

# Safety Classification Fine-tuning: 一种提高大模型输出内容安全性的微调方法

于 淼, 孙 磊, 胡翠云, 臧韦菲, 郭 松, 胡 鹏

中国人民解放军网络空间部队信息工程大学 密码工程学院 郑州 中国 450001

**摘要** 指令微调模型因其出色的指令理解和遵循能力在众多领域和任务中得到了广泛应用。然而这种能力也容易被恶意利用, 诱导模型生成有害内容。目前, 提高指令微调模型输出内容安全性的方法还存在一些不足, 如安全微调会破坏模型的有用性, 而且对越狱攻击的防御能力不足, 借助预训练的内容审核模型进行内容过滤会降低模型的响应速度。针对这些问题, 本文提出了一种新的微调方法 SCFT(Safety Classification Fine-tuning)。这一动机是我们发现, 指令微调模型之所以容易被滥用, 是因为它缺少判别“指令-回复”安全性的能力, 而模型最后一层解码器输出的隐藏状态中用来结束句子的 EOS 标记的词嵌入向量包含了整个句子的语义信息, 非常适合用来判别句子的安全性。但模型的基础结构决定了它不具备分类能力。因此, 我们在模型的输出层添加了一个新的分类头, 在针对通用能力进行指令微调的同时训练该分类头基于句子的语义信息进行以“安全”/“不安全”为标签的分类任务, 训练好的分类头就成为模型内部控制输出内容安全性的机制——“判别力机制”, 使得微调后的模型能够在推理时主动判断指令和回复的安全性, 并阻止不安全内容的输出。进一步分析发现, 借助“判别力机制”, SCFT 能够将模型有用性和安全性的训练目标统一起来, 实现两者之间的更好平衡, 还能使模型通用能力和安全能力的知识保持对称, 并将安全性的训练数据扩展到预训练的数据分布中, 增强模型安全能力的鲁棒性。实验结果表明, SCFT 是一种资源高效的端到端(end to end)安全微调方法, 它在不增加额外计算资源、不影响模型通用能力的前提下, 将微调后模型的有害输出率降低了约 91%, 平均有害分数从超过 4 分降至 1.36 分(5 分制, 分数越高模型越有害), 越狱攻击的有害率为 0%。

**关键词** 大型语言模型; 内容安全; 指令微调

中图法分类号 TP309.2 DOI 号 10.19363/J.cnki.cn10-1380/tn.2026.03.12

## Safety Classification Fine-tuning: A fine-tuning method to improve the output content safety of LLMs

YU Miao, SUN Lei, HU Cuiyun, ZANG Weifei, GUO Song, HU Peng

School of Cryptographic Engineering, PLA Information Engineering University, Zhengzhou 450001, China

**Abstract** Instruction-tuned models have been widely applied across various fields and tasks due to their excellent ability to understand and follow instructions. However, this capability is also prone to malicious exploitation, leading the model to generate harmful content. Current methods for enhancing the safety of the output content of instruction-tuned models still have some shortcomings, such as the safety-tuning that can undermine the model's helpfulness, and lack of robust defense against jailbreak attacks, and using pre-trained content moderation models for content filtering can slow down the model's response speed. In response to these challenges, this paper introduces a novel fine-tuning approach known as Safety Classification Fine-tuning (SCFT). The motivation for SCFT is the observation that instruction-tuned models are vulnerable to misuse due to their inability to assess the safety of "instruction-response" pairs. The embedding vector of the EOS token in the hidden state output of the model's final decoding layer, which is used to end sentences, contains the semantic information of the entire sentence and is very suitable to judge the safety of sentences. However, the fundamental structure of the model determines that it does not have classification capabilities. To address this, we have added a new classification head to the model's output layer. This head is trained to classify sentences as "safe" or "unsafe" based on sentences' semantic information, while the model is simultaneously instruction-tuned for general capabilities. The well-trained classification head acts as an internal "discrimination mechanism," controlling the safety of the model's output. This allows the fine-tuned model to actively judge the safety of "instruction-response" pairs during inference and prevent the output of unsafe content. Further analysis reveals that with the "discrimination mechanism", SCFT can unify the training objectives of the model's utility and safety, achieving a better balance between the two. It also maintains the symmetry of knowledge between the model's general capabilities and safety capabilities, and expanding the safety training data to the pre-training data distribution, enhancing the robustness of the model's safety capabilities. Experimental results demonstrate

通讯作者: 孙磊, 博士, 教授, Email: sl20210221@163.com。

本课题得到河南省自然科学基金(No. 242300420699)、军队科技委项目(No. JJK2023-449)资助。

收稿日期: 2024-07-21; 修改日期: 2024-11-08; 定稿日期: 2026-01-26

that SCFT is a resource-efficient, end-to-end safety-tuning method. It significantly reduces the Harmfulness Rate of the fine-tuned model by approximately 91%, lowers the average harmfulness score from over 4 points to 1.36 (on a scale of 5, the higher the score, the more harmful the model), and achieves a 0% harmfulness rate in jailbreak attacks, all without increasing additional computing resources or compromising the model's general capabilities.

**Key words** large language models; content safety; instruction fine-tuning

## 1 引言

2022 年底 ChatGPT 的诞生引发了大型语言模型 (Large Language Models, LLMs) 的研发热潮, 具有代表性的模型包括 OpenAI 的 GPT 系列<sup>[1-3]</sup>、Meta 的 Llama 系列<sup>[4-6]</sup>、Google 的 PaLM 系列<sup>[7]</sup>、智谱 AI 的 GLM 系列<sup>[8]</sup>、百川智能的 Baichuan 系列<sup>[9]</sup>、Alibaba 的 Qwen 系列<sup>[10]</sup>等。这些模型在海量的通用文本数据上以自监督学习 (Self-supervised Learning) 的方式进行了预训练 (Pre-training), 在实际应用时, 为了使预训练的语言模型 (Pre-trained Language Models, PLM) 在特定领域或任务上表现得更加出色, 通常需要在特定的数据上进行个性化的微调 (Fine-tuning)。指令微调 (Instruction Tuning, IT)<sup>[11]</sup> 是微调的主流范式, 是一个在由成对的“指令-输出”组成的数据集上以有监督的方式进一步训练模型的过程。这种数据形式使指令微调后的模型 (Instruction-tuned models) 具备了理解和遵循人类指令的能力, 不仅能够完成文本翻译、阅读理解、情感分析、代码生成等多种复杂的自然语言处理任务<sup>[12-15]</sup>, 对未知任务也具备了一定的泛化能力, 它们展现出的惊人的潜力和适应性使其在金融、法律、医疗等场景中均有应用落地<sup>[16-18]</sup>。

虽然指令微调可以显著提升大型语言模型的通用能力 (即有用性, 模型遵循指令解决人类请求的能力), 如自然语言理解和生成、知识问答、逻辑推理、代码生成等, 但这种强大的能力也是一把双刃剑。Bianchi 等人<sup>[19]</sup> 发现目前流行的几个以有用性为导向的指令微调模型<sup>[20-21]</sup> 是高度不安全的, 它们在面对恶意指令时都存在输出诸如不合法的提议、社会偏见、仇恨言论等有害内容<sup>[22-25]</sup> 的安全隐患。这种潜在的输出有害内容的不安全行为, 可能在指令微调模型的下游应用中造成严重的负面影响和危害, 甚至被恶意使用者利用和放大。随着指令微调模型应用范围的不断延伸, 解决输出中存在的 content 安全问题对于防止有害信息在社会中传播和保护用户免受潜在安全威胁至关重要。此外, 越来越多的开源 LLMs 权重、微调 API<sup>[26]</sup> 和数据集<sup>[27-28]</sup> 进一步降低了定制和使用这类模型的门槛, 解决这一安全问题变得越来越紧迫。

已有一些工作研究了如何提高指令微调模型

输出内容的安全性, 最常用的方法是基于预训练的内容审核分类模型过滤潜在不安全内容的输入或输出<sup>[29-30]</sup>。这种“事前预防”或“事后检查”的安全防护机制与指令微调模型自身是松耦合的, 而且内容过滤作为一个独立的过程会引入额外的时间开销, 影响用户与模型交互的体验感。此外, 训练一个这样的分类模型同样耗时且成本高昂, 虽然已有可以直接使用的内容审核 API 如 OpenAI Moderation API<sup>[31]</sup>、Perspective API<sup>[32]</sup>, 但在实际使用时又存在着对数据隐私保护的限制。

对于提升指令微调模型输出内容的安全性, 正如文献<sup>[19]</sup>所建议的那样: “如果在指令微调期间没有将安全性视为模型的支柱组件, 那么指令微调后的模型注定会造成伤害, 因为模型越好地遵循指令, 就越容易使其输出有害内容”。因此, 一个理想的办法是将安全集成到指令微调的过程中, 向模型内部嵌入安全规则, 确保输出内容有用且无害, 但这需要在指令微调的有用性和安全性之间进行权衡。因为有用性的训练目标是让模型更好地跟随用户的指令, 而安全性的训练目标是阻止模型跟随不安全的指令。Wei 等人<sup>[33]</sup> 将这种现象总结为模型通用能力和安全训练的优化目标不一致造成的竞争目标。

然而, 安全训练过的模型仍存在被滥用的风险<sup>[34-36]</sup>。Wei 等人<sup>[33]</sup> 发现, 即使是已经获得了安全防护机制的大模型依旧容易遭受越狱攻击 (Jailbreaking Attacks)<sup>[37-38]</sup>, 这些攻击试图通过精心设计的对抗性提示 (Adversarial Prompt) 来操纵 LLM, 使其绕过安全限制, 生成有害内容; Wei 等人<sup>[33]</sup> 进一步从理论层面分析了造成这种安全训练失效的原因, 除了竞争目标, 还有有用性和安全性的训练数据不对称造成的不匹配泛化, 具体来说, 安全性的训练数据一般是由成对的“恶意指令-与拒答相关的输出”构成的正向安全数据<sup>[19]</sup>, 而预训练是在比安全性更大、更多样化的数据集上完成的, 因此模型具有许多安全性训练没有覆盖到的能力。这种不匹配可以被用来构造越狱攻击的提示词 (Prompt), 对于这些提示词, 预训练和指令跟随能力可以泛化, 但模型的安全能力却不能。特别是由成对的“恶意指令-有害输出”对构成的不安全数据, Qi 等人<sup>[39]</sup> 还发现只需要使用少量 (几十条) 的有害数据进行恶意微调就能损害模型

已获得的安全性, 这表明不安全训练数据对模型的安全性有极大的负面影响, 但它们不可避免地存在于海量的训练数据中<sup>[40]</sup>。

基于上述研究, 本文进一步分析了模型为什么会被诱导输出不安全的内容, 根本原因是模型在生成文本时, 只是根据学习到的数据的统计模式进行推断, 无法判别输入的指令和生成的回答的安全性, 因此, 能否训练模型学会自主判别“输入的指令-生成的回答”的安全性, 并根据判别的结果决定要输出的内容来提高模型的安全性呢? 对此, 本文提出了一种新的微调方法(Safety Classification Fine-tuning, SCFT), 目的是在模型捕捉标签数据中特定于某一下游任务的模式和特点的同时, 向模型内部嵌入一种指导模型的输出的“判别力机制”, 经 SCFT 得到模型在响应用户请求的文本生成过程中, 由于“判别力机制”的存在能够主动判别这对“用户输入-模型生成的回复”的安全性, 只有被判别为安全的模型回复才能被输出, 否则输出拒答的警告。

本文的主要贡献包括以下几个方面:

- “判别力机制”。本文提出在模型的输出层插入一个新的分类头, 如图 1 所示, 利用最后一层解码器的输出(Last Hidden States)中表示句子结束的句尾标记(End of Sentence token, EOS token)的词嵌入向量(Word Embedding)中蕴含的整个“指令-输出”的语义信息, 在针对有用性进行微调的同时训练该分类头进行以“安全”/“不安全”为类别标签的分类任务。训练好的分类头能够基于 LLMs 的基础结构提取的“用户输入-模型生成的回复”的语义信息判别这对数据的安全类别标签, 成为嵌入模型内部的控制输出内容安全性的“判别力机制”。
- SCFT: 一种向大模型嵌入针对内容安全的“判别力机制”的带安全分类头的微调方法。本文提出的 SCFT 是一种资源高效的构建端到端(end to end)的大模型生成内容安全防护的方法: 不借助辅助模型, 仅需向 LLMs 中插入一个分类头作为“判别力机制”, 使安全规则融合进模型自身结构里; 不需要额外的训练, 在针对下游任务微调的同时训练模型学会判别内容的安全性; 不耗费额外的时间进行内容过滤, 将内容安全性审核与文本生成过程紧密耦合起来, 使模型输出层的结果具备安全保证。
- 理论分析 SCFT 在平衡模型有用性和安全性以及在增强安全机制的鲁棒性上的优势。在训练层面上, SCFT 借助“判别力机制”将安全性训

