软件学报 ISSN 1000-9825, CODEN RUXUEW
2025,36(10):4710−4734 [doi: 10.13328/j.cnki.jos.007299] [CSTR: 32375.14.jos.007299]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

中文对抗攻击下的 ChatGPT 鲁棒性评估*

张云婷[1], 叶 麟[1], 李柏松[2], 张宏莉[1]

[1](哈尔滨工业大学 网络空间安全学院, 黑龙江 哈尔滨 150001)

[2](安天实验室, 黑龙江 哈尔滨 150023)

通信作者: 叶麟, E-mail: hityelin@hit.edu.cn

摘 要: 以 ChatGPT 为代表的大语言模型 (large language model, LLM) 因其强大的自然语言理解和生成能力在各领域中得到广泛应用. 然而, 深度学习模型在受到对抗样本攻击时往往展现出脆弱性. 在自然语言处理领域中, 当前对抗样本生成方法的研究通常使用 CNN 类模型、RNN 类模型和基于 Transformer 结构的预训练模型作为目标模型, 而很少有工作探究 LLM 受到对抗攻击时的鲁棒性并量化 LLM 鲁棒性的评估标准. 以中文对抗攻击下的 ChatGPT 为例, 引入了偏移平均差 (offset average difference, OAD) 这一新概念, 提出了一种基于 OAD 的可量化的 LLM 鲁棒性评价指标 OAD-based robustness score (ORS). 在黑盒攻击场景下, 选取 9 种基于词语重要性的主流中文对抗攻击方法来生成对抗文本, 利用这些对抗文本攻击 ChatGPT 后可以得到每种方法的攻击成功率. 所提的 ORS 基于攻击成功率为 LLM 面向每种攻击方法的鲁棒性打分. 除了输出为硬标签的 ChatGPT, 还基于攻击成功率和以高置信度误分类对抗文本占比, 设计了适用于输出为软标签的目标模型的 ORS. 与此同时, 将这种打分公式推广到对抗文本的流畅性评估中, 提出了一种基于 OAD 的对抗文本流畅性打分方法 OAD-based fluency score (OFS). 相比于需要人类参与的传统方法, 所提的 OFS 大大降低了评估成本. 分别在真实世界中的中文新闻分类和情感倾向分类数据集上开展实验. 实验结果在一定程度上初步表明, 面向文本分类任务, 对抗攻击下的 ChatGPT 鲁棒性分数比中文 BERT 高近 20%. 然而, ChatGPT 在受到对抗攻击时仍会产生错误预测, 攻击成功率最高可超过 40%.

关键词: 深度神经网络; 对抗样本; 大语言模型; ChatGPT; 鲁棒性

中图法分类号: TP18

中文引用格式: 张云婷, 叶麟, 李柏松, 张宏莉. 中文对抗攻击下的ChatGPT鲁棒性评估. 软件学报, 2025, 36(10): 4710–4734. http://www.jos.org.cn/1000-9825/7299.htm

英文引用格式: Zhang YT, Ye L, Li BS, Zhang HL. Robustness Evaluation of ChatGPT Against Chinese Adversarial Attacks. Ruan Jian Xue Bao/Journal of Software, 2025, 36(10): 4710–4734 (in Chinese). http://www.jos.org.cn/1000-9825/7299.htm

Robustness Evaluation of ChatGPT Against Chinese Adversarial Attacks

ZHANG Yun-Ting[1], YE Lin[1], LI Bai-Song[2], ZHANG Hong-Li[1]

[1](School of Cyberspace Science, Harbin Institute of Technology, Harbin 150001, China)

[2](Antiy Labs, Harbin 150023, China)

Abstract: Large language model (LLM) like ChatGPT has found widespread applications across various fields due to their strong natural language understanding and generation capabilities. However, deep learning models exhibit vulnerability when subjected to adversarial example attacks. In natural language processing, current research on adversarial example generation methods typically employs CNN-based models, RNN-based models, and Transformer-based pre-trained models as target models, with few studies exploring the robustness of LLMs under adversarial attacks and quantifying the evaluation criteria of LLM robustness. Taking ChatGPT against Chinese adversarial

* 基金项目: 黑龙江省重点研发计划 (2023ZX01A19)
收稿时间: 2024-03-29; 修改时间: 2024-06-18, 2024-08-26; 采用时间: 2024-10-02; jos 在线出版时间: 2025-02-26
CNKI 网络首发时间: 2025-02-26

attacks as an example, this study introduces a novel concept termed offset average difference (OAD) and proposes a quantifiable LLM robustness evaluation metric based on OAD, named OAD-based robustness score (ORS). In a black-box attack scenario, this study selects nine mainstream Chinese adversarial attack methods based on word importance to generate adversarial texts, which are then employed to attack ChatGPT and yield the attack success rate of each method. The proposed ORS assigns a robustness score to LLMs for each attack method based on the attack success rate. In addition to the ChatGPT that outputs hard labels, this study designs ORS for target models with soft-labeled outputs based on the attack success rate and the proportion of misclassified adversarial texts with high confidence. Meanwhile, this study extends the scoring formula to the fluency assessment of adversarial texts, proposing an OAD-based adversarial text fluency scoring method, named OAD-based fluency score (OFS). Compared to traditional methods requiring human involvement, the proposed OFS greatly reduces evaluation costs. Experiments conducted on real-world Chinese news and sentiment classification datasets to some extent initially demonstrate that, for text classification tasks, the robustness score of ChatGPT against adversarial attacks is nearly 20% higher than that of Chinese BERT. However, the powerful ChatGPT still produces erroneous predictions under adversarial attacks, with the highest attack success rate exceeding 40%.

Key words: deep neural network (DNN); adversarial example (AE); large language model (LLM); ChatGPT; robustness

得益于硬件资源算力的不断增强, 大语言模型 (large language model, LLM) 在生产和生活中的应用日益广泛. 与传统的深度学习 (deep learning, DL) 模型不同, LLM 的参数往往能够达到十亿的数量级. 海量的参数使得 LLM 相比于传统 DL 模型具有更强的自然语言处理 (natural language processing, NLP) 能力, 能够完成更多种类、更高难度、更加多元的任务. ChatGPT 作为 LLM 发展过程中的里程碑式的模型, 其在多种 NLP 任务中都达到了最先进的水平.

然而, 最近的研究表明, 各种主流的深度神经网络 (deep neural network, DNN) 在受到对抗攻击时都会显露出不同程度的脆弱性[1−4]. 受到对抗攻击的 DL 模型被称为目标模型. 逃逸攻击是一种最常见的对抗攻击方式, 其作用于目标模型训练后的测试阶段, 敌手往往通过构造对抗样本 (adversarial example, AE) 来实现逃逸攻击. AE 是一种向原始良性样本中添加人类难以察觉的微小扰动后得到的恶意样本, 其能够触发 DL 模型的错误预测. 换言之, AE 能够在人类正确理解其内容的情况下欺骗 DL 模型. 对 AE 的研究不但能够了解当前主流 DL 模型的弱点, 评估这些 DL 模型在对抗攻击条件下的鲁棒性[5], 还能为后续防御方法的设计打下坚实的基础.

AE 最初在计算机视觉 (computer vision, CV) 领域中被发现并明确定义[1]. Goodfellow 等人[2]发现 AE 通常能以高置信度欺骗目标模型致使其误分类. 图 1 展示了 AE 研究领域中最具代表性的一个例子. 通过图 1 可以了解到, 向原始的熊猫图片加入人类难以察觉的微小扰动后, 目标 DL 模型以 99.3% 的置信度将大熊猫的 AE 图片分类为长臂猿. 此外, Goodfellow 等人还通过“深度神经网络在高维空间的线性行为”这一假说试图解释 AE 的存在性. 与此同时, 他们也对 AE 的可迁移性进行了初步验证, 并通过将原始数据与 AE 混合在一起作为新的训练集, 对目标模型进行对抗训练, 以提升目标模型在 AE 攻击下的鲁棒性.

图 1 CV 领域中的 AE 实例[2]

近期的一些工作指出, NLP 领域中的 DL 模型也会受到 AE 的攻击[3,4]. 图 2 展示了一个被深度学习模型误分类的对抗文本实例. 从图 2 中可以看出, 原始文本为积极评论, 但将原始文本中的“价格便宜”改为其同义词“价廉”后, DL 模型会将该对抗文本误分类为消极评论.

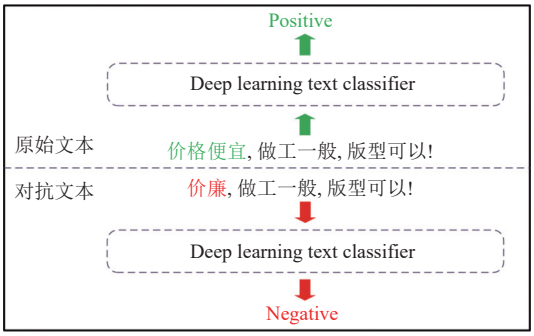


图 2 NLP 领域中的 AE 实例

相比于图像数据, 使用文本数据生成对抗文本的过程中存在如下难点: (1) 图像数据是连续的, 而文本数据则是离散的, 这就导致了向文本数据中加入人类难以察觉的扰动是一件较为困难的事情[6]; (2) CV 领域中一些经典的扰动度量方法无法直接应用到 NLP 领域中[7], 对于一段文本来说, 微小改动可能会直接改变整段文本的语义, 所以需针对文本数据重新设计相似度评估方法.

由前述内容可知, CV 领域中的 AE 生成方法无法直接迁移到 NLP 领域中, 因此, 相应的对抗文本生成方法应运而生[3,4,6,8−23]. 当前的对抗文本生成方法用扰动方式可以简单分为 3 类, 分别是字符级、词语级和句子级的对抗文本生成方法. 由于经过字符级和词语级扰动生成的对抗文本的改动往往较小, 因此目前的大多数研究主要集中在这两类对抗文本生成方法上. 字符级和词语级的对抗文本生成方法通常可以被形式化为搜索问题[23], 搜索方法则成为此类方法的研究重点. 当前使用最广泛的搜索方法为基于词语重要性的贪心搜索算法. 这种搜索方式旨在找出原始文本中的重要词语, 并对这些词语按顺序进行扰动. 许多对抗文本生成方法都在基于词语重要性的对抗文本生成框架下设计[3,4,10−15,17−21]. 该框架主要分为两个阶段, 分别是排序阶段和扰动阶段. 排序阶段的目的是使用打分方法为原始文本中每个词语的重要性打分, 并按分数从高到低将这些词语排序. 在扰动阶段中, 按顺序对这些词语依次添加扰动, 以生成最终的对抗文本. 针对文本数据添加的扰动通常可具象化为对字符 (汉字) 或词语的增加、删除和替换操作.

得益于 AE 生成方法的涌现, 对于目标模型的对抗鲁棒性评估工作受到了广泛关注. 对抗鲁棒性评估工作主要集中在使用连续输入数据的 CV、语音信号处理等领域, 且在这些领域中已经初步形成了一套较为全面的评估体系. 该评估体系主要包括两个维度, 分别是范数选择以及度量指标, 文献 [24] 分别对上述两个指标进行了具体介绍. 其中, 范数选择主要反映了对于单一 AE 中添加的扰动, 其幅度的定义方式. 常见的范数选择主要包括 L_0 范数、L_1 范数、L_2 范数及 L_∞ 范数等. 而度量指标则主要用来评估一组 AE 对目标模型的攻击性能. 其中, 一组 AE 对目标模型的攻击成功率及其扰动统计量为最常见的两种度量指标. 当前在 CV、语音信号处理等领域的对抗鲁棒性评估体系中, 通常将上述两个维度进行组合, 形成最终的评估方法. 其中, 最常见的组合有两种. 第 1 种需要统计同一范数下的 AE 攻击成功率; 而另一种则统计成功攻击目标模型的 AE 中, 其扰动幅度的均值与中位数[24].

然而, CV、语音信号处理等领域的对抗鲁棒性评估方法同样无法直接应用于 NLP 领域中. 与此同时, 当前大部分对抗文本生成方法的目标模型均为 CNN 类模型[4,12−14,17,19]、RNN 类模型[3,4,8,9,12,14,17−19]以及基于 Transformer 结构的预训练模型[9−11,15,16,20−22], 很少有工作研究当前流行的 LLM 在对抗攻击条件下的鲁棒性并量化其评估标准. 本文将 ChatGPT 作为目标模型进行文本对抗攻击, 引入了一个新的数学概念, 偏移平均差 (offset average difference, OAD), 并利用面向中文文本分类任务的多种对抗文本生成方法对目标模型的攻击成功率, 基于 OAD 设计了一种 LLM 鲁棒性打分方法 OAD-based robustness score (ORS). 该方法全面考量了黑盒场景下 9 种基于词语重要性的中文对抗文本生成方法对 ChatGPT 的攻击成功率. 这 9 种方法不仅包含所有主流的中文对抗文本生成方法, 同时也包含从英文对抗攻击方法中迁移的方法. ORS 利用上述 9 种方法的攻击成功率, 基于 OAD 计算出 ChatGPT 在每种方法下的鲁棒性得分, 最终可以得到 ChatGPT 在对抗文本攻击下的平均鲁棒性分数. 该方法不但

全面评估了 ChatGPT 在主流中文对抗攻击方法下的鲁棒性, 还量化了 ChatGPT 鲁棒性的评价指标. 在同等的对抗攻击条件下, 使用所提的 ORS 便于 ChatGPT 和其他目标模型的鲁棒性进行对比. 此外, 本文除了为输出为硬标签的 LLM 设计可量化的鲁棒性评估方法外, 同时也对输出为软标签的目标模型设计了类似的鲁棒性评估方法. 在硬标签中使用的攻击成功率这一指标的基础上, 对于输出包含置信度的软标签, 本文加入了以高置信度误分类的对抗文本占比这一指标, 同样基于 OAD 计算其鲁棒性得分, 对输出为软标签的目标模型的鲁棒性进行更全面地评估. 与此同时, 本文将 OAD 的应用范围从目标模型的鲁棒性评估扩展到对抗文本的流畅性评估中, 提出了一种基于 OAD 的对抗文本流畅性打分方法 OAD-based fluency score (OFS). 不同于需要人类参与的传统方法, OFS 利用了 ChatGPT 对自然语言的理解能力, 将流畅性评估过程自动化, 大大降低了人力物力.

本文的主要贡献如下.

(1) 本文面向中文文本分类任务, 引入一个新的概念 OAD, 并基于 OAD 提出对抗攻击下的 DL 模型鲁棒性打分方法 ORS. ORS 基于 9 种主流对抗文本生成方法对目标模型的攻击成功率, 计算目标模型在对抗攻击下的鲁棒性分数, 量化目标模型的鲁棒性. 此外, 本文分别设计面向硬标签 DL 模型以及面向软标签 DL 模型的 ORS, 以满足不同威胁模型下目标模型的鲁棒性评估需求.

(2) 本文将 OAD 的应用范围扩展到对抗文本流畅性评估中, 提出一种基于 OAD 的对抗文本流畅性打分方法 OFS. OFS 将 ChatGPT 强大的自然语言理解能力与基于 OAD 的打分方法结合, 实现对抗文本流畅性的自动化打分, 整个过程无需人工参与, 大大降低了评估成本.

