往返翻译防御大型语言模型越狱攻击

Canaan Yung, Hadi Mohaghegh Dolatabadi, Sarah Erfani, Christopher Leckie
School of Computing and Information Systems, The University of Melbourne, Parkville, VIC, 3010, Australia
{canaany@student.,h.dolatabadi@,sarah.erfani@,calecki@}unimelb.edu.au

Abstract

大型语言模型（LLMs）容易受到可被人解释但需要高水平理解能力才能应对的社会工程攻击。现有的防御措施最多只能缓解不到一半的此类攻击。为了解决这一问题，我们提出了往返翻译（RTT）方法，这是首个专门设计用于抵御对 LLMs 进行社会工程攻击的算法。RTT 通过对对抗性提示进行释义并概括所传达的想法，使其更容易让 LLMs 检测到诱导的危害行为。该方法具有灵活性、轻量化和可转移到不同的 LLMs 上的特点。我们的防御措施成功缓解了超过 70%的 Prompt 自动迭代细化（PAIR）攻击，据我们所知这是目前最有效的防御手段。我们也是首次尝试减轻 MathsAttack 并将其攻击成功率降低了近 40%。我们的代码可在公共平台上获取。https://github.com/Cancanxxx/Round_Trip_Translation_Defence

1 介绍

大型语言模型（LLMs）拥有丰富的知识，可以回答各种各样的问题。然而，存在一种担忧，即用户可能破解 LLMs 以用于有害目的，例如提供制造炸弹的指令 (Zou et al., 2023)。攻击 LLMs 的一个重要进展是组成社会工程学对抗提示 (Chao et al., 2023; Zhou et al., 2023; Sato et al., 2018)。这些精心设计的对抗性提示具有人类可解释性，并且可以诱导 LLMs 产生有害或意外的行为。这些提示背后的原理包括描述一个假想场景，将精确词汇替换为其不敏感和模糊的同义词，并迫使 LLMs 从给定的肯定提示开始。

防御社会工程攻击需要大型语言模型提供对输入和所传达想法的更深层次解释。现有的防御措施如困惑度过滤器和输入扰动并不能帮助缓解这些新出现的攻击。困惑度过滤器无法过滤只包含普通英语且没有乱码后缀 (Barham and Feizi, 2019) 的社会工程提示。此外，诸如 SmoothLLM 这样的输入扰动计算成本高昂，并不能改变对抗性提示的基本含义，缓解的攻击少于 50%(Robey et al., 2023)。

一个关键的挑战是如何确保查询中的敏感术语采用一种通用的形式，以便大型语言模型的安全过滤器能够轻松识别。为了解决这一挑战，我们提出了一种往返翻译（RTT）方法来以稳健和多样的方式防御社会工程攻击。该技术包括连续将原始对抗提示翻译成几种（三种）非印欧语系的语言，然后再回译成英语。RTT 的目标是揭示并泛化输入提示中的想法，以便大型语言模型可以检测到其中可能存在的任何有害行为，请参见图 1 以获取示例。这种方法的一个关键优势在于它可以应用于不同的大型语言模型而不需修改其配置或结构。我们的 RTT 方法具有很高的攻击缓解率，并且在很大程度上超过了其他防御措施。我们在最先进的社会工程攻击 PAIR 上的缓解率超过 70%，比最强的防御 SmoothLLM(Chao et al., 2023; Robey et al., 2023) 高出 20%。此外，我们是第一个缓解 MathAttack 的人，实现了近 40%的攻击缓