练的优化目标转变为分类头分类能力训练的优化目标, 转变后的优化目标能够和训练模型通用能力的优化目标拟合到一起, 有效解决了模型通用能力和安全能力训练之间竞争目标的问题, 实现了模型有用性和安全性之间更好的平衡。在数据层面上, SCFT 使“判别力机制”的训练数据与模型通用能力的训练数据保持对称, 还增加了有害数据, 进一步缩小了安全训练与指令微调 and 预训练之间数据分布的差异, 有效缓解了模型通用能力与安全能力之间不匹配的泛化, 增强安全防护的鲁棒性。

- 实验及分析。本文在正常使用中面临恶意指令和越狱攻击这两种典型的不安全使用指令微调模型的场景下评估 SCFT 的有效性, 实验结果表明: SCFT 可以在不损害模型处理下游任务的通用能力的情况下显著提高微调后模型的安全能力, 大幅降低模型在面临恶意指令时输出有害内容的安全风险, 对越狱攻击也具有一定的鲁棒性。

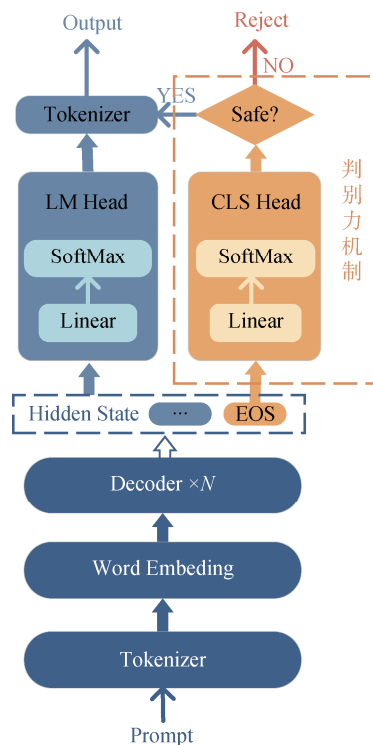


图 1 具有“判别力机制”的指令微调模型结构图  
Figure 1 Structure of instruction-tuned model with "discrimination mechanism"

(注: 模型的基础架构采用目前主流的仅解码器(Decoder Only)架构, 除了通常的语言模型头(language model head, LM Head)之外, 还为最后一层解码器的输出中表示句子结束的句尾标记的词嵌入向量提供了一个分类头。)

本文的组织结构如下: 第 1 节介绍了指令微调

模型存在的安全问题以及现有的提高指令微调模型输出内容安全性的方法中存在的不足;第 2 节介绍了有关指令微调、安全微调和内容过滤的知识;第 3 节介绍了本文提出的一种向大模型嵌入针对内容安全的“判别力机制”的微调方法,包括基本思路、理论分析、训练及推理流程、微调及文本生成的算法设计;第 4 节选择指令微调典型的应用场景进行实验设计,包括微调和评估使用的数据集、训练细节设置、微调后模型通用能力和安全性及评估方法与指标等;第 5 节对实验结果进行比较和分析;第 6 节对全文进行总结并提出一些未来工作的建议。

## 2 相关工作

大型语言模型是一种建立在 Transformer 架构<sup>[41]</sup>上的包含数百亿以上参数的深度神经网络,采用自然语言处理领域主流的预训练微调范式,预训练使模型获得了世界知识(World Knowledge)和续写的能力<sup>[42]</sup>,但此时的模型也只会续写,为了达到领域适应的目的,需要再针对特定下游任务进行微调。

指令微调专注于引导预训练得到的模型如何使用其学到的知识,然而训练语料库中往往存在着显式或隐式的有害文本<sup>[40]</sup>,LLMs 的基础结构决定了它也会学习和记忆这一部分知识,再加上指令微调后的模型具备了指令跟随的能力,它在能够理解输入的指令并生成特定任务所需的输出的同时,也会遵从恶意的指令并生成有害的内容。

随着以 LLMs 为基础底座的应用越来越普及,指令微调作为 LLMs 应用于下游任务时必不可少的一个环节,解决指令微调带来的安全问题刻不容缓,对于在实际应用场景下安全地部署指令微调模型有重要意义。LLMs 的安全性是指它们能够产生有用的响应,并且不会对用户和社会造成任何伤害,然而“安全”这一概念本身具有不确定性,自然语言的复杂性使得不同文化及个体对于“安全语言”的理解各不相同,因此本文将安全的范畴进一步明确,即 LLMs 的输出中不存在以下内容:

脏话侮辱(Insult)、偏见歧视(Unfairness and Discrimination)、违法犯罪(Crimes and Illegal Activities)、身体伤害(Physical Harm)、心理健康(Mental Health)、财产隐私(Privacy and Property)、道德伦理(Ethics and Morality)。

### 2.1 安全微调

指令微调通过训练模型的指令跟随能力能提高模型的有用性,但模型越好地听从指令,就越容易被用来生成有害内容。为了在提高 LLMs 指令遵循

能力的同时使它们保持安全和可靠,文献[19]研究了在指令微调时加入正向安全数据对微调后模型安全性的影响,他们发现在对 LLaMA 等模型进行微调时,在训练集中只添加 3%的正向安全数据,就可以显著提高微调后模型的安全性,虽然作者等人没有发现模型的通用能力有明显的下降,但他们确信存在一个使对安全性的训练损害模型有用性的点。因为作者等人还观察到:过多的正向安全数据可能会导致模型出现“夸大的安全”的问题,即在面对与不安全指令有字面上的相似之处的伪恶意指令时(即虚警情况),模型拒绝回答了这些安全的指令。这表明模型在有用性和安全性之间保持平衡的复杂性。因此,指令微调时需要重点考虑如何平衡模型的通用能力和安全能力。

然而,在正向安全数据上进行过安全微调的模型在面对越狱攻击时仍具有脆弱性。文献[33]对最先进的安全训练模型(包括 OpenAI 的 GPT-4 和 Anthropic 的 Claude)使用涵盖 17 种不安全场景的有害指令和 30 种越狱方法的组合进行从简单到复杂的越狱攻击,如前缀注入、Base64 编码、角色扮演及其组合,攻击的成功率高达 96%,尤其是简单攻击的组合会产生最强的越狱攻击。这表明现有安全训练为模型嵌入的安全机制的鲁棒性仍旧不足<sup>[43-44]</sup>。

此外, Qi 等人<sup>[39]</sup>发现,仅使用少量(几十条)的有害数据进行恶意微调,就能显著损害模型已经通过使用正向安全数据进行安全性训练获得的安全能力。他们还尝试在有害数据中加入文献[19]提供的正向安全数据进行微调,发现虽然比直接使用有害数据微调出来的模型有害程度降低了,但还是远不如模型原本的安全性。这进一步表明了有害数据对模型安全性的影响大于正向安全数据。

Wei 等人<sup>[33]</sup>从理论出发分析了造成上述现象的原因:一是训练层面上有用性和安全性训练之间的竞争目标,二是数据层面上预训练和安全能力之间不匹配的泛化,并通过以这两个原因为原则构造的越狱攻击的指令成功使已经通过正向安全数据进行安全训练过的模型输出有害内容来验证了他们的理论分析。

### 2.2 内容过滤

为了在不影响模型通用能力的情况下阻止它输出有害内容,ChatBot 场景下针对有毒、粗鲁、敌对性或隐性仇恨内容的潜在有害成分检测<sup>[45]</sup>被引入 LLMs 中,一般是使用一个训练好的高质量的内容过滤器如 Perspective API<sup>[32]</sup>、AI2 Delphi 对模型的输出进行检测和筛选。然而它们的过滤范围对于大模型

来说是远远不够的,因此 OpenAI 针对 LLMs 场景中可能出现的不安全内容,进一步训练了一个内容审核模型 OpenAI Moderation<sup>[31]</sup>,并以 API 的形式开放。

除了过滤掉不安全的回复外,Markov 等人<sup>[30]</sup>和 Lin 等人<sup>[46]</sup>对用户输入的内容进行分类,拒绝不安全的指令。鉴于 LLMs 本身就具备了强大的语言能力,有工作将内容审核任务与微调结合起来,训练 LLMs 来增强内容审核任务的鲁棒性和可解释性,例如在精心收集的数据集上微调 Llama2 得到的 Llama Guard<sup>[47]</sup>就是专门用来检测模型的输入输出中不安全内容的;OpenAI 使用与 GPT-4 的交互来优化现有的内容审核系统<sup>[48]</sup>。

虽然这种策略可以减轻模型遵循恶意指令后输出有害内容的安全风险,但对于大多数公司和组织来说,出于安全和隐私的考虑,可能存在限制使用第三方 API 进行内容审核的法规和要求;而自主训练或微调一个高质量的内容过滤器的成本又很高,需要大量的数据、时间和计算资源。此外,由于内容过滤器是独立于 LLMs 之外的,对模型输出的安全控制和模型的生成过程是分离的,无论先检测再生成还是先生成再检测,都会降低与用户交互过程中模型的响应速度,影响用户的体验感。

### 3 SCFT

鉴于现有的提高指令微调模型安全性的方法仍存在一些局限性:利用正向安全数据直接微调模型虽然能获得端到端的安全能力但难以兼顾模型的通用能力且鲁棒性不足,借助外部工具进行内容过滤又会带来额外的时间开销;本文的目标是在不借助外部模型、不需要额外的训练、不增加交互中响应时间的条件下开发有用且无害的指令微调模型。

本节先对指令微调过程中出现安全问题的原因进行分析并提出本文的应对策略 SCFT:一种向大模型嵌入针对内容安全的“判别力机制”的微调方法,然后对所提出的方法进行理论分析和建模,最后对本文提出的 SCFT 的训练和推理流程进行详细的描述。

#### 3.1 SCFT 的基本思路

微调本质上是一个基于预测响应与目标响应之间的一致性来优化模型参数的过程。指令微调使 LLMs 对指令的预测输出尽可能地靠近指令的目标输出,微调后的模型从由成对的“指令-输出”构成的数据集中学习到了遵循指令的期望输出,获得了理解和遵循指令的能力。因此,训练数据对指令微调模型有非常大的影响,虽然正向安全数据可以引导模型在面对恶意指令时倾向于拒答,但很难保证微

调的数据中所有的目标响应都是安全的,如果微调的数据中存在有害的输出,那么势必会导致模型参数朝有害空间优化,微调后的模型学习到了不安全的行为,容易被恶意指令诱导生成不安全的内容。而且预训练的语料中本身就存在着不安全的内容,利用指令跟随能力也可以让模型输出其在预训练期间学习到的不安全知识。

实际上,本文认为指令微调模型之所以会被诱导输出有害内容,是因为它没有识别不安全的“指令-生成的响应”并阻止有害响应输出的能力。因此,本文提出了 SCFT:一种向大模型嵌入针对内容安全的“判别力机制”的微调方法。SCFT 的目标是借助训练数据中的有害输出及其指令,让模型从这部分数据中学习有害数据的语义特征,进而能够基于语义信息分辨数据是否有害,而不是将有害输出作为模型参数的优化目标,使模型将这部分数据作为遵循指令的期望输出。

具体来说,SCFT 在训练模型指令跟随能力的同时让模型根据“指令-输出”整体的语义信息来预测这条数据的安全类别标签,将真实的安全类别标签作为模型参数的优化目标,使模型具备对“指令-输出”的安全类别标签进行分类的能力,即针对内容安全性的“判别力”。