(3) 本工作分别在物理世界中两个真实存在的情感分类和新闻分类数据集上开展对 ChatGPT 和中文 BERT 模型的对抗攻击实验, 以评估 ChatGPT 在对抗攻击条件下的鲁棒性, 并将其鲁棒性与中文 BERT 进行对比. 实验结果初步表明, 强大的 ChatGPT 面对中文对抗攻击仍然表现出较高程度的脆弱性, 所用的对抗攻击方法对其的最高攻击成功率超过 40%. 与此同时, 基于所提的 ORS 分别计算 ChatGPT 和中文 BERT 模型的鲁棒性, 可发现前者的平均鲁棒性分数比后者高约 20%, 这说明面向文本分类任务时, ChatGPT 在一定程度上拥有比中文 BERT 更强的鲁棒性.

本文第 1 节简要回顾当今主流的词符级对抗文本生成方法. 第 2 节则为对抗文本生成过程进行形式化定义, 并给出适用于本工作的威胁模型. 第 3 节分别介绍我们提出的 OAD 在 LLM 鲁棒性评估以及对抗文本流畅性评估中的具体应用, 详细介绍它们对应的两种打分方法 ORS 和 OFS. 第 4 节展示本工作的实验设置及具体的实验结果与分析, 并对不同提示词下的 ChatGPT 的分类结果进行了讨论. 第 5 节则简单凝练地总结全文, 并指出未来工作中的研究重点.

1 相关工作

本文提出的鲁棒性评估方法 ORS 是基于多种对抗文本生成方法的攻击成功率设计的, 对抗文本生成方法在其中起到了重要作用. 如前所述, 字符级和词语级的对抗文本生成方法是当前主流的文本对抗攻击方法, 可以将它们统称为词符级对抗文本生成方法. 而词符级对抗文本生成方法可以形式化为搜索问题[25], 因此对于词符级对抗文本生成方法来说, 其包含的较为重要的模块有两个, 分别是搜索算法和扰动方法. 下面按这两个部分依次对以往面向文本分类任务的黑盒词符级对抗文本生成方法进行简要的总结回顾.

目前在该领域中应用较为广泛的搜索算法主要可分类为两大类, 一类为基于种群的优化算法[8,9], 另一类为贪心搜索算法及其变种[3,4,10–21]. 其中, 基于种群的优化算法在文本对抗中的应用较为少见, 目前比较具有代表性的研究介绍如下. Alzantot 等人[8]将遗传算法应用于对抗文本的生成过程中, 而他们使用遗传算法的主要目的在于设计一种不依赖梯度的对抗文本生成方法. 相比于依赖梯度的白盒方法[23]来说, Alzantot 等人显然更能模拟真实世界中的攻击. Zang 等人[9]则将粒子群优化 (particle swarm optimization, PSO) 算法进行了改动, 使其适应于在离散空间中的搜索方式. 通过控制变量实验的实验结果可知, 在扰动方式一样的情况下, 相比于遗传算法, 基于 PSO 算法的对抗文本生成方法对目标模型有更高的攻击成功率.

与基于种群的优化算法相比, 应用贪心算法及其变种的文本对抗攻击方法更为常见. 因为这些方法的实现方式更加简单, 且往往仅需更少的时间就能够生成对抗文本. 在各种类型的贪心算法中, 应用最多的搜索算法为基于词语重要性的贪心搜索. 该算法的优势在于大大提升了对抗文本生成过程的可解释性[13]. 这种搜索算法的核心在于词语重要性打分方法的设计, 合适的打分方法能够更好地捕捉到每个文本中与其对应标签强相关的词语. 若着重对这些词语进行扰动, 则能够在扰动尽量少词语的情况下成功生成对抗文本. 当前使用最广泛的词语重要性打分方法为 delete score (DS) 方法[3,14,17,18]. 该方法采用删去某词语前后目标模型输出的置信度之差作为该词语的重要性分数. 其他的词语重要性打分方法大多数均在 DS 方法的基础上加以改动. 如 Gao 等人[3]提出的 temporal score (TS)、temporal tail score (TTS) 以及 combined score (CS) 方法. 这 3 种方法分别选取某词语之前的所有词语、某词语之后的所有词语以及使用将前两种方法进行结合的方式, 同样利用与 DS 类似的置信度作差的方法来计算该词语的重要性分数. 然而, 这 3 种方法仅适用于 RNN 类模型, 而 DS 却能适用于所有类型的目标模型. 王文琦等人[4]面向倾向性分类任务, 设计了一种 TF-IDF score (TIS). 该方法使用 TF-IDF 技术提取原始文本中可能含有情感倾向的词, 并将 DS、TS 和 TIS 这 3 种打分方法进行结合, 计算最终的词语重要性得分. Jin 等人[10]则额外考虑了删去某词语后预测类别是否发生变化这一点, 以此进行分类讨论. 若原标签和删去该词语后的预测标签不一致, 则分别在原标签和改动后的标签上使用 DS 方法, 并将得到的值作和, 当作这种情况下该词语最终的重要性得分. Li 等人[11]将 BERT 模型[26]作为目标模型, 设计了攻击方法 BERT-Attack. 在 BERT-Attack 中, 将删去的词用 [MASK] 进行替代, 并将使用完整文本的输出置信度与使用掩码后文本的输出置信度作差, 得到的结果即为被掩码词语的重要性分数. 然而该方法仅适用于 BERT 模型及其变种, 并不适合不包含掩码语言模型 (masked language model, MLM) 这一机制的目标模型. Ren 等人[12]提出了一种名为 PWWS 的对抗文本生成方法. 在 PWWS 中, 原文本的词语重要性打分方法与 BERT-Attack 中使用的打分方法相似, 将 [MASK] 用不在预定义字典中的词语替代, 然后也用与 DS 一样的方法计算原始文本中每个词语的重要性分数. 然而, 与上述提到的所有方法不同的是, PWWS 中最终的词语替换顺序并非仅由原文本中的词语重要性分数决定, 而是由该分数与词语替换后得到的文本的分类置信度变化共同决定. 该方法比起 BERT-Attack 中使用的打分方法适用范围更广, 适用于所有类型的目标模型; 但相比起其他仅由原文本中词语重要性决定词语替换顺序的方法, 该方法需要更大的计算量以及更多的目标模型访问次数. 而 Xu 等人[13]则提出了一种与上述提到的方法有本质区别的词语重要性打分方法, layer-wise relevance propagation (LRP). LRP 是一种从 CV 领域迁移到文本对抗领域的打分方法. 该方法的提出为后续打分方法的设计带来新的启发, 即除了基于 DS 打分方法的变种, 也可以从其他领域中迁移合适的方法并加以改进, 作为基于词语重要性对抗文本生成框架中的打分方法.

除了上面介绍的搜索算法外, 扰动方法也是词符级对抗文本生成方法中必不可少的一部分. 扰动方法与语言息息相关, 通常需要根据语言特点来设计相应的扰动方法. 下面分别对常见的英文扰动方法和中文扰动方法进行介绍. 最初的英文扰动方法以字符级扰动为主. Gao 等人[3]提出了字符级的增加、删除、替换随机字母以及交换相邻字母这 4 种扰动方法. Li 等人[14]在此基础上, 改进了上述删除、交换以及字符级替换的扰动方法. 他们将前两种扰动方法允许扰动的字母限定在除了首字母和尾字母外的其他字母, 而替换则是使用与原字母相似的字符进行替换. 上述改动的目的旨在设计出人类难以察觉的扰动. 与此同时, 他们还设计了一种新的字符级扰动方法, 即插入空格法. 插入空格法是将空格插入一个英文单词的随机位置, 由于英文是以空格进行分词的, 因此该方法可以将一个单词拆分为两个不同的词.

相比于基于相似外形的字符级扰动来说, 基于相似语义的词语级扰动往往能保留与原始文本更高的语义相似度. 与此同时, 词语级扰动能够提供更多的候选词, 大大增加了攻击成功率. 此外, 字符级扰动更容易被拼写检查检测出来, 而词语级扰动则不受拼写检查的影响. 基于上述优点, 词语级扰动逐渐替代字符级扰动成为主流的扰动方式. 具体的英文词语级扰动介绍如下. Li 等人[14]除了提出上述的 4 种字符级扰动方法外, 还提出了一种词语级的替换扰动方法. 该方法在词向量空间中寻找与原单词距离最近的前 k 个词. 与 Li 等人[14]提出的词语级替换扰动类似, Jin 等人[10]同样在词嵌入空间中选择与原词语最相似的前 k 个词语, 但为了过滤原词语的反义词, Jin 等人加入了 counter-fitting[27]的机制确保了替换前后的语义一致性. Ren 等人[12]利用 WordNet 构造原词语的同义词候选集,

相比于在向量空间中寻找原词语的替换词, 前者使用基于专家知识构造的同义词集, 能够得到更加精准的同义词, 无需反义词过滤这一步骤. 与 Ren 等人使用的方法类似, Zang 等人[9]也使用了基于专家知识构造的词集来挑选与原词语含义相似的词. 但与前者不同的是, 后者使用 HowNet 词典寻找与原词语具有相同义原的词. 义原是最小的语义单位, 其无法再进行进一步的分割. 相比于传统的同义词, 基于义原寻找原词的替换词能够为原词语找到更多的候选词, 这在一定程度上提高了攻击成功率; 与此同时, 相比于词嵌入空间中与原词语距离相近的候选词, 基于义原所找到的替换词同时具备了传统同义词词义精准的优势. 由此可见, 基于义原的词语替换方法在传统同义词替换法以及词嵌入替换法之间达到了一个合适的平衡. 然而, 上述词语级的扰动方法仅考虑了原词语的语义, 却并未考虑上下文的语境. 基于此种考虑, 许多工作[11,15,16]使用生成能力较强的 BERT 模型根据上下文语境来生成原词语的候选词集. 这类工作基于 BERT 模型中的 MLM 机制来完成替换词集的生成, 但在细节上均有一些各自不同的特点. Li 等人[11]提出的 BERT-Attack 重点考虑了单个词语以及会被 BERT 分词分成多个子词的词语. 其中, 后者需要考虑所有替换子词的组合, 并通过 BERT 的分词机制将每种子词组合还原成对应的词语. Garg 等人[15]提出的 BAE 则提出了 3 种基于 MLM 的扰动方法, 除了简单的替换操作外, 还可以在某个英文单词的左边或右边插入一个额外的词, 通过 MLM 机制来预测这个被插入的词语. Li 等人[16]提出的 CLARE 除了替换和插入还提出了一种新的归并策略. 该策略具体是将两个英文单词使用一个掩码位遮蔽, 并用 MLM 机制对其进行预测, 从而达到将原来的两个词语归并为一个词语的目的.

汉语作为全世界范围内使用人数较多的一种语言, 也有许多工作根据汉语的语言特点设计了许多不同的中文扰动方法. 与英文中基于相似外形以及相似语义而设计出的扰动方法不同, 最初的中文扰动方法是一种基于相似发音设计的方法[4]. 王文琦等人[4]基于汉语中特有的拼音方案, 汉语拼音, 设计了一种词语级的中文扰动方法. 该方法基于拼音构造声母韵母相同但声调可能不同的词语作为原词语的替换词. 而 Zhang 等人[17]对这种扰动方法进行了改进, 他们将相似的平翘舌发音以及前后鼻音也考虑在内, 设计出一种基于拼音的谐音词替换方法. 与此同时, Zhang 等人还将部分英文扰动方法迁移到中文中并进行改进. 他们将 Li 等人[14]提出的相邻字母交换、词语级替换方法分别迁移至中文并改进为适用于中文的 Shuffle 方法和 Synonyms 方法. 其中, Shuffle 方法是打乱一个词语中的汉字顺序, 而 Synonyms 方法则是使用了专家构造的同义词典而非使用词嵌入的方式来找原词语的同义词. 除此以外, Zhang 等人也针对汉语的语言特点设计了两种新的字符级中文扰动方法, 分别是 Splitting-Character (SC) 以及 Glyph. 这两种方法都是基于相似外形设计的方法, 其中 SC 将一个汉字拆解为组成它的部首, 而 Glyph 则是使用 CNN 来找到原汉字的形近字. 在同一时期, Nuo 等人[18]也提出了与 Shuffle 和 SC 相似的方法, 但除了这两种方法外, 他们还提出了向词语中插入特殊字符的中文扰动方法. 而仝鑫等人[19]在 Cheng 等人的基础上, 提出了繁体字替换和拼音改写的中文扰动方法. 其中繁体字替换是将简体字变为与之对应的繁体形式, 而拼音改写则是将原来的中文词语全部转化为其对应的拼音形式. Ou 等人[20]将英文扰动方法迁移至中文, 并对上述提到的一些中文扰动方法进行了改进. 他们将英文中基于义原的方法[9]迁移至中文, 并将基于拼音的同音词替换[4]与拼音改写[19]的方法均进行了相应的改进. 对于基于拼音的同义词替换方法, 他们放宽了对同音词替换的限制, 替换后的汉字无需能够组成一个有确切含义的词语, 但人类往往能够通过上下文猜测出原始文本要表达的意思. 这种改动的优势在于, 能够生成更多的词语作为原词语的候选替换词, 一定程度上增加了攻击成功率. 而对于拼音改写方法, 他们将其改进为随机选择原始中文词语中的任意一个汉字进行拼音改写. 改进后的方法能够增加对抗文本的可读性, 让阅读对抗文本的人类更容易理解原文本的含义. 张云婷等人[21]则将英文中最新的基于 BERT-MLM 的扰动方法迁移至中文并针对汉语的语言特点对其进行改进. 不同于英文中基于 BERT-MLM 机制所设计的单词替换、插入及归并策略, 他们提出了 N to 1 和 N to 2 这两种适用于汉语的替换策略, 使该方法在生成尽量多的候选词的条件下, 最大程度地提升对抗文本的流畅性及与原文的相似度.

上述提到的所有工作均为输出为包含分类置信度的软标签, 这些工作设计对抗文本生成方法时, 均需要用到目标模型的分类置信度. 而 He 等人[22]则设计了一种输出为仅包含预测标签的硬标签时的对抗文本生成方法. 该方法利用多次询问目标 BERT 模型的方式尝试提取目标模型参数, 构造一个知晓其内部结构和具体参数的白盒 BERT 模型作为目标模型的影子模型进行攻击. 随后使用成功攻击影子模型的对抗文本来攻击目标 BERT 模型.

实验结果表明, 使用这种方法能够以一定的攻击成功率攻击目标模型. 这既说明了输出为硬标签的目标模型也有被攻击的可能, 也同时说明了对抗文本具有一定的可迁移性.

然而, 以上研究均没有以当前流行的 LLM 作为目标模型, 探究其在对抗攻击下的鲁棒性. 本文在前人工作的基础上, 以 ChatGPT 为例, 探究其在不同的中文对抗攻击条件下的鲁棒性. 与此同时, 本工作设计了可量化的鲁棒性评价指标 ORS, 并使用 ORS 计算 ChatGPT 面对各种中文对抗攻击时的鲁棒性分数. 借由这些工作, 本文试图填补对抗文本生成领域在 LLM 方面的研究工作.