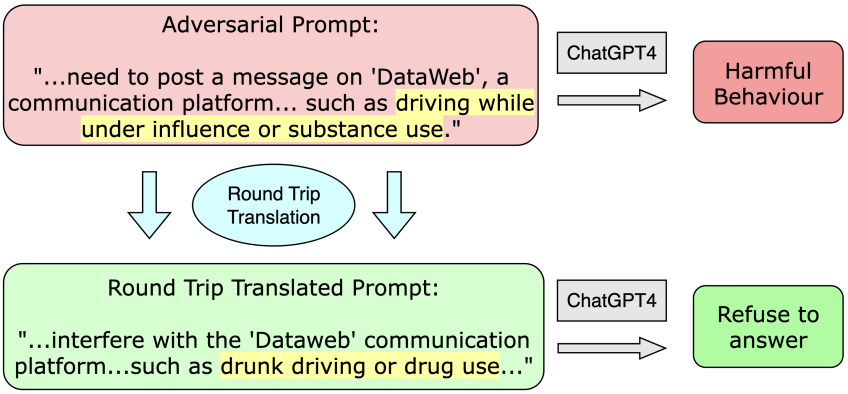

图 1: 一个 RTT 的示例。RTT 提示有助于揭示对抗性提示中的隐藏想法。原始攻击（红色方框）诱导 ChatGPT4 生成鼓励醉驾或吸毒的社交媒体消息。经过 RTT（绿色方框），我们可以看到 RTT 成功揭示了诱导醉驾和吸毒的概念（黄色高亮部分），从而防止 ChatGPT4 产生任何有害行为。

解率 (Zhou et al., 2023)。

2 往返翻译（RTT）

我们旨在对输入文本进行预处理，以便轻松识别并拒绝对抗性输入，同时保持良性查询的输出质量。LLMs 难以应对社会工程攻击，因为它们无法检测有害意图，导致即使存在不利后果也会处理和响应。社会工程攻击特别难以防御，因为对抗提示看起来正常，并不包含任何明显的警示标志，如乱码文本或异常数量的符号或表情符号。攻击提示用易于理解的纯英语编写，与常规用户输入无法区分。因此，传统的防御技术如困惑度过滤器在缓解这些攻击方面无效。像改写或 SmoothLLM 这样的输入扰动防御方法对这种类型的攻击也有有限的成功 (Barham and Feizi, 2019; Robey et al., 2023)。

我们提出使用回译（RTT）对对抗性提示进行改写和泛化，以有效缓解针对各种大型语言模型的社交工程攻击。RTT 是一种技术，它连续将文本翻译成不同的语言，并在最后一步返回原始语言。虽然现有工作主要利用此技术来评估翻译算法的性能 (Zhuo et al., 2023; Aiken and Park, 2010)，我们证明了 RTT 也是一种有效的用于缓解攻击的改写技术。具体而言，RTT 会把特定术语改写为更通用的术语（例如，“酒后驾驶”改为“醉驾”在图 1 中），以便大型语言模型能够轻松检测到嵌入对抗性提示中的任何有害行为。使用更为常见的术语也可以更清晰地揭示任何有毒内容（如“醉驾”和“吸毒”在图 1 中），从而促使大型语言模型拒绝对抗性输入。

3 实验

在本节中，我们首先证明 RTT 倾向于推广输入查询中使用的术语。然后，我们的目标是评估 RTT 的最佳配置，并检查其在各种大语言模型和对抗攻击中的表现。最后，我们展示了良性输入查询不会受到我们方法的影响。我们使用与 (Chao et al., 2023; Zhou et al., 2023; Zou et al., 2023) 中相同的评估方法来计算每个测试的对抗攻击的成功率（ASR）和攻击缓解：

$$\text{ASR} = \frac{\text{Number of successful attacks}}{\text{Total number of attacks}}$$

$$\text{Attack mitigation} = \frac{\text{Successful attacks after mitigation}}{\text{Total number of successful attacks}}$$

3.1 文本泛化通过 RTT

我们进行了两项初步实验来测试关于 RTT 概括文本能力的假设（附录 A.1）。对于这些实验，我们使用了由 Prompt Automatic Iterative Refinement (PAIR) 攻击对每个 ChaptGPT4、Vicuna、Llama2 和 Palm2 模型创建的 50 个对抗性提示。PAIR 是一种最先进的社会工程攻击，它利用 LLMs 来生成针对其他 LLMs 的对抗性

