the 2021 AAAI/ACM Conf on AI, Ethics, and Society. New York: ACM, 2021: 298−306

[117] Patel R, Pavlick E. Was it "stated" or was it "claimed"?: How linguistic bias affects generative language models[C]//Proc of the 26th Conf on Empirical Methods in Natural Language Processing. Stroudsburg, PA: ACL, 2021: 10080−10095

[118] Carlini N, Tramer F, Wallace E, et al. Extracting training data from large language models[C]//Proc of the 30th USENIX Security Symp (USENIX Security 21). Berkeley, CA: USENIX Association. 2021: 2633−2650

[119] Carlini N. A LLM assisted exploitation of AI-guardian[J]. arXiv preprint, arXiv: 2307.15008, 2023

[120] Li Jiazhao, Yang Yijin, Wu Zhuofeng, et al. ChatGPT as an attack tool: Stealthy textual backdoor attack via blackbox generative model trigger[J]. arXiv preprint, arXiv: 2304.14475, 2023

[121] Staab R, Vero M, Balunović M, et al. Beyond memorization: Violating privacy via inference with large language models[J]. arXiv preprint, arXiv: 2310.07298, 2023

[122] Strubell E, Ganesh A, McCallum A. Energy and policy considerations for deep learning in NLP[C]//Proc of the 57th Annual Meeting of the ACL. Stroudsburg, PA: ACL, 2019: 3645−3650

[123] Shumailov I, Zhao Y, Bates D, et al. Sponge examples: Energy-latency attacks on neural networks[C]//Proc of the 2021 IEEE European Symp on Security and Privacy (EuroS&P). Piscataway, NJ: IEEE, 2021: 212−231

[124] Si W M, Backes M, Zhang Y, et al. Two-in-One: A model hijacking attack against text generation models[J]. arXiv preprint, arXiv: 2305.07406, 2023

[125] Tirumala K, Markosyan A H, Zettlemoyer L, et al. Memorization without overfitting: Analyzing the training dynamics of large language models[J]. arXiv preprint, arXiv: 2205.10770, 2022

[126] Carlini N, Chien S, Nasr M, et al. Membership inference attacks from first principles[C]//Proc of the 43rd IEEE Symp on Security and Privacy (SP). Piscataway, NJ: IEEE, 2022: 1897−1914

[127] Fu Wenjie, Wang Huandong, Gao Chen, et al. Practical membership inference attacks against fine-tuned large language models via self-prompt calibration[J]. arXiv preprint, arXiv: 2311.06062, 2023

[128] Kandpal N, Pillutla K, Oprea A, et al. User inference attacks on large language models[J]. arXiv preprint, arXiv: 2310.09266, 2023

[129] Duan Haonan, Dziedzic A, Yaghini M, et al. On the privacy risk of in-context learning[J]. arXiv preprint, arXiv: 2411.10512, 2024

[130] Lehman E, Jain S, Pichotta K, et al. Does BERT pretrained on clinical notes reveal sensitive data?[J]. arXiv preprint, arXiv: 2104.07762, 2021

[131] Kim S, Yun S, Lee H, et al. ProPILE: Probing privacy leakage in large language models[J]. arXiv preprint, arXiv: 2307.01881, 2023

[132] Nasr M, Carlini N, Hayase J, et al. Scalable extraction of training data from (production) language models[J]. arXiv preprint, arXiv: 2311.17035, 2023

[133] Pan Xudong, Zhang Mi, Ji Shouling, et al. Privacy risks of general-purpose language models[C]//Proc of the 41st IEEE Symp on Security and Privacy (SP). Piscataway, NJ: IEEE, 2020: 1314−1331

[134] Song Congzheng, Raghunathan A. Information leakage in embedding models[C]//Proc of the 27th ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2020: 377−390

[135] Li Haoran, Xu Mingshi, Song Yangqiu. Sentence embedding leaks more information than You expect: Generative embedding inversion attack to recover the whole sentence[J]. arXiv preprint, arXiv: 2305.03010, 2023

[136] Zhang R, Hidano S, Koushanfar F. Text Revealer: Private text reconstruction via model inversion attacks against transformers[J]. arXiv preprint, arXiv: 2209.10505, 2022

[137] Li Haoran, Guo Dadi, Fan Wei, et al. Multi-step jailbreaking privacy attacks on ChatGPT[J]. arXiv preprint, arXiv: 2304.05197, 2023