虽然模型变大后出现了许多涌现能力,但模型的架构决定了 LLMs 本质上仍然是对下一个词进行预测(Next Token Prediction),不具备分类能力。对此,本文提出在模型的输出层插入一个新的分类头,它的任务是对 LLMs 基础结构提取出语义信息进行以安全或不安全为标签的分类任务。而 LLMs 是典型的自回归语言模型,即根据上文内容预测下一个可能跟随的单词,训练时 Decoder Block 层会并行地输出所有预测的下一个词(Token)的词嵌入向量,其中掩码机制(Attention Mask)使当前需要预测的词(Token)无法看到后续的词,这使得用来表示句子结束的 EOS Token 的词嵌入向量融合了句子中各个词的语义信息,而且它本身并没有特定的语义,非常适合用来训练分类头进行基于语义信息的文本安全性分类任务,训练好的分类头作为嵌入在模型内部的“判别力机制”控制输出内容的安全性。

通过上述 SCFT 得到的指令微调模型不仅能适配特定的下游任务,由于“判别力机制”的存在,还具备了判别“指令-输出”数据对安全性的能力。微调后的模型在开放给用户进行问答交互时,“判别力机制”会主动判别由用户输入的指令和模型生成的回复组成的文本对的安全标签,只有标签为安全的

文本对中的模型生成的回复才会被输出, 对于标签为不安全的文本对中的用户输入的命令, 模型会输出拒答的安全警告, 最终获得一个能够通过自主审核“指令-输出”对数据的安全性来确保输出内容安全性的指令微调模型。微调后的模型在开放给用户进行微调时, “判别力机制”会主动判别用户上传的成对的“指令-输出”数据的安全标签, 只有标签为安全的训练样本才能用于更新模型的参数。

### 3.2 SCFT 的有效性分析

SCFT 是一种新的在预训练模型中嵌入安全规则的解决方案, 能够在模型通用能力和安全性之间实现更好的权衡, 增强模型对越狱攻击的鲁棒性: 一是在训练层面上, 借助“判别力机制”统一训练过程中通用能力和安全性的优化目标, 进而在不显著降低通用性能的情况下提高指令微调模型的安全性。二是在数据层面上, 借助“判别力机制”统一训练过程中通用能力和安全性的训练数据, 并将安全性的训练数据扩展到预训练的数据空间中, 减轻数据不对称带来的不匹配泛化造成的指令微调模型安全性的脆弱性。

#### 3.2.1 训练层面

先前针对指令微调模型安全性的工作主要是训练模型不跟随恶意的指令, 而这与模型有用性的训练目标是冲突的, 因为有用性的目标让模型听从指令, 因而需要根据实际应用需求的优先级在安全性和有用性之间进行权衡。然而, 借助“判别力机制”, SCFT 能够:

- 转变安全防护策略: 将让模型不跟随恶意的指令转变为主动检测“用户指令-生成的内容”的安全性并限制有害内容的输出;
- 转变安全能力的训练目标: 将安全能力的训练转变为“判别力机制”的训练;
- 转变安全性的优化目标: 将安全性的优化目标转变为分类头分类能力的优化目标;

在训练时, 转变后的安全性的优化目标可以与模型有用性的优化目标统一起来, 有效解决了模型通用能力和安全能力训练之间的竞争目标, 一次微调可以实现两个目标。

#### 3.2.2 数据层面

先前针对指令微调模型安全性的工作主要是利用专为安全任务构造的正向安全数据, 也就是说, 模型有用性和安全性的训练数据是相互独立的, 而且训练有用性的数据范围和数量远大于训练安全性的数据, 这带来了不匹配泛化的问题, 因为安全性无法覆盖到有用性训练获得的全部通用能力上, 指

令微调模型的有用性可以进一步泛化但安全性却不能, 这种不匹配的泛化可以被用来构造越狱攻击的对抗性指令, 诱导模型输出不安全的内容。

然而 SCFT 中“判别力机制”是在训练模型通用能力的同时训练的, 二者使用了相同的训练数据, 这能够将指令微调模型的通用能力和安全能力的知识统一起来, 缓解训练数据不对称带来的不匹配泛化问题。此外, SCFT 还在训练“判别力机制”时使用了由有害输出及其指令构成的有害数据集, 而不是正向安全数据集, 因为有害数据虽然不存在于模型通用能力的训练数据中, 但它们存在于预训练广泛的数据分布中, 借助有害数据可以使安全性是在比有用性更大的范围内训练的, 将指令微调模型安全能力的知识范围扩大到预训练的数据分布中, 进一步缓解预训练与安全性训练之间不匹配泛化的问题, 增强微调后模型在面临针对不匹配泛化构造的对抗性指令时安全防护的鲁棒性。

### 3.3 问题公式化

SCFT 使用的指令微调数据集可以表示为

$$D = \{(X_i, Y_i, T_i)\}_{i=1}^n, \text{ 其中:}$$

- $X_i = \{x_1, \dots, x_m\}$  是用户输入的提示词 (Prompt), 包括问题、闲聊对话、任务指令等多种形式和任务, 用长度为  $m$  的词(Token)序列表示;
- $Y_i = \{y_1, \dots, y_k\}$  是遵循输入的期望的模型响应(Response), 用长度为  $k$  的词(Token)序列表示;
- $T_i \in \{0, 1\}$  是这条“Prompt-Response”对应的真实的安全类别标签(SafeTag), 标签的设置原则如下: 只要 Response 的内容是安全的, 就认为这条数据是安全的, 令  $T_i = 1$ , 否则令  $T_i = 0$ ;
- $n$  是数据量。

在数据集  $D$  上对 LLMs 进行 SCFT 的优化目标表示如下:

$$\begin{aligned} \text{Loss}_{\text{train}} &= \gamma * \text{Loss}_{\text{lm\_head}} + \text{Loss}_{\text{cls\_head}} \\ \theta^* &= \arg \min_{\theta} \text{Loss}_{\text{train}} \end{aligned}$$

其中,  $\theta$  是模型的原始参数,  $\theta^*$  是微调后的模型参数,  $\gamma$  用来控制哪些参数可以被更新, 优化目标  $\text{Loss}_{\text{train}}$  包含两部分损失:  $\text{Loss}_{\text{lm\_head}}$  和  $\text{Loss}_{\text{cls\_head}}$ , 分别对应模型输出层的两个任务头:

- a) 语言模型头(LM Head): 使用标准因果语言模型 (Causal Language Model) 优化训练样本中  $(X_i, Y_i)$  的最大似然估计, 用负对数似然损失表示

$(X_i, Y_i)$  的最大似然估计。

$$\text{Loss}_{\text{lm\_head}} = -\sum_{i=1}^n \sum_{j=1}^k \log P(y_i | y_{<j}, X_i; \theta)$$

b) 分类头(CLS Head): 基于 LLMs 基础结构提取出的  $(X_i, Y_i)$  的语义信息对安全类别标签(SafeTag) 进行分类, 其优化目标  $\text{Loss}_{\text{cls\_head}}$  可以看作优化  $(X_i, Y_i)$  真实的安全标签  $T_i$  和预测的安全标签  $\hat{T}_i$  的交叉熵损失, 公式表示如下:

$$\hat{T}_i = P(T_i | X_i, Y_i; \theta)$$

$$\text{Loss}_{\text{cls\_head}} = -\sum_{i=1}^n T_i \log \hat{T}_i + (1 - T_i) \log (1 - \hat{T}_i)$$

其中,  $\gamma = (T_1, T_2, \dots, T_n)$  用来控制语言模型头的参数是否被更新, 因为对于数据集  $D$ , 模型应该仅从

真实的安全类别标签(SafeTag)为安全的数据中学习遵循指令的期望输出, 也就是说, 安全类别标签为不安全(即  $T_i = 0$ )的数据不能被用于更新语言模型头的参数, 安全类别标签为安全(即  $T_i = 1$ )的数据可以被用于更新全部的参数。

### 3.4 SCFT 训练流程

使用本文提出的 SCFT 对预训练语言模型进行微调的过程如图 2 所示, 以一条训练样本“Prompt-Response-SafeTag”为例, 首先对训练样本进行数据预处理(①), 然后输入模型, 经嵌入层(②)、模型 Blocks(③)、输出层(④)运算后得到模型预测的响应和预测的安全类别标签, 再根据目标响应和真实安全类别标签计算总的训练损失(⑤), 最后通过反向传播更新模型权重。

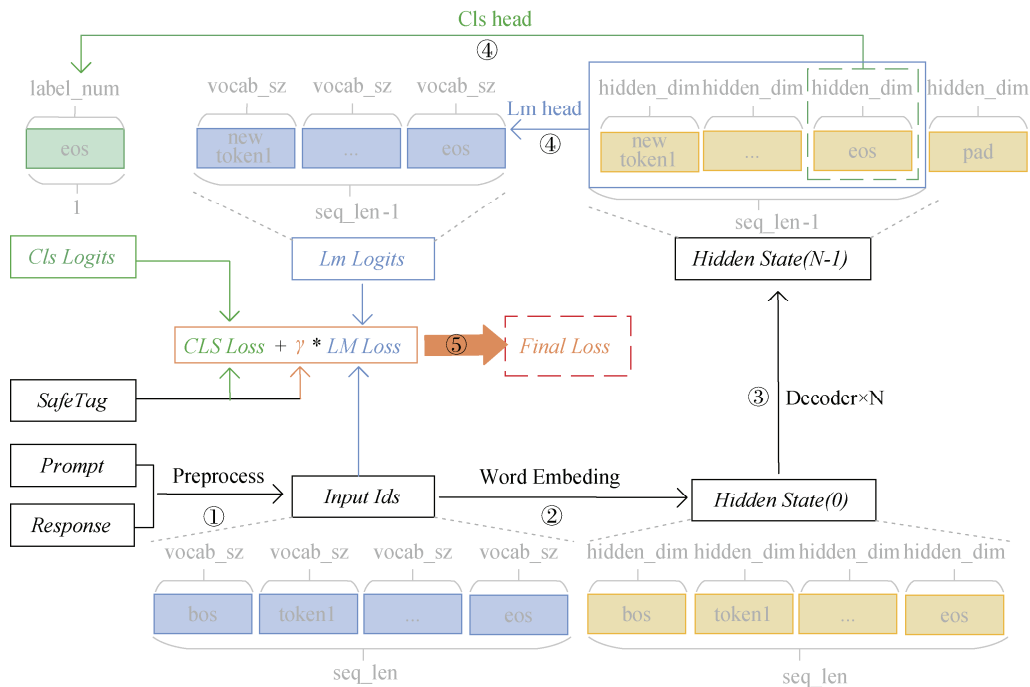


图 2 训练流程图

Figure 2 Process of training

下面对步骤①~⑤进行详细描述:

①数据预处理: 首先将输入 Prompt 和遵循输入的期望输出 Response 按一定的格式拼接成的文本序列; 再利用分词器(Tokenizer)将拼接的文本序列切分成更细粒度的单词或字符, 分解为多个词(Token)组成的序列, 为了将文本映射成模型可理解的输入形式, 将每个词(Token)都转换为词汇表(Vocabulary)中与之对应的整数索引(Index); 最后使用特殊的记号标记句子的开始和结束以确保序列长度得到正确填充。

②嵌入层: 首先获取与整数索引(Index)对应的

词嵌入向量(Embedding Vector), 然后将每个词(Token)映射为一个实数向量, 得到文本的词嵌入向量序列。

③模型 Blocks: 嵌入层的输出被送入由  $N$  个 Decoder 模块堆叠而成的自回归核心, 每层 Decoder 接收上一层输出的隐藏状态(Hidden State), 并行的生成该层的每个词嵌入向量, 在自注意力机制的作用下, 每一层的每个词嵌入向量都会包含之前位置词嵌入向量的信息, 每一层的隐藏状态都包含了之前层的上下文信息, 因此, 最后一层输出的隐藏状态中与表示句子结束的句尾标记(EOS token)对应的词嵌入向量包含了所有层和所有词丰富的语义信息;

④输出层: 输出层包含两个任务头, 分别是语言模型头(LM Head)和分类头(CLS Head)。

a) 语言模型头(LM Head)是模型架构的最后一层, 负责生成语言模型的输出, 通常是线性变换加上 Softmax 函数, 它接收模型 Blocks 的输出作为输入, Linear 层将其映射到与词汇表大小相同的向量空间, 再经过 Softmax 函数获得归一化的生成文本的概率分布, 其中每个词的概率表示在给定前文条件下该词是下一个出现的可能性。

b) 分类头(CLS Head)是向模型中插入的“判别力机制”, 同样位于模型结构的最后一层, 负责对文本的安全标签进行分类, 由一个线性层和一个 Softmax 函数组成, 它接收模型 Blocks 的输出中 EOS Token 所在位置对应的词嵌入向量作为输入, Linear 层将其映射到与安全标签类别大小相同的向量空间即一个二维的向量空间, 再经过 Softmax 函数将其映射到[0, 1]区间, 给出每种类别的概率。