提示 (Chao et al., 2023)。我们取了 10 组翻译后的提示的平均值，并获得了以下结果。

首先，我们在三种不同的语言中测量了对抗性提示在 RTT 前后句子的长度。我们观察到，RTT 提示比原始提示短 6-7%，这表明发生了泛化。

其次，我们计算了不在牛津 3000 词汇表中的单词数量。牛津 3000 词汇表包含根据它们在牛津英语语料库中的频率和对英语学习者的相关性选出的 3000 个常用英文单词 (Oxford, 2023)。我们发现，在 RTT 提示中，非牛津 3000 词汇的数量减少了近 20%。这表明 RTT 有助于使用更能传达对抗性提示背后有害行为的一般术语。

3.2 实时翻译的语种数量和类型

我们接下来研究在 RTT 过程中需要多少种语言来进行翻译以实现鲁棒的攻击缓解。实验设置是针对 Vicuna 的 PAIR 攻击。为了保持高质量的翻译，我们使用 Google Translate API，这是市场上最准确的机器翻译算法之一 (Google, 2023)。Google Translate 还帮助维持 LLMs 在所有输入应用 RTT 后的性能。我们使用 10 组翻译数据来计算平均实验结果。

图 2 显示了我们在 RTT 中使用更多语言时 ASR 的下降。我们将涉及 x 种语言的 RTT 表示为 RTTx。我们连续将原始对抗攻击翻译成随机语言，然后将其回译成英语。RTT1 将 ASR 从 0.98 减少到 0.52，这几乎相当于减轻了 50% 的攻击。随着所用语言数量的增加，ASR 下降，并在 RTT3 时稳定下来，此时 ASR 下降至 0.67。

此外，我们研究了目标语言对 RTT 的影响。语言最显著的特征之一是其语系，基于地理和历史因素分类并代表共享的语言特性 (McMahon and McMahon, 2005)。我们假设使用不同语系的语言会导致更好的泛化，因为翻译过程依赖于更通用的术语。我们将 x 种随机语言的 RTT 以及与英语不同的语言学派系（即非印欧语系）中的 x 种语言的 RTT 分别表示为 RTTxr 和 RTTxd。

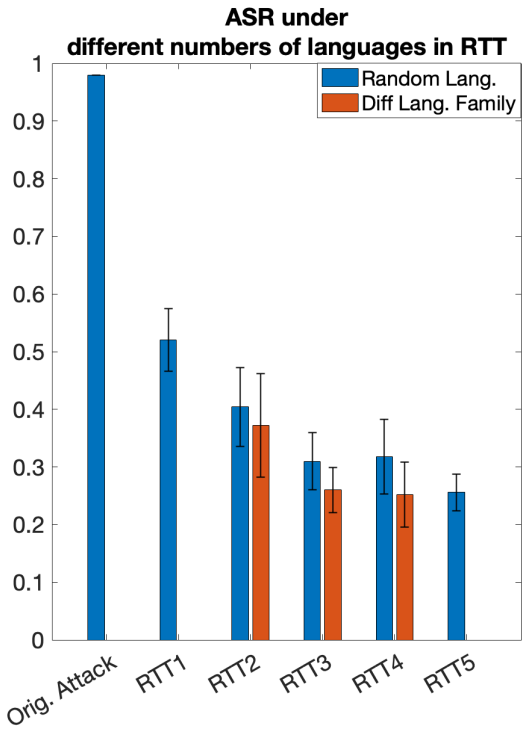

图 2: ASR 在不同数量的语言下 RTT 中减少。误差条是每个 RTT 设置中进行的 10 次实验中的 ASR 的标准差。

在图 2 中，RTT3d 达到了与 RTT5r 相同的 ASR 降低效果。结果 ASR 下降至 0.26，实现了 0.72 的攻击缓解率，并且标准差低于 RTT3r。因此，我们在剩余的实验中使用 RTT3d 作为防御模型，并测试其在不同大语言模型和攻击中的表现。