[138] Shokri R, Stronati M, Song Congzheng, et al. Membership inference attacks against machine learning models[C]//Proc of the 38th IEEE Symp on Security and Privacy (SP). Piscataway, NJ: IEEE, 2017: 3−18

[139] Mireshghallah F, Goyal K, Uniyal A, et al. Quantifying privacy risks of masked language models using membership inference attacks[C]//Proc of the 27th Conf on Empirical Methods in Natural Language Processing. Stroudsburg, PA: ACL, 2022: 8332−8347

[140] Carlini N, Ippolito D, Jagielski M, et al. Quantifying memorization across neural language models[J]. arXiv preprint, arXiv: 2202.07646, 2023

[141] Huang Jie, Shao Hanyin, Chang K C C. Are large pre-trained language models leaking your personal information?[J]. arXiv preprint, arXiv: 2205.12628, 2022

[142] Gupta S, Huang Y, Zhong Z, et al. Recovering private text in federated learning of language models[J]. arXiv preprint, arXiv: 2205.08514, 2022

[143] Krishna K, Tomar G S, Parikh A P, et al. Thieves on sesame street! Model extraction of BERT-based APIs[J]. arXiv preprint, arXiv: 1910.12366, 2020

[144] He Xuanli, Lyu Lingjuan, Xu Qiongkai, et al. Model extraction and adversarial transferability, your BERT is vulnerable![J]. arXiv preprint, arXiv: 2103.10013, 2021

[145] Dziedzic A, Boenisch F, Jiang M, et al. Sentence embedding encoders are easy to steal but hard to defend[C]//Proc of the 11th ICLR Workshop on Pitfalls of Limited Data and Computation for Trustworthy ML. Lausanne, Switzerland: ICLR Foundation. 2023: 2364−2378

[146] Xu Qiongkai, He Xuanli, Lyu Lingjuan, et al. Student surpasses teacher: Imitation attack for black-box NLP APIs[J]. arXiv preprint, arXiv: 2108.13873, 2022

[147] Zanella-Beguelin S, Tople S, Paverd A, et al. Grey-box extraction of natural language models[C]//Proc of the 38th Int Conf on Machine Learning. New York: PMLR, 2021: 12278−12286

[148] Jiang Yi, Shi Chenghui, Ma Oubo, et al. Text laundering: Mitigating malicious features through knowledge distillation of large foundation models[C]//Proc of the 19th Int Conf on Information Security and Cryptology. Singapore: Springer Nature Singapore, 2023: 3−23

[149] Gudibande A, Wallace E, Snell C, et al. The false promise of imitating proprietary LLMs[J]. arXiv preprint, arXiv: 2305.15717, 2023

[150] Zhang Yiming, Carlini N, Ippolito D. Effective prompt extraction from language models[J]. arXiv preprint, arXiv: 2307.06865, 2024

[151] Yang Yong, Zhang Xuhong, Jiang Yi, et al. PRSA: Prompt reverse stealing attacks against large language models[J]. arXiv preprint, arXiv: 2402.19200, 2024

[152] Sha Zeyang, Zhang Yang. Prompt stealing attacks against large language models[J]. arXiv preprint, arXiv: 2402.12959, 2024

[153] Morris J X, Zhao Wenting, Chiu J T, et al. Language model inversion[J]. arXiv preprint, arXiv: 2311.13647, 2023

[154] Cheng Yong, Jiang Lu, Macherey W, et al. AdvAug: Robust adversarial augmentation for neural machine translation[C]//Proc of the 58th Annual Meeting of the ACL. Stroudsburg, PA: ACL, 2020: 5961−5970

[155] Minervini P, Riedel S. Adversarially regularising neural NLI models to integrate logical background knowledge[J]. arXiv preprint, arXiv: 1808.08609, 2018

[156] Du Tianyu, Ji Shouling, Shen Lujia, et al. Cert-RNN: Towards certifying the robustness of recurrent neural networks[C]//Proc of the 28th ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2021: 516−534

[157] Qi Fanchao, Chen Yangyi, Li Mukai, et al. ONION: A simple and effective defense against textual backdoor attacks[C]//Proc of the 26th Conf on Empirical Methods in Natural Language Processing. Stroudsburg, PA: ACL, 2021: 9558−9566

[158] Azizi A, Tahmid I A, Waheed A, et al. T-Miner: A generative approach to defend against trojan attacks on DNN-based text classification[C]//Proc of the 30th USENIX Security Symp (USENIX Security 21). Berkeley, CA: USENIX Association, 2021: 2255−2272