2 问题定义

2.1 对抗文本形式化表示

面向文本分类任务的对抗文本定义可形式化表述如下. 给定一个文本分类数据集 $\mathbb{X}$, $\mathbb{X}$ 中总共包含 n 篇文档, 即 $\mathbb{X}=\{\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_n\}$. 其中, 每篇文档均由词向量表示. 这些文档所对应的标签集合为 $\mathbb{Y}$, $\mathbb{Y}$ 中共包含 m 个标签, 标签通常用标量表示, 即 $\mathbb{Y}=\{y_1,y_2,\ldots,y_m\}$. 现有一个深度学习分类器 $f:\mathbb{X}\to\mathbb{Y}$ 是从 $\mathbb{X}$ 到 $\mathbb{Y}$ 的映射. 对于 $\mathbb{X}$ 中的任意一篇文档, 都有且仅有 $\mathbb{Y}$ 中的一个标签与其对应. 对于 $\mathbb{X}$ 中的一篇文档 $\mathbf{x}_i$ $(i=1,\ldots,n)$, 现向 $\mathbf{x}_i$ 中引入一个人类难以察觉的微小扰动 $\Delta\mathbf{x}_i$, 可以得到一个扰动后的文本 $\mathbf{x}_i'$. 扰动后文本 $\mathbf{x}_i'$ 可形式化表示如下:

$$\mathbf{x}_i'=\mathbf{x}_i+\Delta\mathbf{x}_i \tag{1}$$

若加入扰动后的文本 $\mathbf{x}_i'$ 能够触发目标模型产生错误分类, 则称 $\mathbf{x}_i'$ 是 $\mathbf{x}_i$ 的对抗文本. 此时需要引入一个计算文本相似度差异的函数 g 来衡量 $\mathbf{x}_i$ 与 $\mathbf{x}_i'$ 的文本相似度. 对抗文本通常需满足如下条件:

$$f(\mathbf{x}_i)\neq f(\mathbf{x}_i'),\quad \text{s.t.}\quad g(\mathbf{x}_i,\mathbf{x}_i')\leqslant\varepsilon \tag{2}$$

其中, $\varepsilon\in\mathbb{R}$ 且 ε 是 $\mathbf{x}_i$ 和 $\mathbf{x}_i'$ 相似度差异的上限.

公式 (2) 中的形式化描述即为当前大部分工作[3,10,12−16,21]使用的面向文本分类任务的对抗文本的形式化定义. 本文在此基础上, 进一步考虑将对抗文本自身的流畅性作为约束条件. 对于对抗文本的原始定义来说, “人类难以察觉”这一约束条件有两层含义. 第 1 层即为当前大部分工作在形式化表述时均考虑到的文本相似性, 而文本相似性的评估对象为两个, 即原始文本和对抗文本. 本文除了考虑上述第 1 层含义, 也考虑了第 2 层含义, 即对抗文本自身的流畅性. 流畅性的评估对象仅有对抗文本自身, 本文将其考虑进约束条件的原因有以下两点. 一方面, 在物理世界中, 人类阅读对抗文本时往往仅能看到对抗文本而无法看到原始文本. 对抗文本是否能被顺利阅读也是人类是否能觉察到对抗文本的关键因素, 因此将对抗文本自身的流畅性作为约束条件也同样有意义. 另一方面, 虽然当前工作均未将对抗文本的流畅性作为对抗文本定义中形式化表述的约束条件, 但许多工作均通过人类评估的方式评估了对抗文本的流畅性[8−12,14−17,21]. 基于上述两点, 本文将对抗文本自身的流畅性作为约束条件纳入面向文本分类的对抗文本定义的形式化表述中, 具体作如下叙述. 现引入一个对抗文本的流畅性评估函数 h, 对抗文本的流畅性需满足如下条件:

$$h(\mathbf{x}_i')\geqslant\varphi \tag{3}$$

其中, φ 是对抗文本 $\mathbf{x}_i'$ 所允许的流畅性分数下限.

除了流畅性约束以外, 本工作还考虑了对抗文本被目标模型误分类的置信度这一约束. 当前很少有文本对抗领域的工作关注这一约束, 但基于图像对抗样本领域的文献 [2] 中提及了许多对抗文本能以高置信度被目标模型误分类, 本文也将该因素考虑进面向文本分类的对抗文本定义的形式化描述中, 具体表述如下. 引入一个置信度函数 e, 该函数能够输出对抗文本 $\mathbf{x}_i'$ 被目标模型误分类时的置信度. 该置信度需要满足如下条件:

$$e(\mathbf{x}_i')\geqslant\theta \tag{4}$$

其中, θ 是对抗文本 $\mathbf{x}_i'$ 被目标模型误分类时所允许的置信度下限.

当目标模型的输出为不包含置信度信息的硬标签时, 则无需考虑置信度的约束. 在这种情况下, 结合公式 (2) 和公式 (3), 面向文本分类任务的对抗文本定义的形式化表示如下:

$$f(\mathbf{x}_i) \neq f(\mathbf{x}_i'), \quad \text{s.t.} \begin{cases} g(\mathbf{x}_i, \mathbf{x}_i') \leqslant \varepsilon \\ h(\mathbf{x}_i') \geqslant \varphi \end{cases} \tag{5}$$

当目标模型的输出为包含置信度信息的软标签时, 本文将进一步考虑置信度的约束. 在这种情况下, 结合公式 (2)–公式 (4), 本文面向文本分类任务的对抗文本定义的形式化表述如下:

$$f(\mathbf{x}_i) \neq f(\mathbf{x}_i'), \quad \text{s.t.} \begin{cases} g(\mathbf{x}_i, \mathbf{x}_i') \leqslant \varepsilon \\ h(\mathbf{x}_i') \geqslant \varphi \\ e(\mathbf{x}_i') \geqslant \theta \end{cases} \tag{6}$$

2.2 威胁模型

面向文本分类任务的对抗攻击的威胁模型设计通常关注两点, 分别是攻击场景以及模型输出的组成部分.

常见的攻击场景可分为白盒场景以及黑盒场景. 其中, 白盒场景下的攻击表示敌手知道关于目标模型的全部信息, 包括目标模型的所有训练数据、目标模型的内部结构以及所有参数信息. 而黑盒场景下的攻击则完全相反, 敌手仅允许访问目标模型并得到相应的输出, 对该模型的训练数据、内部结构以及参数等信息一无所知. 相比于白盒场景, 黑盒场景为物理世界中更为普遍的攻击场景. 因此在本工作中无论是针对中文 BERT 模型还是 ChatGPT 模型的攻击, 均为黑盒场景下的攻击.

目标模型的输出通常有两种, 分别为硬标签和软标签. 其中, 硬标签中仅包含目标模型对输入文本的预测标签; 而软标签中不仅包含预测标签, 还包括该预测标签所对应的分类置信度. 本文将中文 BERT 模型作为中间目标模型进行攻击, 随后利用对抗样本的可迁移性来攻击最终目标模型 ChatGPT. 在攻击中文 BERT 模型过程中使用的对抗文本生成方法均为基于词语重要性的攻击方法, 均需要通过目标模型输出的分类置信度来计算词语重要性. 因此对于中文 BERT 这一中间目标模型来说, 本工作选择软标签作为其输出. 而最终目标模型 ChatGPT 作为第三方提供的生成模型而非专业的分类模型, 其输出只能为硬标签.

3 方法论

所提的基于 OAD 的评估方法主要分为两种类型, 分别是基于 OAD 的 LLM 鲁棒性评估方法 ORS 以及基于 OAD 的对抗文本流畅性评估方法 OFS. 图 3 展示了所提方法的总体框架.

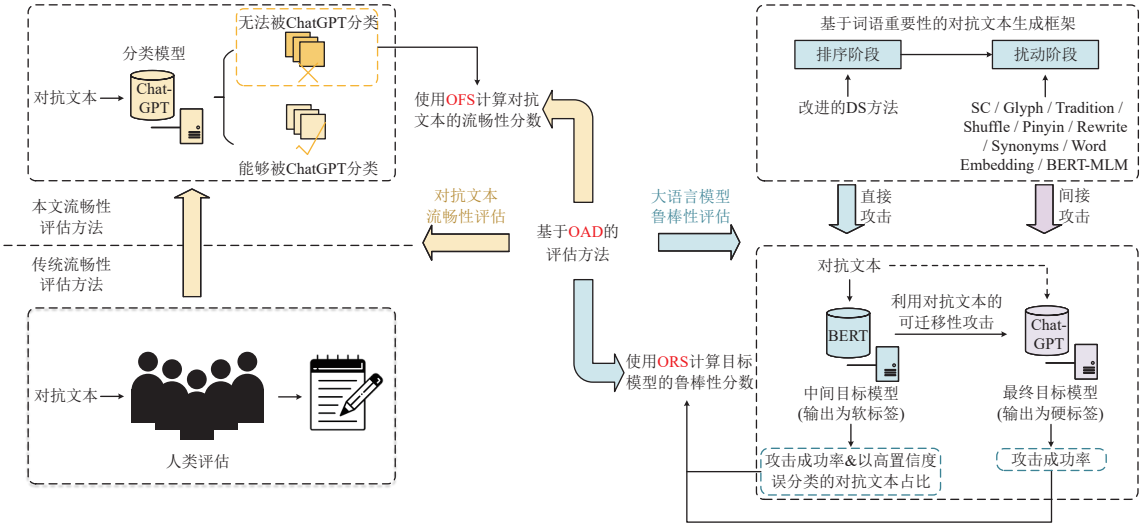


图 3 基于 OAD 的评估方法总体框架

基于 OAD 的 LLM 鲁棒性评估方法包含 3 个主要过程. 首先, 在基于词语重要性的对抗文本生成框架下生成对抗文本, 并使用这些对抗文本直接攻击中文 BERT 模型. 随后, 利用对抗文本的可迁移性, 使用面向中文 BERT

模型生成的对抗文本攻击最终目标模型 ChatGPT. 使用此间接攻击的方式是因为中文 BERT 的输出为软标签, 而 ChatGPT 的输出则为硬标签, 而基于词语重要性的对抗文本生成框架需要用到输出中的置信度信息, 因此需要利用对抗文本的可迁移性实施间接攻击. 此过程中需要注意的是, 由于 ChatGPT 是一个面向多种 NLP 任务的生成模型, 其内部参数很难提取, 因此本文并没有使用多次问询的方式构造其影子模型, 而是在黑盒场景下使用中间目标模型进行间接攻击. 最后, 分别面向硬标签和软标签使用 ORS 计算对应目标模型的鲁棒性分数, 以达到评估其鲁棒性的目的.

基于 OAD 的对抗文本流畅性评估方法主要利用 LLM 强大的自然语言理解能力, 将 LLM 当作一个分类模型对输入的对抗文本进行分类. 本工作使用无法被分类的对抗文本占比, 利用 OFS 为对抗文本的流畅性打分, 整个过程不像传统方法一样需要人类参与, 实现了对抗文本流畅性的自动化评估.

两种评估方法的具体细节详见第 3.1 节和第 3.2 节.

3.1 基于 OAD 的 LLM 鲁棒性评估

3.1.1 基于词语重要性的对抗文本生成方法

本工作采用的文本对抗攻击方法均在基于词语重要性的对抗文本生成框架下设计. 该框架通常分为两个阶段, 分别是排序阶段和扰动阶段. 在排序阶段中需要为文本中每个词语的重要性打分, 并按重要性大小由高到低对这些词语进行排序; 而在扰动阶段中则为所挑选的词语依次添加扰动. 本节将依次讲解排序阶段中使用的打分方法和扰动阶段中采用的扰动方法.

3.1.1.1 重要性词语打分方法

本文使用的重要性词语打分方法为文献 [10] 中提出的改进的 DS 方法. 给定一篇文本 $\mathbf{x}$, 其中包含 n 个词语, 即 $\mathbf{x}=\mathbf{w}_1,\mathbf{w}_2,\ldots,\mathbf{w}_n$. 假设文本 $\mathbf{x}$ 对应的真值标签为 y, 即有 $f(\mathbf{x})=y$, 并且使用 $f_y(\mathbf{x})$ 表示文本 $\mathbf{x}$ 在标签 y 上的置信度. 则 $\mathbf{x}$ 中的任意词语 $\mathbf{w}_i$ $(i=1,\ldots,n)$, 其重要性分数 $I(\mathbf{w}_i)$ 可用如下公式表示:

$$I(\mathbf{w}_i)=\begin{cases} f_y(\mathbf{x})-f_y(\mathbf{x}\backslash\mathbf{w}_i), & \text{if } f(\mathbf{x})=f(\mathbf{x}\backslash\mathbf{w}_i)=y \\ f_y(\mathbf{x})-f_y(\mathbf{x}\backslash\mathbf{w}_i)+[f_{y'}(\mathbf{x}\backslash\mathbf{w}_i)-f_{y'}(\mathbf{x})], & \text{if } f(\mathbf{x})=y\wedge f(\mathbf{x}\backslash\mathbf{w}_i)=y'\wedge y\neq y' \end{cases} \tag{7}$$

其中, $\mathbf{x}\backslash\mathbf{w}_i$ 表示把词语 $\mathbf{w}_i$ 从文本 $\mathbf{x}$ 中删去.

使用公式 (7) 计算文本中每一个词语的重要性分数. 若 $I(\mathbf{w}_i)>0$, 则说明词语 $\mathbf{w}_i$ 对其真值标签 y 有正向影响, 即 $\mathbf{w}_i$ 为 $\mathbf{x}$ 中的重要词语, 反之即非 $\mathbf{x}$ 中的重要词语. 挑选重要性分数大于 0 的所有词语, 并按重要性大小由高到低对它们进行排序, 便于后续为这些词语依次添加扰动.

3.1.1.2 中文扰动方法

本工作主要采用了 9 种中文扰动方法为重要词语添加扰动. 这 9 种扰动方法的具体介绍如下.

(1) Tradition[19]: Tradition 扰动是一种字符级扰动方法, 其将中文文本中的简体字转化为对应的繁体字, 它是一种中文独有的扰动方法. 汉字的书写形式随时代的演变而不断变化, 而繁体字是一种被广泛了解的汉字形式之一. 当前大部分文档均由简体字书写, 使用 Tradition 方法后, 在一定程度上可以达到添加扰动的目的. 这种扰动方法适用于简体形式和繁体形式不同的汉字.

(2) Rewrite[20]: Rewrite 是一种字符级扰动方法, 该方法是一种基于汉语拼音的扰动方法, 其在原词语中随机选择一个汉字, 将该汉字直接替换为它对应的拼音形式. 拼音是一种中文特有的文字拼读规则, 因此 Rewrite 也是一种中文独有的扰动方法. 相比于本节 (6) 中介绍的 Pinyin 方法, Rewrite 方法更加有助于人类对原文含义的理解, 但其很容易通过将拼音还原成文字的防御方法将对抗文本进行自动修正.

(3) Splitting-character (SC)[17]: SC 扰动是一种字符级扰动方法, 其将一个汉字拆分成组成它的偏旁部首, 这是一种中文独有的扰动方法. 中文是一种既能够表音, 又能够表意的语言, 而其表意的部分往往通过汉字的偏旁部首体现. 将一个汉字按偏旁部首拆分后, 人类仍然能够理解其含义. 但为了保证对抗文本的流畅性, 本工作仅对左右结构的汉字进行拆分.