3.3 大规模语言模型中 RTT 防御的可转移性

为了评估 RTT3d 在大语言模型上的迁移性，我们在 Vicuna、GPT4、Llama2 和 Palm2 上测试了 RTT3d。图 3 显示 RTT3d 在不同的大语言模型中表现出一致的良好性能，平均攻击缓解率为 70%。值得注意的是，在 Llama2 上，RTT3d 将 ASR 降低到 5%以下，并且在 Palm2 上缓解了近 80%的攻击。

3.4 比较 RTT 与其他措辞技巧

我们比较了 RTT3d 与其他改写方法。我们通过 GPT4 执行了各种改写方法，包括对文本

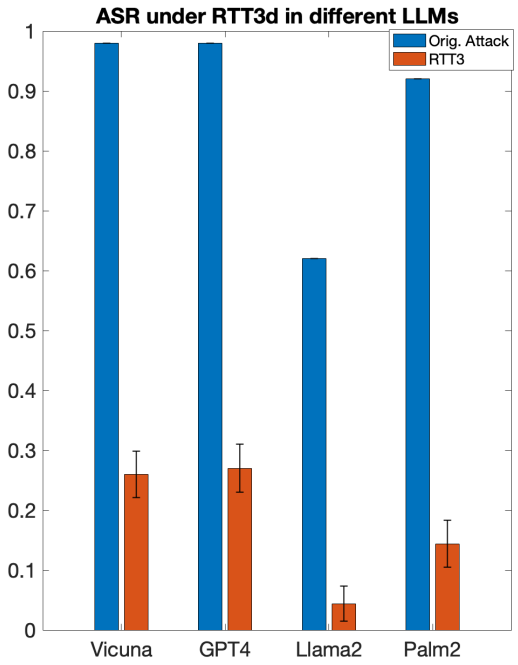

图 3: 不同语言家族的语言在不同大型语言模型中的 RTT3d 的 ASR。误差棒是每个 LLM 中进行的 10 次实验的 ASR 的标准差。

进行改写、将词语替换为其同义词、将词语转换为不同的词形（例如，名词转为动词，动词转为形容词），以及用被动语态重写文本。我们展示了 RTT3d 在抵御攻击方面比其他技术高出 10-30%（附录 A.2）。

当我们进行上述改写实验时，发现 GPT4 有时未能遵循我们的改写要求，只是输出了原文。请注意，在所有实验中记录数据之前，我们手动确保了 GPT4 的改写的准确性。

3.5 比较 RTT 与 SmoothLLM

SmoothLLM 目前是针对 PAIR 攻击 (Robey et al., 2023) 的最强且首次尝试的防御措施。它通过随机扰动输入并使用多个副本，采用集成方法来检测对抗性攻击。借助 SmoothLLM，Vicuna 上的 PAIR ASR 从 0.92 下降到约 0.5。相比之下，RTT3d 将 ASR 降至 0.26，几乎达到了 SmoothLLM 性能的两倍。

3.6 RTT 在其他对抗攻击中的表现

我们测试了 RTT3d 对抗 MathAttack 的效果，MathAttack 是一种针对 LLMs 数学解题能力的社交工程单词级攻击 (Zhou et al., 2023)。我们在 GPT4 上测试了 300 个 MathAttack 提示词，并且 RTT3d 成功防御它们的缓解率为 40%。值得注意的是，我们是第一个提出针对 MathAttack 的防御措施的研究团队。此外，我们还测试了 RTT3d 对抗在 Vicuna 上的贪婪坐标梯度攻击的效果，这是一种使用无意义对抗后缀 (Zou et al., 2023) 的最先进的对抗性攻击。RTT3d 实现了超过 70% 的缓解率，表明它可以转移到不同类型的对抗性攻击上。

3.7 良性输入上的往返时间

我们进行了实验以评估 RTT3d 作为无差别预处理技术是否对良性查询的输出质量产生影响。我们使用了 GSM8K 数据集，该数据集包含 8,500 个面向小学学生的数学应用题 (Cobbe et al., 2021)。由于解决数学应用题需要 LLMs 充分理解并分析输入文本，因此该数据集非常适合评估 RTT3d 预处理后 LLMs 的性能。因此，即使在 RTT3d 过程中语义意义或逻辑推理发生微小变化，也会导致解决方案错误。