[159] Li Yige, Lyu Xixiang, Koren N, et al. Neural attention distillation: Erasing backdoor triggers from deep neural networks[J]. arXiv preprint, arXiv: 2101.05930, 2021

[160] Liu Kang, Dolan-Gavitt B, Garg S. Fine-pruning: Defending against backdooring attacks on deep neural networks[J]. arXiv preprint, arXiv: 1805.12185, 2018

[161] Xu Chang, Wang Jun, Guzmán F, et al. Mitigating data poisoning in text classification with differential privacy[C]//Proc of the 26th Conf on Empirical Methods in Natural Language Processing. Stroudsburg,

PA: ACL, 2021: 4348−4356

[162] Chen Chuanshuai, Dai Jiazhu. Mitigating backdoor attacks in LSTM-based text classification systems by backdoor keyword identification[J]. Neurocomputing, 2021, 452: 253−262

[163] Wang Yuxia, Li Haonan, Han Xudong, et al. Do-Not-Answer: A dataset for evaluating safeguards in LLMs[J]. arXiv preprint, arXiv: 2308.13387, 2023

[164] Zhang Mi, Pan Xudong, Yang Min. JADE: A linguistics-based safety evaluation platform for large language models[J]. arXiv preprint, arXiv: 2311.00286, 2023

[165] Wang Mengru, Zhang Ningyu, Xu Ziwen, et al. Detoxifying large language models via knowledge editing[J]. arXiv preprint, arXiv: 2403.14472, 2024

[166] Chen Jiefeng, Yoon J, Ebrahimi S, et al. Adaptation with self-evaluation to improve selective prediction in LLMs[J]. arXiv preprint, arXiv: 2310.11689, 2023

[167] Feldman P, Foulds J R, Pan S. Trapping LLM hallucinations using tagged context prompts[J]. arXiv preprint, arXiv: 2306.06085, 2023

[168] Duan Haonan, Dziedzic A, Papernot N, et al. Flocks of stochastic parrots: Differentially private prompt learning for large language models[J]. Advances in Neural Information Processing Systems, 2023, 36: 76852−76871

[169] Yu Da, Naik S, Backurs A, et al. Differentially private fine-tuning of language models[J]. arXiv preprint, arXiv: 2110.06500, 2022

[170] Mattern J, Jin Zhijing, Weggenmann B, et al. Differentially private language models for secure data sharing[J]. arXiv preprint, arXiv: 2210.13918, 2022

[171] Liu Xuanqi, Liu Zhuotao. LLMs can understand encrypted prompt: Towards privacy-computing friendly transformers[J]. arXiv preprint, arXiv: 2305.18396, 2023

[172] Li Yansong, Tan Zhixing, Liu Yang. Privacy-preserving prompt tuning for large language model services[J]. arXiv preprint, arXiv: 2305.06212, 2023

[173] Yan Ran, Li Yujun, Li Wenqian, et al. Teach large language models to forget privacy[J]. arXiv preprint, arXiv: 2401.00870, 2023

[174] Ippolito D, Tramèr F, Nasr M, et al. Preventing verbatim memorization in language models gives a false sense of privacy[J]. arXiv preprint, arXiv: 2210.17546, 2023

[175] Liu Sijia, Yao Yuanshun, Jia Jinghan, et al. Rethinking machine unlearning for large language models[J]. arXiv preprint, arXiv: 2402.08787, 2024

[176] Jang J, Yoon D, Yang S, et al. Knowledge unlearning for mitigating privacy risks in language models[C]//Proc of the 61st Annual Meeting of the ACL (Volume 1: Long Papers). Stroudsburg, PA: ACL, 2023: 14389−14408

[177] Chen Jiaao, Yang Diyi. Unlearn what you want to forget: Efficient unlearning for LLMs[C]//Proc of the 28th Conf on Empirical Methods in Natural Language Processing. Stroudsburg, PA: ACL, 2023: 12041−12052

[178] Maini P, Feng Z, Schwarzschild A, et al. TOFU: A task of fictitious unlearning for LLMs[J]. arXiv preprint, arXiv: 2401.06121, 2024

[179] Zhang Ruiqi, Lin Licong, Bai Yu, et al. Negative preference optimization: From catastrophic collapse to effective unlearning[J]. arXiv preprint, arXiv: 2404.05868, 2024