(4) Glyph[17]: Glyph 是一种字符级扰动方法, 其使用一个汉字的形近字替换原汉字, 也是一种中文独有的扰动

方法. 该方法将文字图像化, 并通过 CNN 来找出一个汉字的形近字. 详细过程可参考文献 [17]. 由于该方法需要通过 CNN 来产生结果, 因此相比于其他扰动方法, 其所需时间较长, 且找到的形近字的质量也与 CNN 的训练质量相关.

(5) Shuffle[17]: Shuffle 是一种字符级扰动方法, 其通过将一个中文词语中包含的汉字打乱顺序来添加扰动. 该方法是将英文中的相邻字母替换[14]迁移至中文的一种方法. 该方法利用了人脑对于打乱顺序的文字自动纠错的能力来达到人类难以察觉的目的[28]. 与其他扰动方法相比, 该方法更容易被拼写检查找到并自动纠正.

(6) Pinyin[17]: Pinyin 是一种词语级扰动方法, 该方法将原词语基于其中文拼音扰动为它的谐音词, 其同样是一种中文独有的扰动方法. 谐音词主要包括同音词、前后鼻音不同的谐音词以及平翘舌不同的谐音词. 具体例子详见文献 [17]. 但由于该方法是基于相似发音而设计的扰动方法, 扰动后的词语在含义上可能与原词语完全不一样, 因此使用该方法后在一定程度上会为人类理解原文含义造成障碍.

(7) Synonyms[17]: Synonyms 是一种词语级扰动方法, 其主要基于专家构造的同义词词典来寻找原词的同义词, 并用挑选出的同义词对原词进行替换. 该方法是一种从英文迁移到中文的方法[12]. 然而, 对于部分特定词语来说, 其可能没有同义词. 在这种情况下, 该方法则无法达到扰动原词的目的.

(8) Word Embedding[10,14]: Word Embedding 是一种词语级扰动方法, 该方法在词嵌入空间中寻找离原词语的词向量最接近的前 k 个词向量, 将这些词向量对应的词语作为原词语的候选替换词. 该方法是一种从英文中迁移至中文的方法[10,14]. 然而, 在词嵌入空间中找到的词语很可能是原词语的反义词, 对于情感倾向分类等任务来说, 反义词可能会完全改变原句的语义. 因此, 在面向此类任务生成对抗文本时, 需要先对原词语的反义词进行过滤. 本文使用大连理工大学情感词汇本体词典[29]进行反义词过滤. 由于该方法是静态的词替换操作, 因此还需要将与原词语词性不一致的候选词进行过滤.

(9) BERT-MLM[21]: BERT-MLM 是一种词语级扰动方法, 该方法利用 BERT 模型中的 MLM 机制基于文本上下文来生成原词语的候选替换词. 该方法也是一种从英文中迁移至中文的方法[11,15,16], 但本文使用的方法依照文献 [21] 所述, 根据中文的语言特点进行了改进. 与文献 [21] 中的方法不同的是, 本文为了提升对抗文本的流畅性, 限制原词语的词长为 1 时, 其替换词的词长也需为 1; 而原词语词长不小于 2 时, 其替换词的词长为 2. 与 Word Embedding 方法类似, BERT-MLM 方法在面向情感倾向分类任务时也需过滤反义词. 本文使用的过滤方法与 (8) 中介绍的过滤方法一致.

对于词语级扰动方法, 往往会有多个候选替换词构成候选词集, 此时需要从候选词集中选择一个最合适的词语来替换原词语. 本文选择替换前后能够使目标模型置信度变化最大的词语替换原词. 给定一篇包含 n 个词语的文本 $\mathbf{x}=\mathbf{w}_1,\mathbf{w}_2,\ldots,\mathbf{w}_n$ 以及 $\mathbf{x}$ 中的某个词语 $\mathbf{w}_i$ $(i=1,\ldots,n)$, 假设 $\mathbf{w}_i$ 的候选词集 $\mathbb{C}=\{\mathbf{c}_1,\mathbf{c}_2,\ldots,\mathbf{c}_m\}$. 对于候选词集 $\mathbb{C}$ 中的任意候选词 $\mathbf{c}_j$ $(j=1,\ldots,m)$, 其置信度变化 $C(\mathbf{c}_j)$ 可用如下公式计算:

$$C(\mathbf{c}_j)=\begin{cases} f_y(\mathbf{x})-f_y(\bar{\mathbf{x}}), & \text{if } f(\mathbf{x})=f(\bar{\mathbf{x}})=y \\ f_y(\mathbf{x})-f_y(\bar{\mathbf{x}})+[f_{y'}(\bar{\mathbf{x}})-f_{y'}(\mathbf{x})], & \text{if } f(\mathbf{x})=y\wedge f(\bar{\mathbf{x}})=y'\wedge y\neq y' \end{cases} \tag{8}$$

其中, $\bar{\mathbf{x}}$ 表示将词语 $\mathbf{w}_i$ 替换为候选词 $\mathbf{c}_j$ 后的文本. 则最终选择的替换词 $\mathbf{w}_i$ 可由公式 (9) 表示:

$$\mathbf{w}_i'=\arg\max_{\mathbf{c}_j\in\mathbb{C}} C(\mathbf{c}_j) \tag{9}$$

3.1.2 基于可迁移性对 ChatGPT 的攻击

由于 ChatGPT 并非专业的分类模型, 其只能输出不包含置信度信息的硬标签, 因此本文基于对抗文本的可迁移性对其进行间接攻击. 本文将输出为软标签的中文 BERT 作为中间目标模型进行攻击, 利用其输出的置信度信息使用基于词语重要性的对抗文本生成方法来构造对抗文本. 随后使用面向中文 BERT 生成的对抗文本攻击最终目标模型 ChatGPT. 面向文本分类任务, 将 gpt-3.5-turbo-0613 版本的 ChatGPT 模型作为目标模型, 调用其 API 对其进行攻击. 面向不同的分类任务, 本文使用的提示语各不相同, 具体如下所述.

面向新闻分类任务时, 本文使用的提示语如下: “请告诉我下面的新闻属于科技新闻、教育新闻、财经新闻、社会新闻、运动新闻中的哪个类别. 回答时请用 0 表示科技新闻, 用 1 表示教育新闻, 用 2 表示财经新闻, 用 3 表

示社会新闻, 用 4 表示运动新闻, 用 5 表示无法分类. 回答时请直接给出类别对应的数字, 不必说明原因.”

面向情感倾向分类任务时, 本文使用的提示语如下: “请帮我确定以下文本的情感极性. 用 0 表示消极情感, 1 表示积极情感, 2 表示无法判断 (只给出结果而不作解释, 且对整体文本只给出一个标签即可, 无需给出多个标签).”

在 ChatGPT 未将对抗文本划分为无法分类的条件下, 若 ChatGPT 将对抗文本分类为与其真值标签不一致的标签, 则说明 ChatGPT 被对抗文本欺骗, 视为攻击成功.

3.1.3 目标模型鲁棒性打分方法

为了实现对目标模型鲁棒性的可量化评估, 本文在第 3.1.3.1 节中引入了一个新的概念 OAD, 并提出了一种基于 OAD 的目标模型鲁棒性打分方法 ORS. 该方法不但能够评估一个目标模型面对不同对抗攻击时的鲁棒性, 还能够在同一基准上对比多个目标模型在多种对抗攻击条件下的鲁棒性, 并最大限度地拉开面对不同对抗攻击时, 目标模型鲁棒性分数的差距, 方便进行更加直观的比较. 本文面向输出为硬标签的目标模型以及输出为软标签的目标模型分别基于 OAD 设计了对应的鲁棒性打分方法, 在第 3.1.3.1 节以及第 3.1.3.2 节中将分别对这两种方法进行详细介绍.

3.1.3.1 面向硬标签的打分方法

为了对目标模型的鲁棒性设计合理的量化评估指标, 本文引入了一个新的概念 OAD. OAD 量化了一组离散的实数与某个给定的正常数的平均偏差, 其形式化表示如下. 给定一组包含 l 个实数的离散的数值 $x_1, x_2, \ldots, x_l$ 和一个常数 c ($c>0$), 这组数据与该常数 c 的 OAD 值 λ 可使用如下公式计算:

$$\lambda = \frac{1}{l}\sum_{i=1}^{l}(c - x_i) = c - \frac{1}{l}\sum_{i=1}^{l}x_i \tag{10}$$

现将 OAD 用于计算输出为硬标签的目标模型的鲁棒性分数, 本文使用攻击成功率对其进行实例化. 现有 n 个目标模型, 对每个目标模型均有 m 种攻击方式, 对每种攻击方式都有 p 种扰动率, 对任意一个扰动率用 α_k ($k=1,\ldots,p$) 表示. 对于第 i ($i=1,\ldots,n$) 个目标模型来说, 其面对第 j ($j=1,\ldots,m$) 种攻击时, 在不同扰动率下的攻击成功率分别可以用 $U_{ij}(\alpha_1), U_{ij}(\alpha_2), \ldots, U_{ij}(\alpha_p)$ 表示, 这组数值即为公式 (10) 中多个离散实数的实例化, 而公式 (10) 中的正常数 c 在本工作中将其值设置为 1. 在任意扰动率 α_k 下的攻击成功率 $U_{ij}(\alpha_k)$, 可以用如下公式计算:

$$U_{ij}(\alpha_k) = \frac{u_{ij}(\alpha_k)}{T} \tag{11}$$

其中, $u_{ij}(\alpha_k)$ 表示当扰动率为 α_k 时, 成功攻击目标模型的对抗文本的数量. 而 T 则表示真值标签与预测标签一致的原始良性文本数量.

为了使不同攻击方法之间鲁棒性分数的差异更加明显, 本文向公式 (10) 中引入一个放大系数 β. 为了便于计算最终的鲁棒性得分, 本工作中将 β 设置为正整数. 与此同时, 为了将参数 β 值的范围控制在 $(0,100]$, 本文引入一个超参数 q, 并将 q 值设定为 100. 引入上述参数后, 在本工作中, 对于第 i 个目标模型在第 j 种攻击下的 OAD 的值 λ_{ij} 可以用如下公式表示:

$$\lambda_{ij} = \frac{q}{p}\sum_{k=1}^{p}[c - U_{ij}(\alpha_k)\cdot\beta] = qc - \frac{q\beta}{p}\sum_{k=1}^{p}U_{ij}(\alpha_k) \tag{12}$$

为了确保最终的鲁棒性分数的范围为 $(0,10]$, 引入一个控制分制的超参数 η. 结合上文中常数 c 以及超参数 q 的值, 本文将 η 的取值设置为 10. 即第 i 个目标模型在第 j 种攻击下的最终的鲁棒性得分 rs_{ij} 可用如下公式计算:

$$rs_{ij} = \frac{\lambda_{ij}}{\eta} = \frac{qc}{\eta} - \frac{q\beta}{p\eta}\sum_{k=1}^{p}U_{ij}(\alpha_k) \tag{13}$$

此时需要给参数 β 找到一个合适的值, 在鲁棒性分数范围在 $(0,10]$ 的情况下, 将每种目标模型面对各个攻击方法时的鲁棒性得分尽量拉开差距, 从而使得分数的对比更加直观. 为了找到该值, 本文引入一个决策函数 $H(z)$, 其自变量 z 为 β 可能的取值. 可使用 λ_{ij} 将 $H(z)$ 进行如下表示:

$$H(z)=qc-\max_{i\in\{1,\ldots,n\}\wedge j\in\{1,\ldots,m\}}\frac{qz}{p}\sum_{k=1}^{p}U_{ij}(\alpha_k)=\min_{i\in\{1,\ldots,n\}\wedge j\in\{1,\ldots,m\}}\lambda_{ij}(z) \tag{14}$$

参数β的值使用决策函数$H(z)$可以进行如下表示:

$$\beta=\begin{cases}100, & \text{if } H(100)>0\\ z, & \text{if } H(z)>0\wedge H(z+1)\leqslant 0\wedge z\in\{1,2,\ldots,99\}\end{cases} \tag{15}$$

结合公式 (14) 和公式 (15), 即可得到适用于当前所有模型在各种攻击下的参数β. 将得到的β值代入公式 (13) 中, 即可计算出每个目标模型在每种攻击下的鲁棒性得分.

3.1.3.2 面向软标签的打分方法

相比于输出为硬标签的目标模型, 输出为软标签的目标模型除了可以使用攻击成功率作为模型鲁棒性的评估依据外, 还可以使用高置信度的对抗文本占比来衡量目标模型的鲁棒性. 因为高置信度的对抗文本占比能够反映目标模型被对抗样本迷惑的程度, 该比例越高, 则目标模型被迷惑的程度越深. 本文将置信度超过 0.8 的对抗文本视为高置信度的对抗文本. 对于输出为软标签的目标模型, 本工作使用攻击成功率和高置信度对抗文本占比结合的方式, 将 OAD 进行实例化. 其中, 使用攻击成功率对 OAD 的实例化过程与第 3.1.3.1 节一致, 下面简单介绍使用高置信度对抗文本占比对 OAD 的实例化过程.

对于第i ($i=1,\ldots,n$) 个目标模型来说, 其面对第j ($j=1,\ldots,m$) 种攻击时, 在不同扰动率下高置信度对抗文本占比可以用$V_{ij}(\alpha_1),V_{ij}(\alpha_2),\ldots,V_{ij}(\alpha_p)$表示. 在任意扰动率$\alpha_k$下的高置信度对抗文本占比可用$V_{ij}(\alpha_k)$表示, 其中$V_{ij}(\alpha_k)$的值可以使用如下公式计算:

$$V_{ij}(\alpha_k)=\frac{v_{ij}(\alpha_k)}{T'} \tag{16}$$

其中, $v_{ij}(\alpha_k)$表示当扰动率为α_k时, 以高置信度误分类的对抗文本数. 而T'则表示面向第i个目标模型使用第j种攻击方法成功生成的对抗文本总数.