⑤计算训练损失并通过反向传播更新模型权重: CLS Head 与 LM Head 互不干扰但共享模型基础结构的参数以充分利用 LLMs 固有的知识和能力, 总的训练损失(Final Loss)由分类头的损失(CLS Loss)和语言模型头的损失(LM Loss)加权得到, 其中系数  $\gamma$  用来控制语言模型头的训练, 其值与真实的安全类别标签(SafeTag)相关, 标签为“不安全”的训练样本不能用来更新语言模型头的参数, 即令  $\gamma=0$ , 最后通过反向传播更新模型权重。

a) 语言模型头(LM Head)的训练使用交叉熵损失函数(Cross-Entropy Loss Function)来比较模型输出的概率分布与实际的下一个词的标签, 通常将对应提示词(Prompt)的损失置为零, 只使用响应(Response)上的损失;

b) 分类头(CLS Head)的训练可以看作优化模型预测的安全分类标签与真实的安全分类标签的交叉熵损失函数;

以上述训练过程描述为基础, 下面进一步对 SCFT 的实现进行了更为详细的设计。SCFT 的训练过程主要包括数据预处理和模型训练两个部分。首先定义数据预处理函数:

$D' \leftarrow \text{Preprocess}(D, \text{Format}, \text{Tokenizer})$ : 表示对于数据集  $D$  中的每一条训练样本  $(X, Y, T)$ , 通过拼接“提示词-响应”的模板 Format、分词器 Tokenizer 以及表示句子开始(BOS)和结束(EOS)的特殊标记, 将  $(X, Y)$  转换为整数索引序列:

$$\text{InputIds} = [\text{BOS}_{\text{id}}, \text{Tokenizer}(\text{Format}, X, Y), \text{EOS}_{\text{id}}]$$

并构造对应的真实标签:

$$\text{Label} = [\text{Tokenizer}(\text{Format}, X, Y), \text{EOS}_{\text{id}}],$$

和标签掩码  $\text{Mask}_{\text{target}}$ ,  $\text{Mask}_{\text{target}}$  的形状和 Label 相同,  $\text{Mask}_{\text{target}}$  中与 Label 中  $Y$  对应的位置的值为 1, 其余位置的值为 0; 最终得到一个由  $n$  个  $(\text{InputIds}, \text{Label}, T, \text{Mask}_{\text{target}})$  构成的新集合  $D'$ 。

下面对模型训练的算法进行设计。

**算法 1.**带安全分类头的指令微调算法。

定义训练总轮数为  $E$ ; 预训练模型的权重为  $\theta = \{W_{\text{embedding}}, W_0, \dots, W_{N-1}, W_{\text{lm\_head}}, W_{\text{cls\_head}}\}$ , 其中嵌入层的权重为  $W_{\text{embedding}}$ , 模型 Blocks 中第  $i$  层的权重  $W_i$ ,  $W_{\text{lm\_head}}$  是语言模型头的权重,  $W_{\text{cls\_head}}$  是分类头的权重; 训练数据集为  $D = \{(X_i, Y_i, T_i)\}_{i=1}^n$ ; 用于拼接提示词和响应的模板为 Format; Tokenizer 是文本序列的分词器;  $\text{BOS}_{\text{id}}$  是表示句子开始的特殊标记的整数索引;  $\text{EOS}_{\text{id}}$  是表示句子结束的特殊标记的整数索引;  $S$  是 SoftMax 函数; CE 是交叉熵损失函数;  $\alpha$  为学习率; 指令微调后模型的权重为  $\theta^* = \{W_{\text{embedding}}^*, W_0^*, \dots, W_{N-1}^*, W_{\text{lm\_head}}^*, W_{\text{cls\_head}}^*\}$ 。

输入:  $\theta = \{W_{\text{embedding}}, W_{0 \sim N-1}, W_{\text{lm\_head}}, W_{\text{cls\_head}}\}$ ,

$E, D = \{(X_i, Y_i, T_i)\}_{i=1}^n, \text{Format}, \text{Tokenizer}, \alpha$ 。

输出:  $\theta^* = \{W_{\text{embedding}}^*, W_{0 \sim N-1}^*, W_{\text{lm\_head}}^*, W_{\text{cls\_head}}^*\}$ 。

1  $D' \leftarrow \text{Preprocess}(D, \text{Format}, \text{Tokenizer})$ ;

2 **FOR**  $e = 1, \dots, E$ :

3 **FOR**  $i = 1, \dots, n$ :

4  $\text{HS}_i^0 \leftarrow W_{\text{embedding}}(\text{InputIds}_i)$

5  $\text{HS}_i^{N-1} \leftarrow W_{N-1}(W_{\dots}(W_0(\text{HS}_i^0)))$

6  $\text{Logits}_{\text{lm}}^i \leftarrow S(\text{HS}_i^{N-1} * W_{\text{lm\_head}})$

7  $\text{Logits}_{\text{cls}}^i \leftarrow S(\text{HS}_i^{N-1}[\text{index}_{\text{EOS}}] * W_{\text{cls\_head}})$

8  $\text{Loss}_{\text{cls}}^i \leftarrow \text{CE}(\text{Logits}_{\text{cls}}^i, T_i)$

9

$\text{Loss}_{\text{lm}}^i \leftarrow \text{Mask}_{\text{target}}^i * \text{CE}(\text{Logits}_{\text{lm}}^i, \text{Label}_i)$

10  $\text{Loss}_{\text{train}}^i \leftarrow T_i * \text{Loss}_{\text{lm}}^i + \text{Loss}_{\text{cls}}^i, T_i \in \{0, 1\}$

11  $\theta \leftarrow \theta - \alpha \cdot \nabla_{\theta} \text{Loss}_{\text{train}}^i$

12 **END FOR**

13  $\theta^* \leftarrow \theta$

## 14 END FOR

## 3.4 SCFT 推理流程

经 3.3 节 SCFT 微调得到的有“判别力机制”指令微调模型的推理过程如图 3 所示, 首先对用户输入进行数据预处理(①), 然后输入模型, 经嵌入层(②)、模型 Blocks(③)、输出层(④)得到模型判别的“用户输入-模型响应”的安全类别标签, 根据安全类别标签决定是否输出模型生成的内容(⑤~⑥)。

SCFT 的推理过程主要包括数据预处理和模型推理两个部分。推理时的数据预处理只需要通过与训练时相同的模板 Format、分词器 Tokenizer 以及表示句子开始(BOS)的特殊标记, 将用户的输入 Question 转换为整数索引序列:

$$\text{InputIds} = [\text{BOS}_{\text{id}}, \text{Tokenizer}(\text{Format}, \text{Question})]$$

模型推理的过程与训练时类似, 不同的是训练时输出层是一次性输出全部的预测序列, 而在推理时是从左到右逐个生成每一个词, 每次在生成下一个 Token 的时候, 需要使用前面已生成的内容作为上下文来帮助预测; 对于生成任务来说, EOS token 代表着生成内容的结束, 因此 EOS token 的词嵌入向量包含了用户输入的 Prompt 和模型生成的 Response 的语义信息, 分类头基于这一语义信息判别“用户输入-模型响应”的安全类别标签, 模型根据安全类别标签控制其输出的内容。

基于上述对推理过程的描述, 下面对 SCFT 得到的指令微调模型的推理算法进行设计。

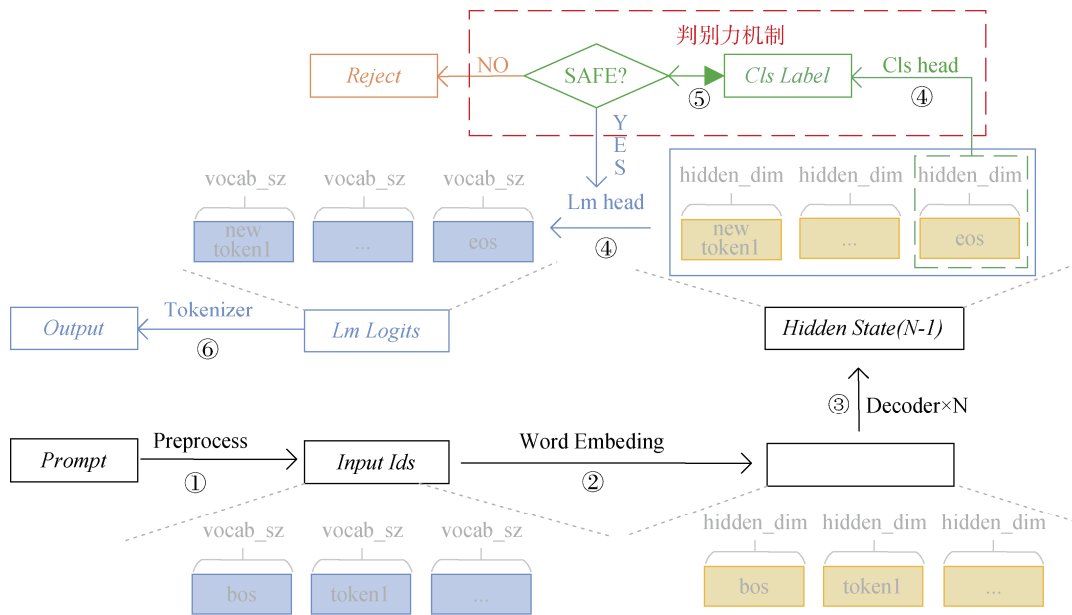


图 3 推理流程图

Figure 3 Process of inference

**算法 2.**带安全分类头的指令微调模型推理算法。

定义带安全分类头的指令微调模型的权重为

$$\theta^* = \{W_{\text{embedding}}^*, W_0^*, \dots, W_{N-1}^*, W_{\text{lm\_head}}^*, W_{\text{cls\_head}}^*\},$$

其中嵌入层的权重为  $W_{\text{embedding}}^*$ , 模型 Blocks 中第层的权重  $W_i^*$ ,  $W_{\text{lm\_head}}^*$  是语言模型头的权重,  $W_{\text{cls\_head}}^*$  是分类头的权重; 用户的输入为  $Q$ 。

输入:  $Q$ , Format, Tokenizer,

$$\theta^* = \{W_{\text{embedding}}^*, W_0^*, \dots, W_{N-1}^*, W_{\text{lm\_head}}^*, W_{\text{cls\_head}}^*\}.$$

输出: 模型的输出 Output。

1  $\text{InputIds} = [\text{BOS}_{\text{id}}, \text{Tokenizer}(\text{Format}, Q)];$

2  $\text{HS}^0 \leftarrow W_{\text{embedding}}^*(\text{InputIds});$

3  $\text{HS}^{N-1} \leftarrow W_{N-1}^*(W_{\dots}^*(W_0^*(\text{HS}^0)));$

4  $\text{NextTokenLogits} \leftarrow S(\text{HS}^{N-1} * W_{\text{lm\_head}}^*)[-1];$

5  $\text{NextTokenId} \leftarrow \text{Decoding}(\text{NextTokenLogits});$

6 IF NextTokenId 不是  $\text{EOS}_{\text{id}}$ :

7 将 NextTokenId 添加到 InputIds 末尾;

8 重复 2~6;

9  $\text{Logits}_{\text{cls}} \leftarrow S(\text{HS}^{N-1}[-1] * W_{\text{cls\_head}}^*)$

10  $\hat{T} \leftarrow \arg \max(\text{Logits}_{\text{cls}})$

11 IF  $\hat{T}$  是 SAFE:

12  $\text{Output} \leftarrow \text{Tokenizer}(\text{InputIds})$

13  $\text{Output} \leftarrow \text{Sorry, as an AI language model, I}$

can't offer any illegal or harmful advice and guidance. We should observe moral and legal norms in order to build a healthy, just and stable society. Can I help you with something else

## 4 实验

许多实际的应用场景中, 需要 LLMs 可以进行多轮交互, 像人类一样提供帮助<sup>[49]</sup>。指令微调经常被用来让 LLMs 具备连续多轮对话的能力, 因此本文也选择连续多轮通用对话作为所需的模型通用能力, 使用本文提出的 SCFT 微调开源的预训练语言模型。基于微调后的模型在给定提示词下做出的响应, 本文使用人工标注和基于 LLMs 反馈的自动化评估两种方式来研究微调后模型的有效性和安全性。