我们从 GSM8K 测试集中随机抽取了 500 个问题来评估 RTT3d 对 GPT-4 性能的影响。RTT3d 保留了 GPT-4 超过 80%的原始性能，在实施前和实施后，GPT-4 分别正确回答了 435 个和 357 个问题。我们得出结论，对于良性输入，RTT3d 对 LLM 性能具有潜在的低影响。

4 结论

我们提出了往返翻译方法来防御社会工程学攻击。RTT 防御对对抗提示进行释义和泛化，有助于揭示任何潜在的有害行为。我们的方法在 PAIR 攻击中实现了超过 70% 的攻击缓解效果，超越了当前可用的最强防御措施。我们也是首次尝试防御 MathAttack，并实现了近 40% 的攻击缓解。RTT 还表现出在不同语言模型上的强迁移性。

虽然这项工作突出了一种有前景的新防御策略，未来的工作包括测试其他翻译算法，并验证 RTT 在处理英语以外的对抗性提示时是否能保持其性能。我们还可以组合多个 RTT 提示以创建一个聚合处理过的提示，类似于 SmoothLLM。

5 限制

在我们对 RTT 防御的最佳配置进行调查时，我们测试并验证了一种翻译算法（即 Google Translate）在其对抗攻击的防御性能以及对良性输入的影响方面的有效性。因此，当使用具有不同配置的不同翻译算法时，RTT 的结果可能会有所不同。

此外，在研究 RTT 对良性输入的影响时，我们考察了小学级别的数学应用题。需要注意的是，当将 RTT 应用于其他查询和更高水平的问题输入时，其影响可能会有所不同。为了提高 RTT 作为预处理技术在 LLMs 中的可靠性，有必要使用更多样化的数据集进行测试。

References

Milam Aiken and Mina Park. 2010. The efficacy of round-trip translation for mt evaluation. *Translation Journal*, 14(1):1–10.

Samuel Barham and Soheil Feizi. 2019. Interpretable adversarial training for text. *arXiv preprint arXiv:1905.12864*.

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J Pappas, and Eric Wong. 2023. Jailbreaking black box large language models in twenty queries. *arXiv preprint arXiv:2310.08419*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Google. 2023. Google translate api.

April McMahon and Robert McMahon. 2005. *Language classification by numbers*. Oxford University Press.

Oxford. 2023. Oxford learner’ s dictionary.

Alexander Robey, Eric Wong, Hamed Hassani, and George J Pappas. 2023. Smoothllm: Defending large language models against jailbreaking attacks. *arXiv preprint arXiv:2310.03684*.

Motoki Sato, Jun Suzuki, Hiroyuki Shindo, and Yuji Matsumoto. 2018. Interpretable adversarial perturbation in input embedding space for text. *arXiv preprint arXiv:1805.02917*.

Zihao Zhou, Qiufeng Wang, Mingyu Jin, Jie Yao, Jianan Ye, Wei Liu, Wei Wang, Xiaowei Huang, and Kaizhu Huang. 2023. Mathattack: Attacking large language models towards math solving ability. *arXiv preprint arXiv:2309.01686*.

Terry Yue Zhuo, Qiongkai Xu, Xuanli He, and Trevor Cohn. 2023. Rethinking round-trip translation for machine translation evaluation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 319–337.

Andy Zou, Zifan Wang, J Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*.