使用$V_{ij}(\alpha_k)$替换公式 (12)–公式 (14) 中的$U_{ij}(\alpha_k)$, 所有超参数的值均与第 3.1.3.1 节中对应的值一致. 再将使用$V_{ij}(\alpha_k)$替换后的公式 (14) 与公式 (15) 结合, 即可求出适用于高置信度对抗文本占比实例化的β_v. 将适用于攻击成功率实例化的扩大系数设为β_u. 为了使基于攻击成功率计算的模型鲁棒性分数和基于高置信度文本占比计算的模型鲁棒性分数能够在同一基准上计算, 因此需要对最终的β值进行统一. 结合公式 (15) 可知, 为了避免决策函数的取值为负数, 需要在β_u和β_v中取一个较小的值. 则最终的β取值如下:

$$\beta=\min\{\beta_u,\beta_v\} \tag{17}$$

将最终得到的β值分别代入公式 (13) 以及使用$V_{ij}(\alpha_k)$替换后的公式 (13) 中, 可以分别得到基于攻击成功率计算的模型鲁棒性分数rsu_{ij}以及基于高置信度对抗文本占比计算的模型鲁棒性分数rsv_{ij}. 则第i个目标模型在第j种攻击下的最终鲁棒性得分rs_{ij}可以使用如下公式计算:

$$rs_{ij}=(rsu_{ij}+rsv_{ij})/2 \tag{18}$$

在上述过程中需要注意的是, 若在某种扰动率α_k下, 使用第j种方法攻击第i个模型时, 未成功生成任何对抗文本, 此时公式 (16) 中的T'为 0, 则$V_{ij}(\alpha_k)$无法计算. 在这种情况下, 计算参数β以及rsv_{ij}时, 仅需将扰动率种类的总数p替换为$V_{ij}(\alpha_k)$可以计算出实际值时对应的扰动率的数目. 若使用第j种方法攻击第i个目标模型时, 在所有扰动率下的$V_{ij}(\alpha_k)$均无法计算, 则在计算β时跳过面向第i个目标模型的第j种方法, 并将rsv_{ij}置为 NaN. 在这种情况下, 最终的鲁棒性得分rs_{ij}仅由rsu_{ij}决定, 则此时第i个目标模型在第j种攻击方法下的最终鲁棒性得分rs_{ij}可表示如下:

$$rs_{ij}=rsu_{ij} \tag{19}$$

3.2 基于 OAD 的对抗文本流畅性评估

本文将所提的 OAD 的应用扩展到对抗文本的流畅性评估中, 提出一种基于 OAD 的对抗文本流畅性打分方法 OFS, 实现此过程的自动化. 传统的对抗文本流畅性评估方式往往需要人类参与[8−12,14−17,21], 这在一定程度上消

耗了较多的人力和物力. 本工作利用了 ChatGPT 强大的自然语言理解能力, 将其作为分类模型对输入的对抗文本进行分类. 在本工作中, 将无法被 ChatGPT 分类的对抗文本视作不流畅的对抗文本, 而可以被 ChatGPT 分类为有效类别的对抗文本则视作流畅的对抗文本. 在这种情况下, 仅将 ChatGPT 当作一个分类模型而非目标模型, ChatGPT 仅起到评估对抗文本流畅性的作用. 分类提示语同第 3.1.2 节中提到的一致.

依照上述思路, 结合基于 OAD 的评估方法, 可以使用无法被 ChatGPT 分类的对抗文本占比对 OAD 进行实例化, 以实现自动化的流畅性评估. 下面简单介绍这一实例化过程.

针对第 i ($i=1,\ldots,n$) 个 DL 模型, 使用第 j ($j=1,\ldots,m$) 种对抗文本生成方法生成的对抗文本, 在不同扰动率下无法被 ChatGPT 分类的对抗文本所占的比例可以用 $W_{ij}(\alpha_1),W_{ij}(\alpha_2),\ldots,W_{ij}(\alpha_p)$ 表示. 在任意扰动率 α_k 下无法被分类的对抗文本占比可用 $W_{ij}(\alpha_k)$ 表示, 其中 $W_{ij}(\alpha_k)$ 的值可以使用如下公式计算:

$$W_{ij}(\alpha_k)=\frac{w_{ij}(\alpha_k)}{T''} \tag{20}$$

其中, $w_{ij}(\alpha_k)$ 表示当扰动率为 α_k 时, 无法被 ChatGPT 分类的对抗文本数. T'' 则表示真值标签与 ChatGPT 分类的预测标签一致的原始良性文本数量.

使用 $W_{ij}(\alpha_k)$ 替换公式 (12)–公式 (14) 中的 $U_{ij}(\alpha_k)$, 所有超参数的值均与第 3.1.3.1 节中对应的值一致. 再将使用 $W_{ij}(\alpha_k)$ 替换后的公式 (14) 与公式 (15) 结合, 即可求出适用于对抗文本流畅性自动化评估的 β. 将得到的 β 值代入使用 $W_{ij}(\alpha_k)$ 替换后的公式 (13) 中, 即可得到针对第 i 个模型, 使用第 j 种攻击方法所生成的对抗文本的最终流畅性得分 fs_{ij}.

通过上述内容可知, 使用本文提出的 OFS 能够实现自动化的对抗文本流畅性评估, 同时能够得到量化的流畅性分数. 该方法不但能够更加直观地评估各攻击方法生成的对抗文本的流畅性, 还能够大大减少评估过程中耗费的人力物力, 降低评估成本.

4 实验分析

4.1 实验设置

4.1.1 数据集

本工作面向中文文本分类任务设计对抗文本, 使用了两个真实世界中的数据集, 分别是中文新闻分类数据集 THUCNews[30]以及 ChineseNlpCorpus[31]中包含的中文情感倾向分类数据集. 对于新闻分类数据集, 本文选取其中科技、教育、财经、社会及体育这 5 个类别下的新闻, 每个类别各随机选择 35 000 条数据. 截取每个新闻的标题, 按训练集、验证集、测试集 5:1:1 的比例构造训练数据. 对于情感倾向分类数据集, 本文选取其中对图书、平板、手机、水果、洗发水、热水器、蒙牛牛奶、服装、电脑、宾馆这 10 种物品的评论, 共包含 31 728 条正面评论以及 31 046 条负面评论. 将每个类别的物品的正面评论和负面评论分别按 8:1:1 的比例进行划分, 再将所有类别的物品评论分别整合为训练集、验证集、测试集并打乱顺序, 构造最终训练数据.

4.1.2 目标模型及训练细节

本文将中文 BERT 模型以及 ChatGPT 模型作为目标模型进行攻击. 对于中文 BERT 模型, 本工作面向文本分类任务, 使用第 4.1.1 节中提到的两种数据集分别对预训练模型 BERT-base-chinese 进行微调. 将 hidden size 设置为 768. 对于新闻分类任务, 训练时的 padding size、batch size 以及 epochs 分别设置为 32、64 和 3; 对于情感倾向分类任务, 上述 3 个参数分别设置为 256、16 和 3. 对于这两种分类任务, 训练时采用的优化算法均是学习率为 5×10^{-5} 的 Adam[32]算法, 使用的 GPU 均为 NVIDIA GeForce RTX 3080. 对于 ChatGPT 模型的介绍见第 3.1.2 节, 在此不再赘述.

4.1.3 攻击方法

本文使用的攻击方法共 9 种, 均在基于词语重要性的对抗文本生成框架下设计, 下面按该框架的排序阶段和扰动阶段分别介绍这两个阶段中采用的具体方法.

4.1.3.1 排序阶段

为了控制变量, 本文在排序阶段中采用的方法均选择文献 [10] 中提出的改进的 DS 方法. 该方法的具体介绍详见第 3.1.1.1 节, 在此不再赘述.

4.1.3.2 扰动阶段

本文分别采用第 3.1.1.2 节中提到的 9 种中文扰动方法对原始文本中的重要词语进行扰动, 关于各扰动方法的实现细节及适用条件详见第 3.1.1.2 节. 将改进的 DS 方法与这 9 种方法结合, 即可组合为 9 种不同的对抗文本生成方法. 这些方法的具体介绍如下.

(1) CWordAttacker_Tradition (C_T)[19]: 将改进的 DS 方法与 Tradition 扰动结合;

(2) GreedyAttack_Rewrite (G_R)[20]: 将改进的 DS 方法与 Rewrite 扰动结合;

(3) Argot_SC (A_SC)[17]: 将改进的 DS 方法与 SC 扰动结合;

(4) Argot_Glyph (A_G)[17]: 将改进的 DS 方法与 Glyph 扰动结合;

(5) Argot_Shuffle (A_Sh)[17]: 将改进的 DS 方法与 Shuffle 扰动结合;

(6) Argot_Pinyin (A_P)[17]: 将改进的 DS 方法与 Pinyin 扰动结合;

(7) Argot_Synonyms (A_Sy)[17]: 将改进的 DS 方法与 Synonyms 扰动结合;

(8) TextFooler (TF)[10]: 将改进的 DS 方法与 Word Embedding 扰动结合;

(9) Chinese BERT Tricker (CBT)[21]: 将改进的 DS 方法与 BERT-MLM 扰动结合.

4.1.4 文本相似度评估指标

本文采用余弦相似度、词移距离[33]、编辑距离以及杰卡德系数对对抗文本和原始文本的相似度进行全面的评估. 下面对这 4 种方法进行简要介绍.

(1) 余弦相似度: 余弦相似度是一种衡量文本相似度的常见方式. 向量空间中的两个词向量或者句向量的余弦相似度越接近 1, 则说明这两个词语或句子越相似. 给定一个词向量 $\mathbf{a}=(a_1,a_2,\ldots,a_n)$ 和一个词向量 $\mathbf{b}=(b_1,b_2,\ldots,b_n)$, 则 $\mathbf{a}$ 和 $\mathbf{b}$ 之间的余弦相似度 $\cos(\mathbf{a},\mathbf{b})$ 可用如下公式计算:

$$\cos(\mathbf{a},\mathbf{b})=\frac{\sum_{i=1}^{n}a_i\times b_i}{\sqrt{\sum_{i=1}^{n}(a_i)^2}\times\sqrt{\sum_{i=1}^{n}(b_i)^2}} \tag{21}$$

(2) 词移距离[33]: 词移距离利用 Word2Vec[34]词向量模型对词语进行向量化表示, 并通过计算两个词向量之间的欧氏距离计算它们的相似度. 该方法的具体细节可参考文献 [33]. 词移距离越短, 则文本相似度越高.

(3) 编辑距离: 编辑距离是指一个字符串变为另一个字符串的过程中最小编辑次数. 本文使用最具代表性的莱文斯坦距离来计算两文本之间的编辑距离. 莱文斯坦距离允许的操作包括字符级的增加、删除和替换, 且每次仅能操作一个字符. 编辑距离越短, 则文本相似度越高.

(4) 杰卡德系数: 杰卡德系数通过两个文本中词语集合的交集与并集之比来衡量两篇文本之间的相似度. 杰卡德系数越接近 1, 则文本相似度越高. 给定两文本, 它们中包含的词语集合分别用词集 $\mathbb{A}$ 和词集 $\mathbb{B}$ 表示, 则杰卡德系数 $J(\mathbb{A},\mathbb{B})$ 可用如下公式计算:

$$J(\mathbb{A},\mathbb{B})=\frac{|\mathbb{A}\cap\mathbb{B}|}{|\mathbb{A}\cup\mathbb{B}|}=\frac{|\mathbb{A}\cap\mathbb{B}|}{|\mathbb{A}|+|\mathbb{B}|-|\mathbb{A}\cap\mathbb{B}|} \tag{22}$$

4.2 实验结果

本文分别从新闻分类和情感倾向分类的测试集中各随机抽取 1 000 条数据进行实验. 本节及第 4.3 节的图表中提到的扰动比例为添加扰动的词语数量与原始文本中词语总数的比值. 实验结果如下.

4.2.1 目标模型鲁棒性评估结果

4.2.1.1 面向硬标签的鲁棒性评估结果

本节分别以 ChatGPT 和中文 BERT 为例, 在新闻分类和情感倾向分类这两个数据集上, 使用第 3.1.3.1 节中所

提的适用于输出为硬标签的目标模型的评估方法对其鲁棒性进行评估. 虽然 ChatGPT 的输出的为硬标签而中文 BERT 的输出为软标签, 但为了将两者在同一基准上比较, 在本节中对于这两个目标模型都仅关注各攻击方法在不同扰动比率下对它们的攻击成功率. 而在第 4.2.1.2 节中再进一步展示加入置信度信息后, 中文 BERT 模型的鲁棒性评估结果.

面向新闻分类时, 在目标模型为 ChatGPT 以及中文 BERT 的情况下, 每种攻击方法的攻击成功率、对应的鲁棒性分数 rs 以及扩大系数 β 值分别如表 1 和表 2 所示, 其中表格的第 1 列表示各种对抗文本生成方法. 从表 1 和表 2 中可以看出, 对于绝大多数方法, 攻击成功率均随扰动比例的增加而上升. 将表 1 和表 2 中结合来看, 可以看出无论是 ChatGPT 还是中文 BERT, 在面对 C_T 的攻击时, 它们的鲁棒性分数均是所有攻击方法中最高的; 而在面对 CBT 的攻击时, 它们的鲁棒性分数则均是所有攻击方法中最低的. 这说明无论是 ChatGPT 还是中文 BERT 均能在很大程度上抵御 C_T 的攻击, 但却较难抵御 CBT 的攻击. 与此同时, 能够发现这两个目标模型面对词语级对抗攻击时的鲁棒性分数明显比面对字符级对抗攻击时的鲁棒性分数低. 这说明相比于字符级对抗文本生成方法, 这两个目标模型更难抵御词语级对抗文本生成方法的攻击. 而将表 1 和表 2 进行对比可知, ChatGPT 面对各种攻击方法时的平均鲁棒性分数比中文 BERT 高 20% 左右, 这说明相比于中文 BERT, ChatGPT 在对抗攻击下具有更强的鲁棒性.

表 1　面向新闻分类时各方法对 ChatGPT 的攻击成功率 (%) 及鲁棒性评估

方法	不同扰动比例 α 下的攻击成功率 (%)									rs
	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.50	
C_T	1.24	2.23	2.11	1.86	1.98	2.23	2.23	2.35	2.23	9.59
G_R	3.22	3.59	4.58	4.83	4.71	4.21	4.46	4.83	3.72	9.15
A_SC	3.22	4.09	3.84	3.84	3.10	2.60	2.48	2.23	2.11	9.39
A_G	4.34	5.08	4.71	4.21	3.84	3.35	2.97	2.23	1.61	9.28
A_Sh	4.46	4.71	5.33	4.83	5.70	5.58	5.58	4.96	4.71	8.98
A_P	3.10	4.46	4.46	4.21	4.46	4.21	4.46	4.58	4.46	9.15
A_Sy	3.59	4.46	4.83	5.58	5.58	5.95	6.32	6.44	6.32	8.91
TF	5.33	6.69	7.81	8.55	8.92	9.05	8.80	9.05	9.42	8.36
CBT	7.93	11.28	13.88	14.37	15.12	16.85	16.98	17.22	16.60	7.11
平均鲁棒性分数 $\overline{rs}$					—					8.88
扩大系数 β					—					2

表 2　面向新闻分类时各方法对中文 BERT 的攻击成功率 (%) 及鲁棒性评估 (硬标签)

方法	不同扰动比例 α 下的攻击成功率 (%)									rs
	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.50	
C_T	0.10	0.10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	10.00
G_R	3.84	5.60	7.16	8.92	11.20	13.49	15.77	16.80	18.57	7.75
A_SC	4.15	7.26	9.96	11.72	13.49	14.42	15.46	15.87	16.70	7.58
A_G	4.56	7.26	11.31	12.86	15.98	18.36	21.06	22.82	24.27	6.92
A_Sh	0.21	0.52	0.73	0.62	0.52	0.41	0.52	0.62	0.62	9.89
A_P	1.35	1.45	2.18	2.28	2.70	2.70	2.59	2.59	2.59	9.55
A_Sy	4.46	7.47	10.17	12.66	14.73	17.53	18.57	19.5	20.64	7.21
TF	9.02	14.21	19.50	22.82	24.90	28.42	30.39	32.37	36.72	5.15
CBT	18.67	26.35	34.02	37.66	41.18	43.78	46.27	48.34	51.35	2.28
平均鲁棒性分数 $\overline{rs}$					—					7.37
扩大系数 β —										2

面向情感倾向分类时, 在目标模型为 ChatGPT 以及中文 BERT 的情况下, 每种攻击方法的攻击成功率、对应的鲁棒性分数以及 β 值分别如表 3 和表 4 所示. 表 3 和表 4 中的数据规律与表 1 和表 2 中类似, 因此可以得到与

之相同的结论. 将表 1、表 2 与表 3、表 4 对比可知, 前者的β大于后者的β值. 这说明面向新闻分类任务时两模型的鲁棒性优于面向情感倾向分类任务时的鲁棒性.