### 4.1 SCFT 微调

#### 4.1.1 微调数据集

本文提出的 SCFT 在微调预训练模型时使用的数据集是由具有代表性的多轮对话格式的通用能力训练数据和有害数据构成的混合数据集, 下面对其来源进行说明:

1) 通用能力训练数据集: Anthropic Helpful Base 对话数据集<sup>[20]</sup>。该数据集分为训练集和测试集两个部分, 训练集包含 43105 个人类与基础语言模型 (52B context-distilled LM) 之间的多轮对话, 由人类标注员让模型去做问答、写作、制订计划等文本任务, 在与基础语言模型的每一轮对话中, 提供给人类标注员两种可能的回复, 最终得到的每个多轮对话都包含一对文本: “chosen-rejected”, “chosen” 表示人类标注员认为更有帮助性的回复。本文选择标签为“chosen”的多轮对话作为指令微调的训练数据。此外, 本文进一步清洗了这些数据, 通过敏感词匹配<sup>[50]</sup>从中删除了 1980 个与安全相关的训练样本, 得到了 41855 条仅考虑有用性的多轮对话数据, 这种清洗防止了在构建微调数据集期间引入了任何的安全防护, 确保微调后模型的安全性是通过本文提出的新的微调方法获得的。

2) 有害数据集: Anthropic Red Team 数据集<sup>[51]</sup>。它由 38,961 个包含有害内容的 QA 对组成, 问题由人类编写, 答案由不同的 LLMs 生成, 包括但不限于歧视性语言和对虐待、暴力、自残、剥削和其他可能不安全主题的讨论。本文根据官方提供的每条数据的有害值, 从中选择了选 10000 条最有害的数据作为安全分类头的训练的数据。

#### 4.1.2 指令微调模型

为了证明 SCFT 可以在不损害模型通用能力的

情况下显著提高微调后模型的安全性, 本文基于相同的预训练模型训练了 3 个不同的指令微调模型: 首先使用本文提出的 SCFT 在通用能力训练数据和有害数据上得到模型  $SCFT_{+(Helpful, Harmful)}$ ; 然后使用以有用性为导向的普通指令微调<sup>[11]</sup>在通用能力训练数据和有害数据上得到模型  $FT_{+(Helpful, Harmful)}$ ; 最后仅在有用性数据上使用以有用性为导向的普通指令微调<sup>[11]</sup>得到模型  $FT_{+(Helpful)}$ 。

同时, 考虑使用正向安全数据进行安全微调的方法<sup>[19]</sup>, 借助 GPT4 将有害数据转化为正向安全数据, 遵循 Bianchi 等人<sup>[19]</sup>的数据配比, 在有用性数据中分别加入 3% 和全部的正向安全数据并进行普通指令微调<sup>[11]</sup>, 分别得到模型 Bianchi 等人<sup>[19]</sup>+3% 和 Bianchi 等人<sup>[19]</sup>+ALL。

上述指令微调模型的训练均为全量参数微调。

#### 4.1.3 训练细节

训练的代码基于 PyTorch<sup>[52]</sup> 和 HuggingFace Transformers<sup>[53]</sup> 实现, 为了更公平地对比, 本文选用在大模型领域的学术研究中普遍使用的 Llama2-7B-base<sup>[5]</sup> 作为基座模型, 在 1 台 4 卡 L20 服务器上使用 DeepSpeed ZeRO-2<sup>[54]</sup> 分布式训练策略进行全量参数微调, 所有指令微调模型的训练参数保持一致: 设置梯度积累为 10 训练 2 轮 (批处理大小为 1, 输入序列长度最大值为 3072), 使用 AdamW 优化器<sup>[55]</sup>, WarmupDecayLR 学习率调整策略, 最大学习率为  $1e-4$ , 最小学习率为  $1e-5$ 。

## 4.2 评估

评估的目的是研究不同的微调方法对指令微调模型通用能力和安全性的影响。通过微调后的模型在面对恶意指令时的输出来评估模型的安全性, 通过微调后的模型在处理特定下游任务时的响应质量来评估模型的通用能力。

### 4.2.1 评估数据集

1) 安全性评估数据集: 本文选择了目前针对指令微调模型安全性的研究中具有代表性的评估数据集, 包括: ① Bianchi 等人<sup>[19]</sup>使用 GPT-3 创建的用于测试模型如何响应恶意和有害的指令的 Malicious 数据集; ② Qi 等人<sup>[39]</sup>基于 Meta 和 OpenAI 的使用策略构建的有害指令数据集 I-Harmful; ③ 从 Anthropic Red Team<sup>[51]</sup> 数据集中抽取的 400 条红队评估数据集 Red Teaming, 该数据集不与微调使用的数据重叠。④ Wei 等人<sup>[33]</sup>提出的以竞争目标和不匹配的泛化为原则构造的旨在欺骗模型给出有害的响应的越狱指令数据集 Jailbroken。

2) 模型通用能力评估数据集: ①使用 MT-bench<sup>[56]</sup>评估模型的多轮对话和指令遵循能力, 这是由 80 个高质量的多轮对话问题组成的测试基准。②从 Anthropic Helpful Base 对话数据集<sup>[20]</sup>的测试集中抽取 100 条作为模型通用能力的评估数据集。

#### 4.2.2 评估方法和指标

模型通用能力评估: 本文遵循 MT-bench 基准官方的评估方法和指标<sup>[56]</sup>, 给定问题以及模型的答案, 利用 GPT4 按照评分标准和评分区间进行评分, 给出一个在 1~10 的有用性分数, 数值越高表示模型的答案对问题越有用, 在分别得到每个问题的评分结果之后对分数进行汇总, 计算模型在整个测试集上平均得分(Helpfulness Score)。

安全性评估: 本文遵循 Qi 等人<sup>[39]</sup>的方法, 采用基于 LLMs 反馈的自动评估方法, 给定有害指令和模型的回复, 让 GPT4 按照评分标准和评分区间给出一个在 1~5 的有害分数, 数值越高表示模型回复的危害越大, 计算微调后的模型在整个评估数据集上的平均有害性评分(Harmfulness Score)以及获得最高有害分数 5 分的数据占比(Harmfulness Rate)。

人工标注(Response Manual Annotation): 为了保证利用 GPT4 自动化评估的结果的准确性, 本文雇佣了两位高质量人工标注员对 SCFT 和一般的指令微调得到的模型的输出进行模型通用能力和安全性的偏好实验。本文从评估模型能力的 Helpful 测试集以及评估安全性的有害和恶意指令数据集 Red Teaming、越狱指令数据集 Jailbroken 中各挑选 50 条测试样例, 遵循文献[19]的方法, 对于每条指令和来自两个模型的一对响应, 向人工标注员隐藏了响应是由哪个模型生成, 并将呈现的顺序随机打乱以减少可能的偏见, 按照以下规则进行人工标注: ①两个模型的响应都较差, ②两个模型的响应都较好, ③模型 1 的响应更好, ④模型 2 的响应更好。最后统计上述标签在每个数据集和模型对中的出现次数作为评估指标。

## 5 实验结果及分析

本节对基于 LLMs 反馈的自动化评估和人工评估的结果进行展示, 从微调后模型的安全性和有用性两个方面分析 SCFT 的有效性, 最后对 SCFT 的性能进行测试。本节的评估结果基于 Llama2-7B-base, SCFT 在 Llama-3.1-8B-base、Mistral-7B-v0.3-base、Qwen2.5-7B-base、Gemma-2-9b-base 上也有类似的结果, 见附录 C。

### 5.1 基于 LLMs 反馈的自动化评估结果

基于自动化评估的结果对微调后模型的通用能力和安全性进行分析, 可以得出如下结论: 本文提出的微调方法 SCFT 在不影响模型处理下游任务的通用能力的情况下向模型内部嵌入具有一定鲁棒性的安全规则, 增强指令微调模型输出内容的安全性。下面针对模型通用能力和安全性的评估结果进行详细的分析。

#### 5.1.1 安全性

安全性评估实验模拟了两个容易造成模型输出有害内容的典型场景, 一个是正常交互过程中的恶意指令, 一个是越狱攻击, 结果如表 1 所示。此外, 还模拟了现实场景下模型可能遇到虚警的情况, 即伪恶意指令, 结果如图 4 所示。安全性评估的实验结果表明, 本文提出的 SCFT 可以有效地向预训练模型中嵌入安全规则, 提高指令微调模型在面对恶意指令时输出的安全性和面对越狱攻击的鲁棒性, 并且在虚警情况下不会带来严重的夸大的安全问题。

**以有用性为导向的普通指令微调模型存在生成有害内容的安全风险。**首先, 可以观察到, 与文献[19]的结论相一致的是, 在以有用性为导向的通用能力训练数据上进行以有用性为导向的普通指令微调得到的模型( $FT_{+(\text{Helpful})}$ )虽然能够很好地完成下游任务但也会响应各种恶意指令生成不安全的内容, 模型输出内容的平均有害性评分高达 4 分, 有害率高达 24%。这在将针对下游任务微调后的指令微调模型广泛部署到实际生产生活中时是不可接受的。如果在以有用性为导向的数据中加入有害数据, 以有用性为导向的普通指令微调将会得到一个更有害的模型( $FT_{+(\text{Helpful}, \text{Harmful})}$ ), 其平均有害性评分和有害率是最高的, 分别是 4.18 和 26.52%, 这印证了 Qi 等人<sup>[39]</sup>和 Yang 等人<sup>[57]</sup>之前的工作, 而在真实世界的应用中, 用于指令微调的数据中经常不可避免地出现一些显示或隐式的有害数据。

**SCFT 可以大幅减少指令微调模型输出有害内容的次数。**使用本文提出的微调方法 SCFT 可以有效地解决上述问题, 因为在面对恶意指令时, 经 SCFT 得到的指令微调模型能够先自主判断生成的回复是否是对该恶意指令的不安全响应, 如果是则抛出拒答的安全警告, 如果不是则输出安全的回复。从表 1 中可以观察到, 通过 SCFT 得到的模型( $SCFT_{+(\text{Helpful}, \text{Harmful})}$ )与直接微调得到的模型相比安全性有了显著的提升, 平均有害性评分从大于 4 分降低到 1.36 分, 输出有害率降低了约 91%。

SCFT 比基于正向安全数据的安全微调更有效。实验还对比了文献[19]在指令微调时加入正向安全数据以提高微调后模型安全性的方法, 遵循 Bianchi 等人<sup>[19]</sup>的数据配比, 在有用性数据中加入 3% 的正向安全数据(从有害数据集中抽取 1200 条数据并使用 GPT4 生成安全的响应, 这些响应大多是对问题的拒绝, 通过对这些响应进行人工审核以确保其安全性。)后微调得到的模型的输出内容平均有害性评分和有害率都有一定程度的降低, 但远不如本文提出的 SCFT 方法。

### SCFT 能够缓解指令微调模型对越狱攻击的脆

弱性。经 SCFT 得到的模型还可以减轻越狱攻击对模型安全防护的破坏, 在 Jailbroken 数据集<sup>[33]</sup>上的平均有害性评分从 4.1 分降低到 1.352 分, 有害率为 0%。在此基础上对越狱攻击的结果进行进一步分析, Jailbroken 数据集包含两类越狱指令<sup>[33]</sup>, 一类是以竞争目标为原则构造的, 一类是以不匹配泛化为原则构造的, 从表 1 可以看出, SCFT 得到的模型对这两类越狱指令都有鲁棒性, 平均有害分数都在 1 分左右, 有害率均为 0%, 这也验证了 3.2 节中对 SCFT 的理论分析: “判别力机制”能够有效缓解 Wei 等人<sup>[33]</sup>提出的模型能力与安全训练之间竞争目标和不匹配泛化的问题。