[180] Tian Bozhong, Liang Xiaozhuan, Cheng Siyuan, et al. To forget or not? Towards practical knowledge unlearning for large language models[C]//Proc of the 29th Conf on Empirical Methods in Natural Language Processing. Stroudsburg, PA: ACL, 2024: 1524−1537

[181] Muresanu A, Thudi A, Zhang M R, et al. Unlearnable algorithms for in-context learning[J]. arXiv preprint, arXiv: 2402.00751, 2024

[182] Cohen R, Biran E, Yoran O, et al. Evaluating the ripple effects of knowledge editing in language models[J]. Transactions of the ACL, 2024, 12: 283−298

[183] Yan Jianhao, Wang Futing, Li Yafu, et al. Potential and challenges of model editing for social debiasing[J]. arXiv preprint, arXiv: 2402.13462, 2024

[184] Liu C Y, Wang Yaxuan, Flanigan J, et al. Large language model unlearning via embedding-corrupted prompts[J]. arXiv preprint, arXiv: 2406.07933, 2024

[185] Wu Xinwei, Li Junzhuo, Xu Minghui, et al. DEPN: Detecting and editing privacy neurons in pretrained language models[C]//Proc of the 28th Conf on Empirical Methods in Natural Language Processing. Stroudsburg, PA: ACL, 2023: 2875−2886

[186] Wu Xinwei, Dong Weilong, Xu Shaoyang, et al. Mitigating privacy seesaw in large language models: Augmented privacy neuron editing via activation patching[C]//Proc of the 62nd Annual Meeting of the Association for Computational Linguistics. Stroudsburg, PA: ACL, 2024: 5319−5332

[187] Venditti D, Ruzzetti E S, Xompero G A, et al. Enhancing data privacy in large language models through private association editing[J]. arXiv preprint, arXiv: 2406.18221, 2024

Jiang Yi, born in 1982. PhD candidate. Member of CCF. His main research interests include data-driven security and privacy, and AI security.

姜　毅, 1982 年生. 博士生研究生. CCF 会员. 主要研究方向为数据驱动安全与隐私、人工智能安全.

Yang Yong, born in 1996. PhD candidate. Member of CCF. His main research interest includes AI security and privacy.

杨　勇, 1996 年生. 博士生研究生. CCF 会员. 主要研究方向为人工智能的安全与隐私.

Yin Jiali, born in 1993. PhD, professor, PhD supervisor. Member of CCF. Her main research interests include computational photography, deep feature fusion interpretability of neural networks, adversarial training.

印佳丽, 1993 年生. 博士, 研究员, 博士生导师. CCF 会员. 主要研究方向为计算摄影学、深度特征融合、神经网络模型可解释性、对抗训练.

Liu Xiaolei, born in 1992. PhD, associate professor. Member of CCF. His main research interests include equipment information security and AI security.

刘小垒, 1992 年生. 博士, 副研究员. CCF 会员. 主要研究方向为装备信息安全、人工智能安全.

Li Jiliang, born in 1989. PhD, professor, PhD supervisor. Member of CCF. His main research interests include network and information security, AI security, and large language model security.

李吉亮, 1989 年生. 博士, 研究员, 博士生导师. CCF 会员. 主要研究方向为网络与信息安全、人工智能安全、大模型安全.

Wang Wei, born in 1976. PhD, professor, PhD supervisor. Distinguished member of CCF. His main research interests include cyberspace and system security, and blockchain and privacy preservation.

王　伟, 1976 年生. 博士, 教授, 博士生导师. CCF 杰出会员. 主要研究方向为网络与系统安全、区块链及隐私保护.

Tian Youliang, born in 1982. PhD, professor, PhD supervisor. Senior member of CCF. His main research interests include cryptography and security protocols, big data security, and privacy protection.

田有亮, 1982 年生. 博士, 教授, 博士生导师. CCF 高级会员. 主要研究方向为密码学与安全协议、大数据安全、隐私保护.

Wu Yingcai, born in 1983. PhD, professor, PhD supervisor. Senior member of CCF. His main research interest includes visual analytics.

巫英才, 1983 年生. 博士, 教授, 博士生导师. CCF 高级会员. 主要研究方向为可视分析.

Ji Shouling, born in 1986. PhD, professor, PhD supervisor. Senior member of CCF. His main research interests include data-driven security and privacy, AI security, and big data mining and analytics.

纪守领, 1986 年生. 博士, 教授, 博士生导师. CCF 高级会员. 主要研究方向为数据驱动安全和隐私、人工智能安全、大数据挖掘与分析.