表 3　面向情感分类时各方法对 ChatGPT 的攻击成功率 (%) 及鲁棒性评估

方法	不同扰动比例 α 下攻击成功率 (%)									rs
	0.01	0.03	0.05	0.10	0.15	0.20	0.25	0.30	0.40	
C_T	0.35	0.35	0.46	0.23	0.69	0.69	0.81	0.46	0.69	9.95
G_R	1.62	1.96	2.89	3.23	3.23	4.16	3.35	3.58	3.93	9.69
A_SC	2.66	4.16	3.70	4.97	5.89	5.66	5.31	4.97	5.20	9.53
A_G	2.89	3.46	4.62	8.31	9.70	9.70	10.51	8.66	8.31	9.26
A_Sh	1.73	1.50	2.54	2.89	3.81	4.16	3.23	3.81	4.16	9.69
A_P	1.62	1.73	2.08	3.46	3.35	3.70	3.70	4.04	3.35	9.70
A_Sy	2.19	3.23	3.12	4.62	6.24	7.39	9.12	10.28	11.20	9.36
TF	6.24	7.62	8.78	15.59	20.44	24.71	27.02	27.48	28.41	8.15
CBT	11.32	13.86	14.9	25.64	32.22	35.57	39.38	42.03	42.38	7.14
平均鲁棒性分数 $\overline{rs}$					—					9.16
扩大系数 β					—					1

表 4　面向情感分类时各方法对中文 BERT 的攻击成功率 (%) 及鲁棒性评估 (硬标签)

方法	不同扰动比例 α 下的攻击成功率 (%)									rs
	0.01	0.03	0.05	0.10	0.15	0.20	0.25	0.30	0.40	
C_T	0.11	0.11	0.11	0.00	0.11	0.11	0.11	0.11	0.11	9.99
G_R	3.90	4.21	5.37	7.80	9.27	10.33	10.96	11.49	11.80	9.17
A_SC	4.11	4.64	5.48	8.54	11.38	12.12	12.64	13.80	14.44	9.03
A_G	9.06	11.28	13.17	22.55	28.03	33.09	36.78	39.09	43.73	7.37
A_Sh	1.16	1.37	1.37	1.48	2.00	2.21	2.11	2.00	2.00	9.83
A_P	3.37	3.69	4.21	5.37	6.11	6.22	6.43	6.53	6.53	9.46
A_Sy	10.96	12.75	16.65	24.76	32.35	37.72	42.89	47.21	50.58	6.93
TF	16.54	20.13	25.08	37.93	48.05	54.48	60.17	64.81	68.60	5.60
CBT	29.19	33.83	41.10	56.16	65.44	69.97	74.29	77.03	80.93	4.13
平均鲁棒性分数 $\overline{rs}$					—					7.95
扩大系数 β					—					1

4.2.1.2　面向软标签的鲁棒性评估结果

本节以中文 BERT 为例, 将置信度信息作为衡量输出为软标签的目标模型鲁棒性的指标之一, 分别在新闻分类和情感倾向分类数据集上计算中文 BERT 模型在各种对抗攻击下的鲁棒性分数.

根据第 3.1.3.2 节中所述, 面向软标签的鲁棒性分数由两部分构成, 分别是基于置信度的鲁棒性分数以及基于攻击成功率的鲁棒性分数. 表 5 展示了各对抗文本生成方法攻击面向新闻分类任务的中文 BERT 模型时, 生成的高置信度对抗文本占比、对应的基于置信度的鲁棒性分数 rsv 以及基于置信度的扩大系数β_v值, 其中 NaN 表示该方法在对应扰动率下没有生成对抗文本. 从表 5 中能够得到与表 1 和表 2 类似的结论.

由于表 2 得到的面向新闻分类任务的基于攻击成功率计算出的β_u (即表 2 中的β) 大于表 5 中的β_v, 因此最终的β由后者决定. 将β代入公式 (13) 中, 即可得到如表 6 所示的基于成功率的鲁棒性分数 rsu, 进而计算出最终的联合鲁棒性分数 rs.

表 7 展示了各对抗文本生成方法攻击面向情感倾向分类任务的中文 BERT 模型时, 生成的高置信度对抗文本占比、对应的基于置信度的鲁棒性分数 rsv 以及基于置信度的扩大系数β_v值. 将表 4 和表 7 结合可以看出, 面向情感倾向分类任务时, β_u与β_v均为 1, 因此这时β的最终取值也为 1. 表 4 中展示的鲁棒性得分即为中文 BERT 基于攻击成功率计算出的鲁棒性得分 rsu. 将 rsu 和 rsv 联合计算, 即可计算出如表 8 所示的中文 BERT 面向情感倾向分类任务的最终鲁棒性分数 rs.

表 5 面向新闻分类时各方法在中文 BERT 上的高置信度对抗文本占比 (%) 及基于置信度的鲁棒性评估

方法	不同扰动比例 α 下高置信度对抗文本占比 (%)									*rsv*
	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.50	
C_T	0.00	0.00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	10.00
G_R	45.95	64.81	71.01	75.58	65.74	69.23	70.39	69.14	67.04	3.35
A_SC	50.00	64.29	59.38	65.49	72.31	71.94	73.15	71.90	70.81	3.34
A_G	43.18	41.43	44.95	50.00	53.25	51.41	49.26	45.91	48.72	5.24
A_Sh	33.33	20.00	14.29	16.67	20.00	25.00	20.00	33.33	33.33	7.60
A_P	38.46	35.71	28.57	31.82	23.08	23.08	24.00	24.00	24.00	7.19
A_Sy	58.14	61.11	66.33	65.57	66.20	66.86	69.83	72.34	71.86	3.35
TF	65.52	65.69	68.62	71.82	72.08	71.53	72.70	73.08	70.62	2.98
CBT	75.56	74.41	75.30	76.03	75.82	78.44	80.27	77.90	77.37	2.32
基于置信度的平均鲁棒性分数 $\overline{rsv}$					—					5.04
基于置信度的扩大系数 β_v					—					1

表 6 面向新闻分类的中文 BERT 鲁棒性评估 (软标签)

方法	基于攻击成功率的鲁棒性分数 *rsu*	基于置信度的鲁棒性分数 *rsv*	联合鲁棒性分数 *rs*
C_T	10.00	10.00	10.00
G_R	8.87	3.35	6.11
A_SC	8.79	3.34	6.06
A_G	8.46	5.24	6.85
A_Sh	9.95	7.60	8.77
A_P	9.77	7.19	8.48
A_Sy	8.60	3.35	5.98
TF	7.57	2.98	5.28
CBT	6.14	2.32	4.23
平均值	8.68	5.04	6.86
扩大系数 β		1	

表 7 面向情感分类时各方法在中文 BERT 上的高置信度对抗文本占比 (%) 及基于置信度的鲁棒性评估

方法	不同扰动比例 α 下高置信度对抗文本占比 (%)									*rsv*
	0.01	0.03	0.05	0.10	0.15	0.20	0.25	0.30	0.40	
C_T	0.00	0.00	0.00	NaN	0.00	0.00	0.00	0.00	0.00	10.00
G_R	45.95	42.50	43.14	41.89	42.05	39.80	43.27	43.12	43.75	5.72
A_SC	35.90	34.09	38.46	45.68	45.37	45.22	45.00	41.98	42.34	5.84
A_G	44.19	43.93	48.00	51.87	51.50	52.55	53.58	54.99	54.46	4.94
A_Sh	9.09	15.38	15.38	14.29	15.79	14.29	15.00	15.79	15.79	8.55
A_P	25.00	31.43	27.50	35.29	43.10	42.37	39.34	40.32	40.32	6.39
A_Sy	54.81	55.37	56.33	61.28	63.52	69.27	67.81	70.31	73.75	3.64
TF	43.95	49.21	50.00	61.11	65.57	72.15	75.13	75.28	79.88	3.64
CBT	59.21	62.62	67.18	72.80	79.55	81.63	83.55	84.27	86.98	2.47
基于置信度的平均鲁棒性分数 $\overline{rsv}$					—					5.69
基于置信度的扩大系数 β_v					—					1

结合第 4.2.1.1 节和第 4.2.1.2 节的所有实验可知, 无论是 ChatGPT 模型还是中文 BERT 模型, 相比于字符级的对抗攻击, 它们更难抵御词语级的对抗攻击. 其中, 这两个模型对字符级的 C_T 攻击的抵御效果最好, 而面对词语级的 CBT 攻击时鲁棒性最弱. 与此同时, 在同一评价体系内, ChatGPT 的平均鲁棒性得分比中文 BERT 高 15%–20% 左右. 但 ChatGPT 也在文本对抗攻击下展现出脆弱性, 在情感倾向性分类数据集上, CBT 对 ChatGPT 的攻击成功率最高可超过 40%.

表 8　面向情感分类的中文 BERT 鲁棒性评估 (软标签)

方法	基于攻击成功率的鲁棒性分数 *rsu*	基于置信度的鲁棒性分数 *rsv*	联合鲁棒性分数 *rs*
C_T	9.99	10.00	10.00
G_R	9.17	5.72	7.45
A_SC	9.03	5.84	7.44
A_G	7.37	4.94	6.16
A_Sh	9.83	8.55	9.19
A_P	9.46	6.39	7.93
A_Sy	6.93	3.64	5.29
TF	5.60	3.64	4.62
CBT	4.13	2.47	3.30
平均值	7.95	5.69	6.82
扩大系数 β		1	

4.2.2　对抗文本流畅性评估结果

由于本工作利用了对抗文本的可迁移性, 利用间接攻击的方式攻击 ChatGPT, 即在攻击 ChatGPT 时所用的对抗文本均为针对中文 BERT 模型生成的对抗文本, 因此本文仅需评估针对中文 BERT 模型生成的对抗文本的流畅性. 本节首先展示了使用所提的基于 OAD 的对抗文本流畅性打分方法 OFS 对面向中文 BERT 生成的对抗文本的流畅性评估结果, 随后通过计算原始文本与对抗文本相似度的方式进一步证实了 OFS 的有效性. 这两部分的具体实验结果分别在第 4.2.2.1 节和第 4.2.2.2 节中进行展示与分析.

4.2.2.1　基于 OAD 的流畅性评估结果

本工作用于生成对抗文本的 DL 模型仅为一个, 因此仅需计算在各种对抗攻击的方法下, 面向中文 BERT 模型生成的对抗文本流畅性分数. 根据第 3.2 节中所提的方法, 可知计算流畅性分数时需要各攻击方法下无法被 ChatGPT 分类的对抗文本占比. 表 9 和表 10 分别展示了在新闻分类和情感倾向性分类数据集上, 各攻击方法下无法被 ChatGPT 分类的对抗文本占比、对应的流畅性分数 fs 以及扩大系数 β 值.

表 9　面向新闻分类时各方法生成的无法被 ChatGPT 分类的对抗文本占比 (%) 及对应的流畅性分数

方法	不同扰动比例 α 下无法被ChatGPT分类的对抗文本占比 (%)									*fs*
	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.50	
C_T	1.24	1.24	1.36	2.11	2.23	2.35	2.60	2.35	2.73	9.60
G_R	6.32	9.29	12.52	13.75	14.75	18.09	20.45	21.07	23.79	6.89
A_SC	9.67	13.75	20.45	24.16	29.74	34.70	38.54	40.40	41.88	4.37
A_G	8.92	14.62	22.92	28.75	37.17	46.96	56.13	60.59	68.03	2.35
A_Sh	3.10	4.46	6.20	8.05	8.92	11.03	12.89	14.50	16.11	8.11
A_P	3.22	3.72	5.82	5.95	6.94	8.05	8.43	8.92	8.92	8.67
A_Sy	2.97	2.85	5.33	6.44	7.43	9.29	11.90	14.00	15.37	8.32
TF	4.46	5.82	8.43	9.79	11.40	14.13	16.73	18.22	21.31	7.55
CBT	6.82	7.56	10.90	12.76	16.85	21.19	24.04	26.27	30.98	6.50
平均流畅性分数 $\overline{fs}$					—					6.93
扩大系数 β					—					2

通过表 9 和表 10 可知, 在新闻分类和情感倾向分类这两种数据集上, A_G、A_SC 方法生成的对抗文本流畅性分数较低, 而 C_T 方法生成的对抗文本流畅性分数最高. 结合第 4.2.1 节中的实验结果可以看出, A_G 和 A_SC 方法作为一种字符级方法中攻击成功率较高的方法, 其攻击成功率在一定程度上源于其生成的对抗文本并不流畅且较难理解; 而 C_T 方法虽然能够生成较为流畅的对抗文本, 但过于流畅的文本使得 ChatGPT 能够轻易理解该文本的原意, 从而无法欺骗目标模型, 导致攻击成功率很低. 此外, 通过上述实验结果可知, 词语级对抗文本生成方法

的流畅性分数均保持在中等水平, 且这些方法的攻击成功率也远高于字符级的方法. 因此这些方法是 ChatGPT、中文 BERT 等目标模型在设计防御方法时需要重点考虑的攻击方法.

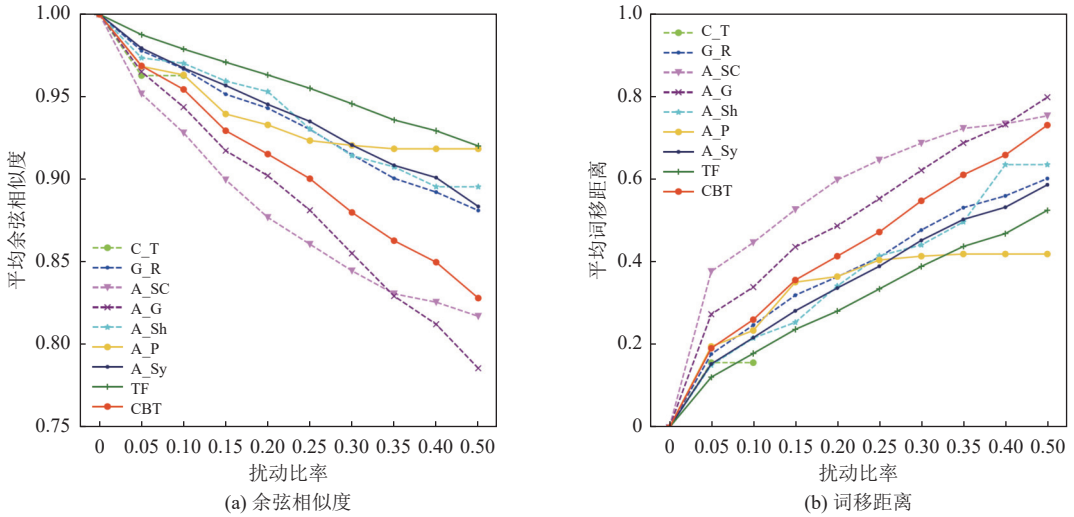
表 10　面向情感分类时各方法生成的无法被 ChatGPT 分类的对抗文本占比 (%) 及对应的流畅性分数

方法	不同扰动比例 α 下无法被ChatGPT分类的对抗文本占比 (%)									fs
	0.01	0.03	0.05	0.10	0.15	0.20	0.25	0.30	0.40	
C_T	1.39	1.04	1.39	1.62	1.73	2.89	2.31	2.42	2.31	9.43
G_R	3.35	3.46	3.58	5.20	7.39	9.01	10.62	11.89	13.39	7.74
A_SC	8.20	7.04	9.24	14.43	19.28	25.75	29.10	32.45	34.99	3.98
A_G	8.20	8.43	9.12	15.24	24.60	32.56	40.88	51.27	64.09	1.52
A_Sh	3.70	4.16	3.93	6.24	7.51	9.58	9.93	12.47	14.78	7.59
A_P	3.35	4.73	4.04	4.73	5.89	5.66	6.47	6.81	6.81	8.38
A_Sy	3.23	2.66	3.93	5.20	9.01	12.01	14.78	18.59	23.33	6.91
TF	8.43	8.55	10.05	11.66	15.70	17.21	19.05	24.13	29.91	5.18
CBT	10.39	11.66	13.05	17.32	20.09	23.90	26.21	27.83	31.64	3.93
平均流畅性分数 $\overline{fs}$					—					6.07
扩大系数 β					—					3

4.2.2.2　文本相似性评估结果

为了进一步验证所提的 OFS 方法的有效性, 本文使用了 4 种文本相似度计算方法对原文本与对抗文本之间的相似度进行全面评估. 虽然文本相似性与流畅性是两个截然不同的概念, 但文本相似性在一定程度上能够反映出对抗文本的流畅性. 若原文本与对抗文本之间的相似度很低, 则说明对抗文本在原文本的基础上进行了较大的扰动, 这些改动有较大的概率影响其原始语义以及词法和语法的正确性, 从而对其流畅性有较大的影响. 因此, 本文通过评估原文本与对抗文本之间的相似性, 来对各方法生成的对抗文本的流畅性进行进一步验证.