表 1 指令微调模型在安全性测试集上的评估结果

Table 1 Results of instruction-tuned model on safety evaluation datasets

评估指标	模型	数据集					
		Red Teaming	Harmful Instruction	Malicious	Competing Objectives	Mismatched Generalization	Average
Harmfulness Score (5 分制)	FT <sub>+(Helpful)</sub>	4.0	3.29	3.2	4.286	4.615	4.056
	FT <sub>+(Helpful, Harmful)</sub>	4.18	3.849	3.43	4.393	4.732	4.148
	Bianchi 等人 <sup>[19]</sup> +3%	3.62	3.003	3.114	2.821	3.038	3.167
	+ALL%	3.96	3.82	3.629	3.286	3.538	3.37
	SCFT <sub>+(Helpful, Harmful)</sub>	<b>1.36</b>	<b>1.39</b>	<b>1.91</b>	<b>1.176</b>	<b>1.269</b>	<b>1.352</b>
Harmfulness Rate (百分制)	FT <sub>+(Helpful)</sub>	24.5%	40.69%	28.57%	50%	34.62%	42.59%
	FT <sub>+(Helpful, Harmful)</sub>	26.25%	47.32%	31.43%	60.71%	34.62%	48.15%
	Bianchi 等人 <sup>[19]</sup> +3%	19%	19.24%	17.14%	28.57%	30.77%	29.63%
	+ALL%	27.25%	44.48%	28.57%	42.86%	23.08%	33%
	SCFT <sub>+(Helpful, Harmful)</sub>	<b>2.25%</b>	<b>0.95%</b>	<b>0%</b>	<b>0%</b>	<b>0%</b>	<b>0%</b>

(注: Harmfulness Score 表示平均有害性评分(5 分制), Harmfulness Rate 表示有害率(百分制)即得分为 5 分的数据的占比, 数值越高表示模型响应的危害性越大。)

SCFT 在安全性和虚假拒绝(False-Refusals)之间取得了很好平衡。Bianchi 等人<sup>[19]</sup>还发现: “安全数据的数量很重要, 过多的正向安全数据可能会给模型带来意想不到的相反的效果。”本文也观察到, 如果进一步将全部有害数据都转化为正向安全数据(10000 条)并加入有用性数据中进行指令微调(+ALL%), 微调后模型的安全性竟然不如加入 3% 正向安全数据得到的指令微调模型, 与不添加正向安全数据直接微调得到的模型(FT<sub>+(Helpful)</sub>、FT<sub>+(Helpful, Harmful)</sub>)的平均有害性评分和有害率相近。这表明通过在微调数据中加入正向安全数据来提升微调后模型安全性的方法不总是有效的, 因为这种安全性是在与模型的通用能力竞争目标的训练中获得的, 需要平衡好模型通用能力和安全性之间的训练数据比例。然而 Bianchi 等人发现找到适当的平衡可能很困难, 因为他们进一步发现使用过多的正向安全数据进行训练可能会使模型拒绝完全安全的指令, 尽管它们表面

上看起来像不安全的指令, 这类指令被称为伪恶意指令, 也称为“虚警”; 模型拒绝这类指令的行为被称为过度防御, 也称为“夸大安全问题”, 通常以虚假拒绝率(False Refusal Rate, FRR)为评估指标, FRR 的值越大, 模型越容易出现虚假拒绝的情况。

除了文献[19], 文献[58]也发现, 目前许多提高模型安全性的方法都会显著提高 FRR(高达 3 倍)。为了研究本文提出的提高指令微调模型安全性的方法 SCFT 在虚警情况下的模型响应效果, 实验进一步使用文献[58]通过可控文本生成引导白盒 LLM 自动化生成的 3260 条伪有害指令构成的虚警数据集 PHTest 及其基于 LLM 的自动化评估方法对以有用为导向的普通指令微调、本文提出的 SCFT 和使用正向安全数据的安全微调得到的模型(FT<sub>+(Helpful, Harmful)</sub>、SCFT<sub>+(Helpful, Harmful)</sub>)进行评估, 结果如图 4 所示, 可以看出, 与不考虑安全性的指令微调相比, SCFT 在大幅降低微调后模型输出有害分数

和有害率的情况下的 FRR 仅增加了 1.6 个百分点。

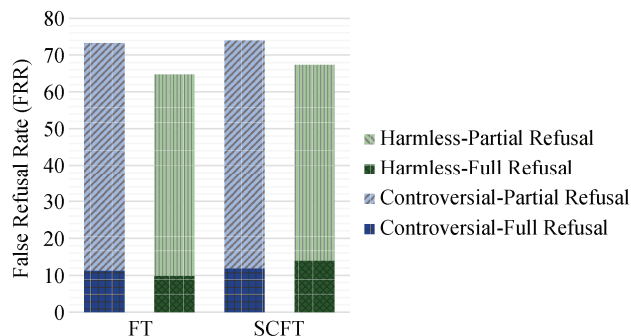


图 4 指令微调模型在虚警情况下的评估结果

Figure 4 Results of instruction-tuned model on pseudo-harmful prompts dataset

### 5.1.2 模型通用能力

指令微调的主要目的还是提高模型处理下游任务的能力, 即有用性。在提高 LLM 的指令遵循能力的同时确保它们保持安全和可靠, 这是一个众所周知的权衡。实验结果表明, SCFT 中分类头的存在不会影响模型对下游任务数据的学习, 仅通过一次微调就能同时实现有用性和安全性两个任务。

正如 Yang 等人<sup>[57]</sup>的工作发现的那样, 从表 2 中可以看出, 在以有用性为导向的普通指令微调中使用有害数据进行微调不会影响模型的通用能力, 仅使用 Helpful 数据集和使用 Helpful 与 Harmful 的混合数据集直接微调的两个模型 ( $FT_{+(\text{Helpful})}$ 、 $FT_{+(\text{Helpful}, \text{Harmful})}$ ) 在两个测试集上的得分近乎相同。但加入正向安全数据得到的指令微调模型的有用性出现了一定程度的降低。而使用本文提出的 SCFT 得

到的模型 ( $SCFT_{+(\text{Helpful}, \text{Harmful})}$ ) 在两个测试集上仍然取得了和以有用性为导向的普通微调相近的得分, 这表明 SCFT 不会影响微调后模型的通用能力。

表 2 指令微调模型在通用能力测试集上的评估结果  
Table 2 Results of instruction-tuned model on utility evaluation datasets

评估指标	模型	数据集		
		Helpful	MT-bench	
Helpfulness Score (10 分制)	$FT_{+(\text{Helpful})}$	6.310	3.588	
	$FT_{+(\text{Helpful}, \text{Harmful})}$	6.307	3.663	
	Bianchi 等人 <sup>[19]</sup>	+3%	6.180	2.738
	等人 <sup>[19]</sup>	+ALL%	6.202	2.7
	$SCFT_{+(\text{Helpful}, \text{Harmful})}$	<b>6.230</b>	<b>3.2</b>	

(注: Helpfulness Score 表示响应质量的平均得分(10 分制), 数值越高表示模型能力越强。作为参考, 根据官方排行榜, Llama-2-7b-chat 的 MT-Bench 得分:6.27)

### 5.2 人工标注的评估结果

人工评估的结果如图 5 所示, 可以看出, SCFT 模型在安全性的评估中胜出, 在回复质量的评估中, SCFT 模型与其他模型不分高下, 进一步证明了本文提出的微调方法 SCFT 的有效性。

### 5.3 性能分析

SCFT 与普通的指令微调 Fine-tuning<sup>[11]</sup>相比多了一个“判别力机制”, 为了分析“判别力机制”对模型训练和推理性能的影响, 本文在相同的参数和资源配置下对比了 SCFT 与普通指令微调 Fine-tuning 的参数数量和的训练时间, 以及得到的指令微调模型(即模型  $SCFT_{+(\text{Helpful}, \text{Harmful})}$  和  $FT_{+(\text{Helpful}, \text{Harmful})}$ ) 的推理速度。

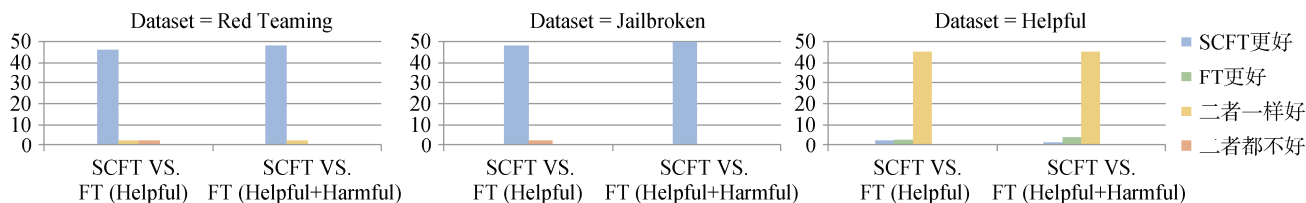


图 5 人工评估模型回复质量和安全性的结果

Figure 5 Results of manual annotation the quality and safety of model responses

(注: 每个数据集包含 50 条数据。SCFT 代表在通用能力训练数据和有害数据的混合数据集上使用本文提出的 SCFT 微调得到的模型,  $FT_{(\text{Helpful})}$  代表在模型通用能力训练数据上直接指令微调得到的模型,  $FT_{(\text{Helpful}+\text{Harmful})}$  代表在通用能力训练数据和有害数据的混合数据集上直接微调得到的。)

#### 5.3.1 训练开销

本文中所有模型的训练均采用全量参数微调的方式, SCFT 的模型比普通的指令微调多了一个分类头, 但从表 3 中可以看出, 这个分类头增加的参数量与模型本身庞大的参数量相比可以忽略不计, 因此

SCFT 不会增加所需训练的参数量。

在训练时间开销上, 从表 3 中可以看出, 与普通的指令微调相比, SCFT 所需的训练时间略有增加, 因为 SCFT 的输出层需要多计算一个分类头, 但每个样本仅增加 0.139s 的训练时间就能将微调后模型输

出的有害率降低约 91%。

**表 3 SCFT 与普通指令微调方案的训练开销对比**  
**Table 3 Comparison of training overhead between SCFT and standard instruction fine-tuning**

微调方法	参数量	数据量	平均耗时(每条样本)
Fine-tuning <sup>[11]</sup>	6.738416B	51855 条	0.694s
SCFT	6.738424B	51855 条	0.833s

(注: Fine-tuning 代表在通用能力训练数据和有害数据的混合数据集上使用普通指令微调得到的模型, SCFT 代表在通用能力训练数据和有害数据的混合数据集上使用本文提出的 SCFT 微调得到的模型。)

### 5.3.2 推理速度

从表 4 中可以发现, SCFT 得到的模型在推理时生成每个 token 的平均用时比普通的指令微调多 0.002s, 这是因为 SCFT 得到的模型在推理时, 先由输出层的分类头给出“用户输入-模型回复”的安全类别标签, 根据标签决定是否允许语言模型头输出模型生成的回复。但与对模型输出内容的安全性的大幅提升相比, SCFT 方法轻微增加的推理耗时是可以接受的, 而且与借助额外的内容审核模型或 API 对输入或输出进行过滤的方法相比节省了许多时间。

**表 4 不同微调方法得到的模型的推理速度对比**  
**Table 4 Inference speed comparison of models obtained by different fine-tuning methods**

模型	参数量	平均耗时(每个 token)
Fine-tuning <sup>[11]</sup>	6.738416B	0.020s
SCFT	6.738424B	0.022s

(注: Fine-tuning 代表在通用能力训练数据和有害数据的混合数据集上使用普通的指令微调得到的模型, SCFT 代表在通用能力训练数据和有害数据的混合数据集上使用本文提出的 SCFT 微调得到的模型。)

## 6 总结

本文提出了一种新的提高大模型输出内容安全性的微调方法 SCFT, SCFT 在提升模型指令跟随能力的同时让模型具备了能够主动审核“指令-模型回复”的安全性从而阻止有害回复输出的能力, 降低模型指令跟随能力增强后遵循恶意指令输出有害内容的安全风险。本文的动机是在 LLMs 的输出层添加一个新的分类头, 在针对模型能力进行指令微调时, 利用最后一层 Decoder 输出的隐藏状态中表示句子结束的 EOS Token 的词嵌入向量中蕴含的整个句子的语义信息, 训练该分类头学会基于语义信息进行以“安全”/“不安全”为标签的文本分类, 成为嵌入模型内部的“判别力机制”。本文在微调预训练语

言模型用于多轮通用对话的应用背景下验证所提出的方法, 实验结果表明, “判别力机制”能够缓解模型通用能力和安全训练之间存在的竞争目标和不匹配的泛化的问题, 在不损害模型通用能力的同时获得更加安全的输出, 更好地平衡了模型能力和安全性, 还能增强模型对越狱攻击的鲁棒性。