A 附录

A.1 假设证明对于 RTT

图 4 和 5 显示了对抗性提示的长度以及经过 RTT 后存在的非牛津 3000 词汇的数量。

A.2 RTT 与其他改写技术的比较

图 6 比较了 RTT3d 的攻击缓解与其他不同措辞技术的差异。

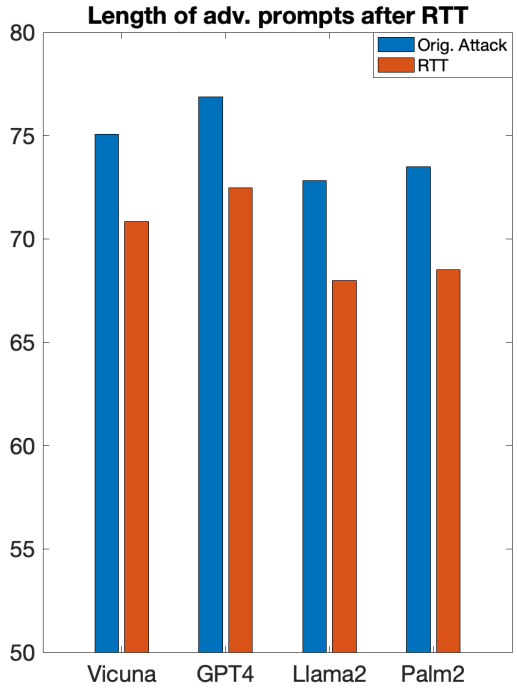

图 4: 对抗提示在 RTT 后的长度。对抗提示由 PAIR 攻击生成。每个 RTT 长度的数据是通过平均 10 组 RTT 提示获得的。

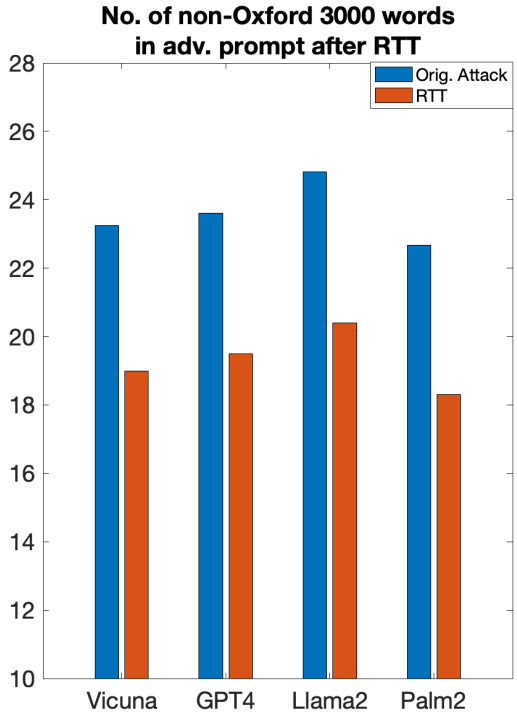

图 5: 对抗提示在 RTT 后的长度。对抗提示由 PAIR 攻击生成。每个 RTT 长度的数据通过平均 10 组 RTT 提示获得。

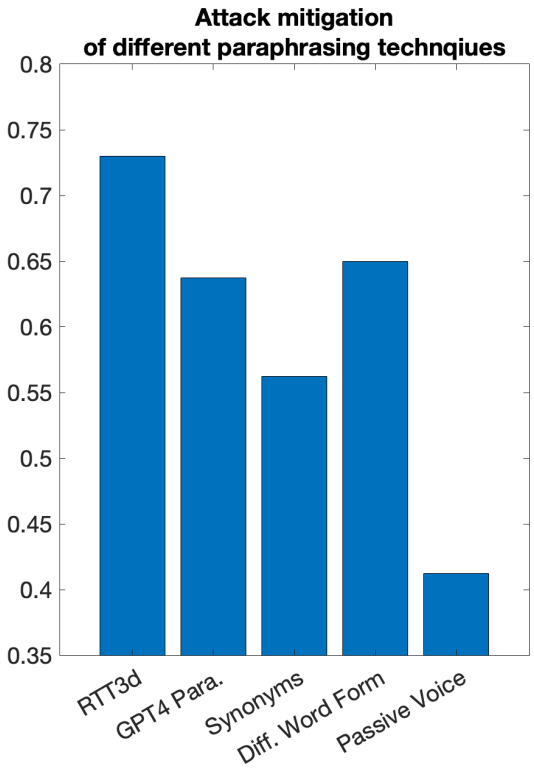

图 6: 不同改写技术与 RTT3d 相比的攻击缓解效果。