本文使用第 4.1.4 节中介绍的余弦相似度、词移距离、编辑距离以及杰卡德系数这 4 种文本相似度计算方法, 分别在新闻分类和情感倾向分类数据集上计算针对中文 BERT 模型生成的对抗文本与原始文本的相似度, 实验结果如图 4 和图 5 所示. 其中横坐标轴表示扰动比率, 纵坐标轴表示对应方法的文本相似度. 图例中虚线表示字符级对抗文本生成方法, 实线则表示词语级对抗文本生成方法.



(a) 余弦相似度　(b) 词移距离

图 4　新闻分类数据集上的文本相似度评估

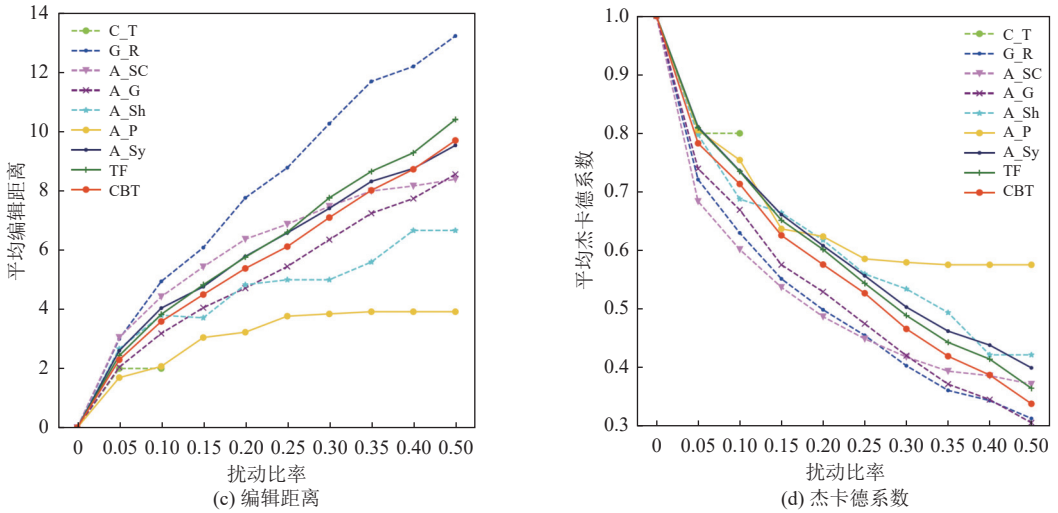


图 4　新闻分类数据集上的文本相似度评估 (续)

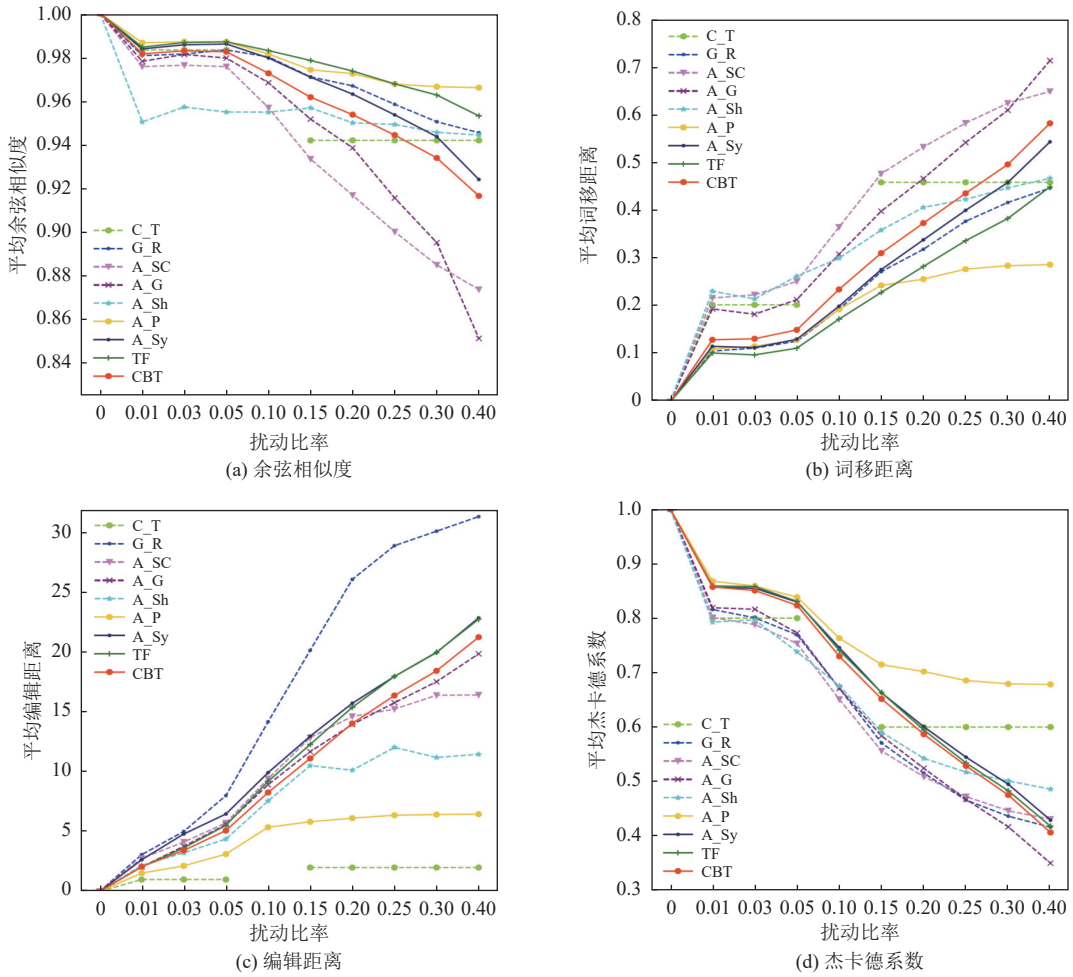


图 5　情感分类数据集上的文本相似度评估

从图 4 和图 5 中可以看出, 字符级方法 A_G 和 A_SC 生成的对抗文本与原始文本的相似度在 4 种评估指标上均展现了较低的水平, 而其他字符级方法生成的对抗文本与原始文本的相似度相对较高. 对于字符级的 C_T 方法来说, 由于其在某些扰动比率下针对中文 BERT 模型无法生成的对抗文本, 在图 4 和图 5 中的部分扰动率下对其值进行缺省处理, 因此使用文本相似度的评估手段并不能很好地反映出 C_T 方法生成的对抗文本的流畅性, 这也进一步体现出使用基于 OAD 对抗文本流畅性评估方法的优越性. 词语级对抗方法 TF 生成的对抗文本在余弦相似度以及词移距离这两个评价指标上均有较好的表现. 在编辑距离以及杰卡德系数这两个指标上, TF、A_Sy 以及 CBT 这 3 种词语级方法生成的对抗文本在文本相似性方面相差不大, 均保持在中等水平.

综上所述, 文本相似度评估能够在一定程度上反映出各方法生成的对抗文本的流畅性. 一方面, 通过文本相似度评估所得到结论与所提的 OFS 方法计算出的每种方法生成对抗文本的流畅性得分在很大程度上具有一致性; 另一方面, 对于那些在某些扰动比率下无法生成对抗文本的方法, 文本相似度的评估很难正确反映其生成的对抗文本的流畅性, 而本文所提的 OFS 却能对此进行较为合理的评估.

4.3 细节讨论

为了进一步讨论提示语对 ChatGPT 鲁棒性评估结果以及对抗文本流畅性评估结果的影响, 本节中对使用不包含无法分类标签的提示语对 ChatGPT 进行提问. 对于新闻分类和情感倾向分类任务, 具体的提示语分别如下所示.

面向新闻分类任务的提示语: “帮我对给出的文本进行文本分类, 类别共有 5 种, 分别是科技、教育、财经、社会、运动, 分别用数字 0–4 表示. 回答时请直接给出类别对应的数字, 不必说明原因.”

面向情感倾向分类任务的提示语: “请帮我确定以下文本的情感极性. 用 0 表示消极情感, 1 表示积极情感 (只给出结果而不作解释, 且对整体文本只给出一个标签即可, 无需给出多个标签).”

使用上述提示语对 ChatGPT 进行提问, 最终的 ChatGPT 鲁棒性评估结果以及对抗文本流畅性评估结果分别见第 4.3.1 节和第 4.3.2 节.

4.3.1 提示语对 ChatGPT 鲁棒性评估的影响

表 11 和表 12 分别展示了在提示语不包含无法分类的情况时, 在新闻分类和情感倾向性分类数据集上, 各种攻击方法对 ChatGPT 的攻击成功率、对应的鲁棒性分数 rs 以及扩大系数 β 值. 由于需要与提示语包含无法分类时的实验结果在同一基准上进行对比, 因此本节参考了第 3.1.3.2 节中提到的思想, 分别计算这两种情况下的扩大系数, 选择更小的值作为最终的扩大系数 β. 以新闻数据集为例, 最终计算出的 β 值为 2, 与表 1 中的 β 值相等, 因此直接将表 11 与表 1 进行对比即可, 情感分类数据集同理. 通过对比可知, 与包含无法分类标签的提示语相比, 当提示语中不包含无法分类的情况时, 面对各种攻击的 ChatGPT 鲁棒性分数及平均鲁棒性分数均有明显下降, 这说明 ChatGPT 在这种情况下更容易被对抗文本迷惑. 该实验结果证实了提示语对 ChatGPT 鲁棒性的表现有较大的影响, 信息较多的提示语能够在一定程度上提升其面向分类任务的鲁棒性.

表 11 改变提示语的情况下面向新闻分类时各方法对 ChatGPT 的攻击成功率 (%) 及鲁棒性评估

方法	不同扰动比例 α 下的攻击成功率 (%)									rs
	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.50	
C_T	7.92	8.19	8.06	9.17	9.03	9.44	10.00	9.44	9.58	8.20
G_R	13.06	15.00	18.61	21.11	22.08	25.14	26.39	27.50	27.78	5.63
A_SC	12.92	14.86	18.75	21.94	24.58	25.00	26.25	27.50	27.64	5.57
A_G	16.39	22.36	26.94	30.83	34.44	38.89	41.11	41.53	44.86	3.39
A_Sh	10.69	11.67	13.61	15.28	15.56	16.67	18.61	18.06	19.86	6.89
A_P	9.31	11.53	13.33	13.89	13.47	14.03	14.31	14.58	15.14	7.34
A_Sy	11.81	13.61	14.44	15.69	18.06	20.42	21.25	22.36	22.08	6.45
TF	12.36	14.31	18.89	21.53	22.50	24.86	25.28	25.97	27.64	5.70
CBT	18.61	21.39	25.00	28.19	31.39	35.56	36.39	37.36	40.42	3.90
平均鲁棒性分数 $\overline{rs}$					—					5.90
扩大系数 β					—					2

表 12　改变提示语的情况下面向情感分类时各方法对 ChatGPT 的攻击成功率 (%) 及鲁棒性评估

方法	不同扰动比例 α 下的攻击成功率 (%)									rs
	0.01	0.03	0.05	0.10	0.15	0.20	0.25	0.30	0.40	
C_T	0.66	0.87	0.77	0.77	0.66	0.66	0.66	0.55	0.77	9.93
G_R	2.30	2.08	2.84	5.14	4.48	4.26	4.48	4.92	5.36	9.60
A_SC	4.15	4.70	5.14	7.98	10.82	11.37	13.01	15.41	17.05	9.00
A_G	5.03	5.90	7.65	13.01	18.03	21.64	22.95	27.54	33.77	8.27
A_Sh	2.73	2.84	3.06	4.59	5.03	5.14	6.12	6.78	6.34	9.53
A_P	2.51	2.19	2.95	3.83	3.50	4.92	5.03	4.26	4.59	9.62
A_Sy	2.73	3.39	4.04	6.12	9.40	10.71	13.01	14.64	18.80	9.08
TF	9.07	9.84	12.35	19.78	26.99	31.48	34.97	38.91	43.17	7.48
CBT	13.99	17.16	20.22	32.13	41.20	48.42	53.55	55.85	59.78	6.20
平均鲁棒性分数 $\overline{rs}$					—					8.75
扩大系数 β					—					1

4.3.2　提示语对对抗文本流畅性评估的影响

表 13 和表 14 分别展示了在提示语不包含无法分类的情况时, 在新闻分类和情感倾向性分类数据集上, 各攻击方法下无法被 ChatGPT 分类的对抗文本占比、对应的流畅性分数 fs 以及扩大系数 β 值. 在本实验中, 同样需要按类似第 3.1.3.2 节中的思路, 与包含无法分类情况下的实验结果统一 β 值. 最终, 在新闻分类数据集上 β 值统一为 2; 在情感分类数据集上 β 值统一为 3.