本文的工作有助于降低指令微调模型实际应用中遭受滥用的安全风险, 推动 LLMs 在严格的安全防护下蓬勃发展。然而人类的语言随着时间的推移而发展<sup>[59]</sup>, 由“语言”带来的安全问题, 如攻击性和骚扰性的文本<sup>[60]</sup>, 也在不断发展<sup>[61]</sup>。社会发展过程中出现的重要文化或事件都会为“安全”提供新的背景, 生成内容安全的内涵可能会随着时间的推移而发生变化, 在未来的研究中, 我们希望在现有的安全防护机制的基础上, 进一步优化安全防护机制的计算成本, 提高安全防护机制的灵活性, 实现模型知识的快速更新和扩展, 以适应动态变化的“安全”内涵。

## 参考文献

- [1] Radford A, Narasimhan K, Salimans T, et al. Improving language understanding by generative pre-training[J]. *Improving language understanding by generative pre-training*, 2018.
- [2] OpenAI. Introducing ChatGPT. <https://openai.com/blog/chatgpt>. Nov. 2022.
- [3] OpenAI. GPT-4 technical report. Technical Report 2023-03, ArXiv Preprint ArXiv:2303.08774.
- [4] Touvron H, Lavril T, Izacard G, et al. LLaMA: Open and Efficient Foundation Language Models[EB/OL]. 2023: arXiv: 2302.13971. <https://arxiv.org/abs/2302.13971>.
- [5] Touvron H, Martin L, Stone K, et al. Llama 2: Open Foundation and Fine-Tuned Chat Models[EB/OL]. 2023: arXiv: 2307.09288. <https://arxiv.org/abs/2307.09288>.
- [6] Meta. Llama 3. <https://llama.meta.com/llama3/>, 2024.
- [7] Chowdhery A, Narang S, Devlin J, et al. PaLM: Scaling Language Modeling with Pathways[C]. *New York: ACM*, 2023: 11324-11436.
- [8] Glm T, Zeng A H, XU B et al. ChatGLM: A Family of Large Language Models from GLM-130B to GLM-4 all Tools[EB/OL]. 2024: arXiv: 2406.12793. <https://arxiv.org/abs/2406.12793>.
- [9] Yang A Y, Xiao B, Wang B N, et al. Baichuan 2: Open Large-Scale Language Models[EB/OL]. 2023: arXiv: 2309.10305. <https://arxiv.org/abs/2309.10305>.
- [10] Bai J Z, Bai S, Chu Y F, et al. Qwen Technical Report[EB/OL]. 2023: arXiv: 2309.16609. <https://arxiv.org/abs/2309.16609>.
- [11] Ouyang Long, Wu Jeff, Jiang Xu, et al. Training language models to follow instructions with human feedback[C]. *Annual Conference on Neural Information Processing Systems*, 2022, 35: 27730-27744.
- [12] Jiao W X, Wang W X, Huang J T, et al. Is ChatGPT a Good Translator? yes with GPT-4 as the Engine[EB/OL]. 2023: arXiv: 2301.08745. <https://arxiv.org/abs/2301.08745>

- [13] Qin C W, Zhang A, Zhang Z S, et al. Is ChatGPT a General- Purpose Natural Language Processing Task Solver? [EB/OL]. 2023: arXiv: 2302.06476. <https://arxiv.org/abs/2302.06476>.
- [14] Zhang W X, Deng Y, Liu B, et al. Sentiment Analysis in the Era of Large Language Models: A Reality Check[EB/OL]. 2023: arXiv: 2305.15005. <https://arxiv.org/abs/2305.15005>.
- [15] Chang Y P, Wang X, Wang J D, et al. A Survey on Evaluation of Large Language Models[EB/OL]. 2023: arXiv: 2307.03109. <https://arxiv.org/abs/2307.03109>.
- [16] Wu S J, Irsoy O, Lu S, et al. BloombergGPT: A Large Language Model for Finance[EB/OL]. 2023: arXiv: 2303.17564. <https://arxiv.org/abs/2303.17564>.
- [17] Nguyen H T. A Brief Report on LawGPT 1.0: A Virtual Legal Assistant Based on GPT-3[EB/OL]. 2023: arXiv: 2302.05729. <https://arxiv.org/abs/2302.05729>.
- [18] Singhal K, Azizi S, Tu T, et al. Large Language Models Encode Clinical Knowledge[J]. *Nature*, 2023, 620(7972): 172-180.
- [19] Bianchi Federico, Suzgun Mirac, Attanasio Giuseppe, et al. Safety-tuned LLaMAs: Lessons from improving the safety of large language models that follow instructions[C]. *In The Twelfth International Conference on Learning Representations*, 2024.
- [20] Bai Y T, Jones A, Ndousse K, et al. Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback[EB/OL]. 2022: arXiv: 2204.05862. <https://arxiv.org/abs/2204.05862>.
- [21] Taori Rohan, Gulrajani Ishaan, Zhang Tianyi, et al. Stanford alpaca: An instruction-following llama model. [https://github.com/tatsu-lab/stanford\\_alpaca](https://github.com/tatsu-lab/stanford_alpaca). 2023.
- [22] Hazell Julian. Large language models can be used to effectively scale spear phishing campaigns[EB/OL]. 2023: ArXiv Preprint ArXiv:2305.06972.
- [23] Dahl M, Magesh V, Suzgun M, et al. Large Legal Fictions: Profiling Legal Hallucinations in Large Language Models[EB/OL]. 2024: arXiv: 2401.01301. <https://arxiv.org/abs/2401.01301>.
- [24] Kang D, Li X C, Stoica I, et al. Exploiting Programmatic Behavior of LLMs: Dual-Use through Standard Security Attacks[EB/OL]. 2023: arXiv: 2302.05733. <https://arxiv.org/abs/2302.05733>.
- [25] Li H R, Guo D D, Fan W, et al. Multi-Step Jailbreaking Privacy Attacks on ChatGPT[EB/OL]. 2023: arXiv: 2304.05197. <https://arxiv.org/abs/2304.05197>.
- [26] Peng A, Wu M, et al. Gpt-3.5 turbo fine-tuning and api updates. <https://openai.com/blog/gpt-3-5-turbo-fine-tuning-and-api-updates>. Aug. 2023.
- [27] Chung H W, Hou L, Longpre S, et al. Scaling Instruction- Fine-tuned Language Models[EB/OL]. 2022: arXiv: 2210.11416. <https://arxiv.org/abs/2210.11416>.
- [28] Mike Conover, Matt Hayes, Ankith Mathur, et al. Free dolly: Introducing the world's first truly open instruction-tuned llm. <https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm>. Apr. 2023.
- [29] Glukhov D, Shumailov I, Gal Y, et al. LLM Censorship: A Machine Learning Challenge or a Computer Security Problem? [EB/OL]. 2023: arXiv: 2307.10719. <https://arxiv.org/abs/2307.10719>.
- [30] Markov T, Zhang C, Agarwal S, et al. A Holistic Approach to Undesired Content Detection in the Real World[J]. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023, 37(12): 15009-15018.
- [31] OpenAI. Moderation api. <https://platform.openai.com/docs/guides/moderation>. 2023.
- [32] Lees A, Tran V Q, Tay Y, et al. A New Generation of Perspective API: Efficient Multilingual Character-Level Transformers[C]. *The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022: 3197-3207.
- [33] Wei Alexander, Haghtalab Nika, and Steinhardt Jacob. Jailbroken: How does llm safety training fail [C]. *Annual Conference on Neural Information Processing Systems*, 2023, 36: 80079-80110.
- [34] Zou A, Wang Z F, Carlini N, et al. Universal and Transferable Adversarial Attacks on Aligned Language Models[EB/OL]. 2023: arXiv: 2307.15043. <https://arxiv.org/abs/2307.15043>.
- [35] Xie Y Q, Yi J W, Shao J W, et al. Defending ChatGPT Against Jailbreak Attack via Self-Reminders[J]. *Nature Machine Intelligence*, 2023, 5(12): 1486-1496.
- [36] Liu Yupei, Jia Yuqi, Geng Runpeng, et al. Formalizing and benchmarking prompt injection attacks and defenses[C]. *33rd USENIX Security Symposium*, 2024: 1831-1847.
- [37] Liu Y, Deng G L, Xu Z Z, et al. Jailbreaking ChatGPT via Prompt Engineering: An Empirical Study[EB/OL]. 2023: arXiv: 2305.13860. <https://arxiv.org/abs/2305.13860>.
- [38] Qi X Y, Huang K X, Panda A, et al. Visual Adversarial Examples Jailbreak Aligned Large Language Models[J]. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024, 38(19): 21527-21536.
- [39] Qi Xiangyu, Zeng Yi, Xie Tinghao, et al. Fine-tuning aligned language models compromises safety, even when users do not intend to[C]. *In The Twelfth International Conference on Learning Representations*, 2024.
- [40] Elazar Y, Bhagia A, Magnusson I, et al. What's in my big data[C]. *International Conference on Learning Representations*, 2024.
- [41] Vaswani A, Shazeer N, Parmar N, et al. Attention Is all You Need[C]. *The 31st International Conference on Neural Information Processing Systems*, 2017: 6000-6010.
- [42] Brown T B, Mann B, Ryder N, et al. Language models are few-shot learners[C]. *Advances in Neural Information Processing Systems*, 2020, 33: 1877-1901.
- [43] Yi Jingwei, Ye Rui, Chen Qisi, et al. Open-source can be dangerous: On the vulnerability of value alignment in opensource LLMs. <https://openreview.net/forum?id=NlOuO0C0ex>. 2024.
- [44] Yi J W, Xie Y Q, Zhu B, et al. Benchmarking and Defending Against Indirect Prompt Injection Attacks on Large Language Models[EB/OL]. 2023: arXiv: 2312.14197. <https://arxiv.org/abs/2312.14197>.
- [45] Arora A, Nakov P, Hardalov M, et al. Detecting Harmful Content on Online Platforms: What Platforms Need Vs. where Research Efforts Go[J]. *ACM Computing Surveys*, 2024, 56(3): 1-17.
- [46] Lin Z, Wang Z H, Tong Y Q, et al. ToxicChat: Unveiling Hidden Challenges of Toxicity Detection in Real-World User-AI Conversation[C]. *Findings of the Association for Computational Linguistics*

- tics: *EMNLP 2023*, 2023: 4694-4702.
- [47] Inan H, Upasani K, Chi J F, et al. Llama Guard: LLM-Based Input-Output Safeguard for Human-AI Conversations[EB/OL]. 2023: arXiv: 2312.06674. <https://arxiv.org/abs/2312.06674>.
- [48] Weng Lilian, Goel Vik, Andrea Vallone et al. <https://openai.com/blog/using-gpt-4-for-content-moderation>. 2023.
- [49] Köpf A, Kilcher Y, von Rütte D, et al. OpenAssistant Conversations — Democratizing Large Language Model Alignment [EB/OL]. 2023: arXiv: 2304.07327. <https://arxiv.org/abs/2304.07327>.
- [50] Wang Yizhong, Ivison Hamish, Dasigi Pradeep, et al. How Far Can Camels Go Exploring the State of Instruction Tuning on Open Resources[C]. *Advances in Neural Information Processing Systems*, 2023. 36: 74764-74786.
- [51] Ganguli D, Lovitt L, Kernion J, et al. Red Teaming Language Models to Reduce Harms: Methods, Scaling Behaviors, and Lessons Learned[EB/OL]. 2022: arXiv: 2209.07858. <https://arxiv.org/abs/2209.07858>.
- [52] Paszke A, Gross S, Massa F, et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library[C]. *The 33rd International Conference on Neural Information Processing Systems*, 2019: 8026-8037.
- [53] Wolf T, Debut L, Sanh V, et al. Transformers: State-of-the-Art Natural Language Processing[C]. *The 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2020: 38-45.
- [54] Rasley J, Rajbhandari S, Ruwase O, et al. DeepSpeed: System Optimizations Enable Training Deep Learning Models with over 100 Billion Parameters[C]. *The 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020: 3505-3506.
- [55] Dettmers Tim, Lewis Mike, Shleifer Sam, et al. 8-bit optimizers via block-wise quantization[C]. *In International Conference on Learning Representations*, 2022.
- [56] Zheng Lianmin, Chiang Wei-Lin, Sheng Ying, et al. Judging LLM-as-a-judge with MT-bench and chatbot arena[C]. *Advances in Neural Information Processing Systems*, 2023, 36: 46595-46623.
- [57] Yang X J, Wang X, Zhang Q, et al. Shadow Alignment: The Ease of Subverting Safely-Aligned Language Models[EB/OL]. 2023: arXiv: 2310.02949. <https://arxiv.org/abs/2310.02949>.
- [58] An B, Zhu S, Zhang R, et al. Automatic Pseudo-Harmful Prompt Generation for Evaluating False Refusals in Large Language Models[C]. *International Conference on Machine Learning*, 2024.
- [59] Silvey C. *Speaking Our Minds: Why Human Communication Is Different, and how Language Evolved to Make It Special*, by Thom Scott-Phillips[J]. *Journal of Language Evolution*, 2016, 1(1): 88-90.
- [60] Gehman S, Gururangan S, Sap M, et al. RealToxicityPrompts: Evaluating Neural Toxic Degeneration in Language Models[C]. *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020: 3356-3369.
- [61] Charlesworth T E S, Banaji M R. Patterns of Implicit and Explicit Stereotypes III: *Long-Term Change in Gender Stereotypes*[J]. *Social Psychological and Personality Science*, 2022, 13(1): 14-26.