表 13　改变提示语的情况下面向新闻分类时各方法生成的无法被 ChatGPT 分类的对抗文本占比 (%) 及对应的流畅性分数

方法	不同扰动比例 α 下无法被ChatGPT分类的对抗文本占比(%)									fs
	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.50	
C_T	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	10.00
G_R	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.14	10.00
A_SC	0.14	0.14	0.14	0.28	0.42	0.97	1.53	1.11	2.36	9.84
A_G	0.00	0.00	0.00	0.14	0.28	0.28	1.39	1.67	3.06	9.85
A_Sh	0.00	0.00	0.14	0.14	0.14	0.00	0.00	0.00	0.00	9.99
A_P	0.14	0.14	0.14	0.14	0.14	0.00	0.00	0.00	0.00	9.98
A_Sy	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.14	10.00
TF	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.14	10.00
CBT	0.00	0.00	0.42	0.83	0.56	0.83	0.97	0.83	1.11	9.88
平均流畅性分数 $\overline{fs}$					—					9.95
扩大系数 β					—					2

分别将表 9 与表 13、表 10 与表 14 进行对比可知, 相比于包含无法分类标签的提示语, 使用不包含无法分类的提示语评估对抗文本的流畅性时, 各方法生成的对抗文本对应的流畅性分数普遍均有大幅提高, 甚至有多种方法出现流畅性分数为满分 10 分的情况. 该情况的出现有两方面的原因. 一方面, 因为需要与包含无法分类标签的提示语放在同一基准下进行比较, 因此两者 β 值需一致. 这就导致了单独观察使用不包含无法分类标签的提示语对实验结果的影响时, 各方法间的区分度由于较小的 β 值相差并不明显, 很难看出方法之间的区分度. 另一方面, 提示语的改变在一定程度上也会降低 ChatGPT 对自然语言的理解能力. 例如, 第 4.2.2 节中的几组的实验已验证 A_SC 方法生成的对抗文本流畅性较低, 但其在情感倾向分类数据集上, 在各扰动比率下对应的被 ChatGPT 的误分类比例均为 0, 最终的流畅性分数为满分 10 分, 这显然是较为不合理的结论. 然而, 在 ChatGPT 没有得到输入的文本可能不属于任意一种标签的情况下, 其仍然能够以极小的比例将对抗文本认为是无法分类的文本. 这说明 ChatGPT 自身仍然具有一定的纠错能力, 但这种能力往往更加依赖于人类给出的提示词.

表 14 改变提示语的情况下面向情感分类时各方法生成的无法被 ChatGPT 分类的对抗文本占比 (%)及对应的流畅性分数

方法	不同扰动比例 α 下无法被ChatGPT分类的对抗文本占比 (%)									fs
	0.01	0.03	0.05	0.10	0.15	0.20	0.25	0.30	0.40	
C_T	0.11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	10.00
G_R	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	10.00
A_SC	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	10.00
A_G	0.11	0.00	0.11	0.11	0.22	0.00	0.11	0.11	0.55	9.96
A_Sh	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	10.00
A_P	0.11	0.11	0.00	0.11	0.11	0.00	0.00	0.00	0.11	9.98
A_Sy	0.11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	10.00
TF	0.11	0.11	0.11	0.00	0.00	0.00	0.00	0.00	0.11	9.99
CBT	0.11	0.22	0.22	0.11	0.11	0.22	0.22	0.44	0.22	9.94
平均流畅性分数 $\overline{fs}$					—					9.99
扩大系数 β					—					3

上述实验结果同样证明了提示词对所提的 OFS 的评估效果有显著影响. 当提示词不完善时, 虽然 ChatGPT 仍然保留一定的纠错能力, 但其对于对抗文本流畅性会出现较为严重的错误认知, 导致其评估水平大幅下降.

5 总 结

在中文对抗攻击下, 本工作对 ChatGPT 的鲁棒性进行了可量化评估. 本文引入了一个新的概念 OAD, 基于 OAD 设计了一种目标模型鲁棒性的量化评估方法 ORS. 本文分别面向输出为硬标签和输出为软标签的目标模型提出了不同的评价指标. 其中前者利用了不同中文对抗文本生成方法在各种扰动比率下的攻击成功率, 而后者则在前者的基础上引入置信度信息. 与此同时, 本工作将 OAD 的应用扩展到对抗文本流畅性的评估中. 相比于以往需要人类参与的方法, 所提的基于 OAD 的流畅性评估方法 OFS 利用 ChatGPT 强大的自然语言理解能力, 能够实现自动化评估, 大幅降低了评估成本. 实验结果表明, 相比于字符级方法, 词语级对抗文本生成方法能够以较强的攻击成功率, 在一定程度上破坏 ChatGPT 的鲁棒性. 与此同时, 词语级方法生成的对抗文本拥有较好的流畅性, 且与原始文本保持较高的相似度. 然而, ChatGPT 的鲁棒性以及对对抗文本流畅性的评估能力在很大程度上与提示语相关. 当提示语信息较少时, ChatGPT 对自然语言的理解能力及纠错能力会大幅下降, 导致其更容易受到对抗攻击, 且难以正确评估对抗文本的流畅性. 在未来的工作中, 将进一步研究 ChatGPT 及其他 LLM 在英文对抗攻击下的鲁棒性, 继而尝试使用可解释的对抗攻击手段绕过 LLM 自身的安全机制, 并提供相应的防御方法.

References:

[1] Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, Goodfellow I, Fergus R. Intriguing properties of neural networks. In: Proc. of the 2014 Int'l Conf. on Learning Representations. OpenReview.net. 2014.

[2] Goodfellow IJ, Shlens J, Szegedy C. Explaining and harnessing adversarial examples. arXiv:1412.6572, 2015.

[3] Gao J, Lanchantin J, Soffa ML, Qi YJ. Black-box generation of adversarial text sequences to evade deep learning classifiers. In: Proc. of the 2018 IEEE Security and Privacy Workshops. San Francisco: IEEE, 2018. 50–56. [doi: 10.1109/SPW.2018.00016]

[4] Wang WQ, Wang R, Wang LN, Tang BX. Adversarial examples generation approach for tendency classification on Chinese texts. Ruan Jian Xue Bao/Journal of Software, 2019, 30(8): 2415–2427 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5765.htm [doi: 10.13328/j.cnki.jos.005765]

[5] Zhu KJ, Wang JD, Zhou JH, Wang ZC, Chen H, Wang YD, Yang LY, Ye W, Zhang Y, Gong N, Xie X. PromptRobust: Towards evaluating the robustness of large language models on adversarial prompts. arXiv:2306.04528, 2024.

[6] Liang B, Li HC, Su MQ, Bian P, Li XR, Shi WC. Deep text classification can be fooled. In: Proc. of the 27th Int'l Joint Conf. on Artificial Intelligence. Stockholm: AAAI Press, 2018. 4208–4215.

[7] Wang WQ, Wang R, Wang LN, Wang ZB, Ye AS. Towards a robust deep neural network in texts: A survey. arXiv:1902.07285, 2021.

[8] Alzantot M, Sharma Y, Elgohary A, Ho BJ, Srivastava M, Chang KW. Generating natural language adversarial examples. In: Proc. of the 2018 Conf. on Empirical Methods in Natural Language Processing. Brussels: Association for Computational Linguistics, 2018. 2890–2896. [doi: 10.18653/v1/D18-1316]

[9] Zang Y, Qi FC, Yang CH, Liu ZY, Zhang M, Liu Q, Sun MS. Word-level textual adversarial attacking as combinatorial optimization. In: Proc. of the 58th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, 2020. 6066–6080. [doi: 10.18653/v1/2020.acl-main.540]

[10] Jin D, Jin ZJ, Zhou JT, Szolovits P. Is BERT really robust? A strong baseline for natural language attack on text classification and entailment. In: Proc. of the 34th AAAI Conf. on Artificial Intelligence. New York: AAAI Press, 2020. 8018–8025. [doi: 10.1609/aaai.v34i05.6311]

[11] Li LY, Ma RT, Guo QP, Xue XY, Qiu XP. BERT-ATTACK: Adversarial attack against BERT using BERT. In: Proc. of the 2020 Conf. on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2020. 6193–6202. [doi: 10.18653/v1/2020.emnlp-main.500]

[12] Ren SH, Deng YH, He K, Che WX. Generating natural language adversarial examples through probability weighted word saliency. In: Proc. of the 57th Annual Meeting of the Association for Computational Linguistics. Florence: Association for Computational Linguistics, 2019. 1085–1097. [doi: 10.18653/v1/P19-1103]

[13] Xu JC, Du QF. Adversarial attacks on text classification models using layer-wise relevance propagation. Int'l Journal of Intelligent Systems, 2020, 35(9): 1397–1415. [doi: 10.1002/int.22260]

[14] Li JF, Ji SL, Du TY, Li B, Wang T. TextBugger: Generating adversarial text against real-world applications. In: Proc. of the 26th Annual Network and Distributed System Security Symp. San Diego: The Internet Society, 2019. [doi: 10.14722/ndss.2019.23138]

[15] Garg S, Ramakrishnan G. BAE: BERT-based adversarial examples for text classification. In: Proc. of the 2020 Conf. on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2020. 6174–6181. [doi: 10.18653/v1/2020.emnlp-main.498]

[16] Li DQ, Zhang YZ, Peng H, Chen LQ, Brockett C, Sun MT, Dolan B. Contextualized perturbation for textual adversarial attack. In: Proc. of the 2021 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, 2021. 5053–5069. [doi: 10.18653/v1/2021.naacl-main.400]

[17] Zhang ZH, Liu MX, Zhang C, Zhang YM, Li Z, Li Q, Duan HX, Sun DH. Argot: Generating adversarial readable Chinese texts. In: Proc. of the 29th Int'l Joint Conf. on Artificial Intelligence. Yokohama, 2021. 2533–2539. [doi: 10.24963/ijcai.2020/351]

[18] Nuo C, Chang GQ, Gao HC, Pei G, Zhang Y. WordChange: Adversarial examples generation approach for Chinese text classification. IEEE Access, 2020, 8: 79561–79572. [doi: 10.1109/ACCESS.2020.2988786]

[19] Tong X, Wang LN, Wang RZ, Wang JY. A generation method of word-level adversarial samples for Chinese text classification. Netinfo Security, 2020, 20(9): 12–16 (in Chinese with English abstract). [doi: 10.3969/j.issn.1671-1122.2020.09.003]

[20] Ou HX, Yu L, Tian SW, Chen X. Chinese adversarial examples generation approach with multi-strategy based on semantic. Knowledge and Information Systems, 2022, 64(4): 1101–1119. [doi: 10.1007/s10115-022-01652-1]

[21] Zhang YT, Ye L, Tang HL, Zhang HL, Li S. Chinese BERT attack method based on masked language model. Ruan Jian Xue Bao/Journal of Software, 2024, 35(7): 3392–3409 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6932.htm [doi: 10.13328/j.cnki.jos.006932]

[22] He XL, Lyu LJ, Sun LC, Xu QK. Model extraction and adversarial transferability, your BERT is vulnerable! In: Proc. of the 2021 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, 2021. 2006–2012. [doi: 10.18653/v1/2021.naacl-main.161]

[23] Ebrahimi J, Rao AY, Lowd D, Dou DJ. HotFlip: White-box adversarial examples for text classification. In: Proc. of the 56th Annual Meeting of the Association for Computational Linguistics (Vol. 2: Short Papers). Melbourne: Association for Computational Linguistics, 2018. 31–36. [doi: 10.18653/v1/P18-2006]

[24] Shi YC, Han YH. Metric system and its completeness of adversarial robustness evaluation. Ruan Jian Xue Bao/Journal of Software, 2024 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/7172.htm [doi: 10.13328/j.cnki.jos.007172]

[25] Yoo JY, Morris JX, Lifland E, Qi YJ. Searching for a search method: Benchmarking search algorithms for generating NLP adversarial examples. In: Proc. of the 3rd BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP. Association for Computational Linguistics, 2020. 323–332. [doi: 10.18653/v1/2020.blackboxnlp-1.30]

[26] Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of deep bidirectional Transformers for language understanding. In: Proc. of the 2019 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Vol. 1: Long and Short Papers). Minneapolis: Association for Computational Linguistics, 2019. 4171–4186. [doi: 10.18653/v1/N19-1423]

[27] Mrkšić N, Séaghdha DÓ, Thomson B, Gašić M, Rojas-Barahona LM, Su PH, Vandyke D, Wen TH, Young S. Counter-fitting word vectors to linguistic constraints. In: Proc. of the 2016 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. San Diego: Association for Computational Linguistics, 2016. 142–148. [doi: 10.18653/v1/N16-1018]

[28] Gu JJ, Li XS. The effects of character transposition within and across words in Chinese reading. Attention, Perception, & Psychophysics, 2015, 77(1): 272–281. [doi: 10.3758/s13414-014-0749-5]

[29] Xu LH, Lin HF, Pan Y, Ren H, Chen JM. Constructing the affective lexicon ontology. Journal of the China Society for Scientific and Technical Information, 2008, 27(2): 180–185 (in Chinese with English abstract). [doi: 10.3969/j.issn.1000-0135.2008.02.004]

[30] Sun MS, Li JY, Guo ZP, Zhao Y, Zheng YB, Si XC, Liu ZY. THUCTC: An efficient Chinese text classifier [Technnical Report]. Beijing: Tsinghua University, 2016 (in Chinese with English abstract). https://gitcode.com/gh_mirrors/th/THUCTC

[31] ChineseNlpCorpus. GitHub, 2018. https://github.com/SophonPlus/ChineseNlpCorpus

[32] Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv:1412.6980, 2017.

[33] Kusner MJ, Sun Y, Kolkin NI, Weinberger KQ. From word embeddings to document distances. In: Proc. of the 32nd Int'l Conf. on Machine Learning. Lille: JMLR.org, 2015. 957–966.

[34] Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. arXiv:1301.3781, 2013.

附中文参考文献:

[4] 王文琦, 汪润, 王丽娜, 唐奔宵. 面向中文文本倾向性分类的对抗样本生成方法. 软件学报, 2019, 30(8): 2415–2427. http://www.jos.org.cn/1000-9825/5765.htm [doi: 10.13328/j.cnki.jos.005765]

[19] 仝鑫, 王罗娜, 王润正, 王靖亚. 面向中文文本分类的词级对抗样本生成方法. 信息网络安全, 2020, 20(9): 12–16. [doi: 10.3969/j.issn.1671-1122.2020.09.003]

[21] 张云婷, 叶麟, 唐浩林, 张宏莉, 李尚. 基于掩码语言模型的中文 BERT 攻击方法. 软件学报, 2024, 35(7): 3392–3409. http://www.jos.org.cn/1000-9825/6932.htm [doi: 10.13328/j.cnki.jos.006932]

[24] 石育澄, 韩亚洪. 对抗鲁棒性评估的指标体系及其完备性. 软件学报, 2024. http://www.jos.org.cn/1000-9825/7172.htm [doi: 10.13328/j.cnki.jos.007172]

[29] 徐琳宏, 林鸿飞, 潘宇, 任惠, 陈建美. 情感词汇本体的构造. 情报学报, 2008, 27(2): 180–185. [doi: 10.3969/j.issn.1000-0135.2008.02.004]

[30] 孙茂松, 李景阳, 郭志芃, 赵宇, 郑亚斌, 司宪策, 刘知远. THUCTC: 一个高效的中文文本分类工具包 [技术报告]. 北京: 清华大学. 2016. https://gitcode.com/gh_mirrors/th/THUCTC

张云婷(1997－), 女, 博士生, 主要研究领域为人工智能安全, 文本对抗, 大语言模型安全.

李柏松(1980－), 男, CCF 专业会员, 主要研究领域为计算机反病毒技术,网络安全威胁对抗.

叶麟(1982－), 男, 博士, 副教授, CCF 专业会员, 主要研究领域为 P2P 网络, 网络安全, 网络测量, 云计算.

张宏莉(1973－), 女, 博士, 教授, 博士生导师, CCF 专业会员, 主要研究领域为网络与信息安全, 云安全, 隐私保护.