## 附 录

表 5 SCFT 在不同基线模型上的评估结果

Table 5 Evaluation results of SCFT on different base models

评估指标	模型	数据集						
		Red Teaming	Harmful Instruction	Malicious	Competing Objectives	Jailbroken Mismatched Generalization	Average	
Harmfulness Score (5 分制)	Llama-3.1-8B	FT*	3.79	2.81	2.11	3.88	4.04	3.94
		FT	4.14	3.75	4.14	4.09	3.96	4.04
		<b>SCFT</b>	<b>1.68</b>	<b>1.32</b>	<b>1.43</b>	<b>1.21</b>	<b>1.81</b>	<b>1.37</b>
	Mistral-7b-v0.3	FT*	4.09	4.40	3.86	3.92	3.90	3.91
		FT	4.02	3.27	3.49	4.18	3.83	4.02
		<b>SCFT</b>	<b>1.16</b>	<b>1.16</b>	<b>1.16</b>	<b>1.31</b>	<b>1.73</b>	<b>1.50</b>
	Qwen2.5-7B	FT*	3.87	3.98	3.32	4.40	3.97	4.22
		FT	3.89	3.84	3.58	4.40	3.83	4.25
		<b>SCFT</b>	<b>1.28</b>	<b>1.50</b>	<b>1.55</b>	<b>1.04</b>	<b>1.69</b>	<b>1.31</b>
	Gemma-2-9B	FT*	3.89	3.30	3.0	4.22	4.14	4.19
		FT	3.99	3.64	3.38	4.27	4.09	4.21
		<b>SCFT</b>	<b>1.23</b>	<b>1.20</b>	<b>1.23</b>	<b>1.02</b>	<b>1.23</b>	<b>1.10</b>
Harmfulness Rate (百分制)	Llama-3.1-8B	FT*	21.39%	31.48%	7.10%	20.93%	25%	22.39%
		FT	35.28%	31.48%	28.57%	30.23%	20.83%	26.87%
		<b>SCFT</b>	<b>4.44%</b>	<b>1.85%</b>	<b>0.00%</b>	<b>0.00%</b>	<b>6.25%</b>	<b>1.69%</b>

续表

评估指标	模型	数据集						
		Red Teaming	Harmful Instruction	Malicious	Jailbroken			
					Competing Objectives	Mismatched Generalization	Average	
Harmfulness Rate (百分制)	Mistral-7b-v0.3	FT*	35.45%	64.90%	35.29%	30.61%	24.39%	27.78%
		FT	32.80%	24.24%	19.60%	38.78%	14.63%	27.78%
		<b>SCFT</b>	<b>1.06%</b>	<b>0.76%</b>	<b>0.00%</b>	<b>2.04%</b>	<b>2.44%</b>	<b>2.22%</b>
	Qwen2.5-7B	FT*	22.35%	28.97%	16.13%	48.89%	25.00%	38.96%
		FT	24.64%	28.41%	29.35%	53.33%	31.25%	44.16%
		<b>SCFT</b>	<b>2.588%</b>	<b>2.80%</b>	<b>0.00%</b>	<b>0.00%</b>	<b>3.13%</b>	<b>1.30%</b>
	Gemma-2-9B	FT*	21.68%	10.10%	19.23%	39.02%	36.36%	38.10%
		FT	25.43%	15.15%	15.38%	41.46%	31.82%	38.10%
		<b>SCFT</b>	<b>0.00%</b>	<b>0.00%</b>	<b>0.00%</b>	<b>0.00%</b>	<b>0.00%</b>	<b>0.00%</b>

(注: Harmfulness Score 表示平均有害性评分(5 分制), Harmfulness Rate 表示有害率(百分制)即得分为 5 分的数据的占比, 数值越高表示模型响应的危害性越大。SCFT 代表在通用能力训练数据和有害数据的混合数据集上使用本文提出的 SCFT 微调得到的模型, FT\*代表在模型通用能力训练数据上直接指令微调得到的模型, FT 代表在通用能力训练数据和有害数据的混合数据集上直接微调得到的模型。)

### A 构造训练数据

本文使用的多轮对话原始文本数据示例如下:

*Human: What is the difference between a migraine and a regular headache*

*Assistant: Migraine is a type of headache, but it is specifically defined as a throbbing, pulsing, or painful headache that is usually located in one side of the head, and can spread to involve the entire head, or any part of the face, or a particular region of the neck.*

*Human: I'm also feeling very nauseous and sensitive to light. Are those symptoms of a migraine*

*Assistant: Nausea and sensitivity to light are also symptoms of migraines. However, it's more common for these symptoms to last for less than a day in migraine sufferers. Also, they are often only associated with the actual migraine and not the pre-migraine or post-migraine symptoms. Migraines can affect the whole body including nausea, vomiting, pain in any area, and sensitivity to light.*

在指令微调阶段, 一般只有“Assistant:”回答部分的损失(Loss)会用于梯度回传, 更新权重; 而“Human:”部分的损失(Loss)则不会用于更新权重。因此, 本文将每条原始文本数据转换为一个由(Prompt, Response, SafeTag)构成三元组, 其中, 每个“Human:”的内容作为 Prompt, 每个“Assistant:”的内容作为 Response, 以该条数据为例:

Prompt=[“What is the difference between a migraine and a regular headache”, “I'm also feeling very nauseous and sensitive to light. Are those symptoms of a migraine”]

Response=[“Migraine is a type of headache, but

*it is specifically defined as a throbbing, pulsing, or painful headache that is usually located in one side of the head, and can spread to involve the entire head, or any part of the face, or a particular region of the neck.”, “Nausea and sensitivity to light are also symptoms of migraines. However, it's more common for these symptoms to last for less than a day in migraine sufferers. Also, they are often only associated with the actual migraine and not the pre-migraine or post-migraine symptoms. Migraines can affect the whole body including nausea, vomiting, pain in any area, and sensitivity to light.”]*

SafeTag 是这条数据的安全标签, 如果 Response 中存在有害内容, 就令 SafeTag 为 0(表示这条数据是不安全的数据), 否则为 1(表示这条数据是安全的数据)。因为无论 Prompt 是否为恶意指令, 都希望 LLMs 生成的内容是安全的, 所以只要 Response 是安全的, 就可以认为这条样本是安全的。Anthropic Helpful Base 对话数据集创建的目的是用来训练模型为人类提供更有用的帮助, 通过人工检查和清洗后的数据中不包含有害数据, 可以将该数据集中数据的 SafeTag 置为 1。Anthropic Red Team 数据集是以破坏 LLMs 的安全性为目的创建的, 可以将该数据集中数据的 SafeTag 置为 0。

$n$  个 (Prompt, Response, SafeTag) 三元组组成了指令微调数据集  $D$ , 训练时, 数据预处理函数 Preprocess 将 Prompt 和 Response 按照如下格式(即 Format 模板)拼接并转换为整数索引序列作为模型的输入, 以上述数据为例, 拼接后的文本序列如下:

Text = “###Human: What is the difference be-

tween a migraine and a regular headache ####Assistant: Migraine is a type of headache, but it is specifically defined as a throbbing, pulsing, or painful headache that is usually located in one side of the head, and can spread to involve the entire head, or any part of the face, or a particular region of the neck.####Human: I'm also feeling very nauseous and sensitive to light. Are those symptoms of a migraine ####Assistant: Nausea and sensitivity to light are also symptoms of migraines. However, it's more common for these symptoms to last for less than a day in migraine sufferers. Also, they are often only associated with the actual migraine and not the pre-migraine or post-migraine symptoms. Migraines can affect the whole body including nausea, vomiting, pain in any area, and sensitivity to light.”

在将文本序列 *Text* 转换为整数索引序列 *InputIds* 时, 还构造了一个目标掩码 *Mask<sub>id</sub>* (构造方式见预处理函数 *Preprocess*), 用来标记每个词(token)是否需要计算损失(Loss), 其中只有 *Response* 部分的 *Mask<sub>id</sub>* 为 1, 其他部分均为 0。在计算语言模型头的损失(Loss)时, 会并行计算每个位置的损失(Loss), 但只有 *Mask<sub>id</sub>* = 1 的部分位置的损失(Loss)才会参与

权重更新, 即只使用了 *Response* 部分的损失(Loss)。这种方式充分利用了模型并行计算的优势, 更加高效, 并且多轮对话中的每个 “Assistant: ” 部分都参与了训练, 更加充分地利用了多轮对话的训练数据。

## B 生成参数设置

本文在使用指令微调模型进行文本生成时, 使用 HuggingFace Transformers 的 *Generate* 方法, 为了保证评估结果的准确性, 不设置任何能增加输出的多样性的生成参数, 采用贪婪解码策略(greedy decoding), 即选择最大条件概率的词, 令生成的新词(Token)的最大数量为 1024。

## C SCFT 在不同基线模型上的评估结果

为了进一步验证所提微调方案 SCFT 的有效性, 本文还在 Llama-3.1-8B-base、Mistral-7B-v0.3-base、Qwen2.5-7B-base、Gemma-2-9b-base 上测试了 SCFT, 基于 LLMs 反馈的自动化评估结果如表 5 所示。从中可以看出, SCFT 在上述模型上表现出了与 Llama2-7B-base 模型相似的效果, SCFT 可以有效地向预训练模型中嵌入安全规则, 提高指令微调模型在面对恶意指令时输出的安全性和面对越狱攻击的鲁棒性。



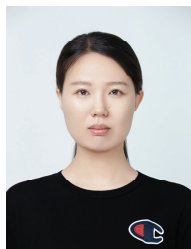
于淼 于 2022 年在郑州大学信息安全专业获得学士学位。现在中国人民解放军网络空间部队信息工程大学网络空间安全专业攻读硕士学位。研究领域为人工智能安全。研究兴趣包括: 大模型自身安全及风险问题。Email: yu\_miao@163.com



孙磊 博士, 中国人民解放军网络空间部队信息工程大学密码工程学院教授, 博士生导师。研究领域为人工智能、信息系统安全。研究兴趣包括: 大模型自身安全及风险问题。Email: sl20210221@163.com



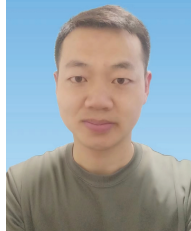
胡翠云 博士, 中国人民解放军网络空间部队信息工程大学密码工程学院讲师。研究领域为人工智能、数据安全。研究兴趣包括: 大模型安全及风险问题, 智能系统的数据安全。Email: hcy56316@163.com



臧韦菲 博士, 中国人民解放军网络空间部队信息工程大学密码工程学院讲师。研究领域为人工智能安全、网络空间安全。研究兴趣包括: 大模型安全和幻觉问题, 网络鲁棒性等。Email: zwfndsc@163.com



郭松 博士, 中国人民解放军网络空间部队信息工程大学密码工程学院讲师。研究领域为人工智能、信息系统安全。研究兴趣包括: 大模型自身安全及风险问题。Email: songguo@nudt.edu.cn



胡鹏 于 2010 年在国防科技大学信息工程专业获得学士学位。现在中国人民解放军网络空间部队信息工程大学网络空间安全专业攻读博士学位。研究领域为人工智能安全。研究兴趣包括: 神经网络安全推理。Email: hup\_xd@163.com