软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2023,34(7):3313−3328 [doi: 10.13328/j.cnki.jos.006525]

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

基于义原级语句稀释法的文本对抗攻击能力强化方法*

叶文滔[1], 张 敏[2], 陈仪香[3]

[1](华东师范大学 软件工程学院, 上海 200062)

[2](上海市高可信计算重点实验室, 上海 200062)

[3](教育部软硬件协同设计技术与应用工程研究中心, 上海 200062)

通信作者: 张敏, E-mail: mzhang@sei.ecnu.edu.cn

摘 要: 随着近年来机器学习方法在自然语言处理领域的应用越发广泛, 自然语言处理任务的安全性也引起了研究者们重视. 现有研究发现, 向样本施加细微扰动可能令机器学习模型得到错误结果, 这种方法称之为对抗攻击. 文本对抗攻击能够有效发现自然语言模型的弱点从而进行改进. 然而, 目前的文本对抗攻击方法都着重于设计复杂的对抗样本生成策略, 对抗攻击成功率提升有限, 且对样本进行高侵入性修改容易导致样本质量下降. 如何更简单、更高效地提升对抗攻击效果, 并输出高质量对抗样本已经成为重要需求. 为解决此问题, 从改进对抗攻击过程的新角度, 设计了义原级语句稀释法 (sememe-level sentence dilution algorithm, SSDA) 及稀释池构建算法 (dilution pool construction algorithm, DPCA). SSDA 是一种可以自由嵌入经典对抗攻击过程中的新过程, 它利用 DPCA 构建的稀释池先对输入样本进行稀释, 再进行对抗样本生成. 在未知文本数据集与自然语言模型的情况下, 不仅能够提升任意文本对抗攻击方法的攻击成功率, 还能够获得相较于原方法更高的对抗样本质量. 通过对不同文本数据集、稀释池规模、自然语言模型, 以及多种主流文本对抗攻击方法进行对照实验, 验证了 SSDA 对文本对抗攻击方法成功率的提升效果以及 DPCA 构建的稀释池对 SSDA 稀释能力的提升效果. 实验结果显示, SSDA 稀释过程能够比经典对抗攻击过程发现更多模型漏洞, 且 DPCA 能够帮助 SSDA 在提升成功率的同时进一步提升对抗样本的文本质量.

关键词: 对抗攻击; 机器学习; 自然语言处理; 边界值分析; 义原

中图法分类号: TP309

中文引用格式: 叶文滔, 张敏, 陈仪香. 基于义原级语句稀释法的文本对抗攻击能力强化方法. 软件学报, 2023, 34(7): 3313–3328. http://www.jos.org.cn/1000-9825/6525.htm

英文引用格式: Ye WT, Zhang M, Chen YX. Enhancement of Textual Adversarial Attack Ability Based on Sememe-level Sentence Dilution Algorithm. Ruan Jian Xue Bao/Journal of Software, 2023, 34(7): 3313–3328 (in Chinese). http://www.jos.org.cn/1000-9825/6525.htm

Enhancement of Textual Adversarial Attack Ability Based on Sememe-level Sentence Dilution Algorithm

YE Wen-Tao[1], ZHANG Min[2], CHEN Yi-Xiang[3]

[1](Software Engineering Institute, East China Normal University, Shanghai 200062, China)

[2](Shanghai Key Laboratory of Trustworthy Computing, Shanghai 200062, China)

[3](MOE Engineering Research Center for Software/Hardware Co-design Technology and Application, Shanghai 200062, China)

Abstract: With machine learning widely applied to the natural language processing (NLP) domain in recent years, the security of NLP tasks receives growing natural concerns. Existing studies found that small modifications in examples might lead to wrong machine learning

* 基金项目: 科技部重点研发项目 (2020AAA0107800); 国家自然科学基金 (61672012)
收稿时间: 2021-06-22; 修改时间: 2021-09-22; 采用时间: 2021-10-25; jos 在线出版时间: 2022-09-09
CNKI 网络首发时间: 2022-11-15

predictions, which was also called adversarial attack. The textual adversarial attack can effectively reveal the vulnerability of NLP models for improvement. Nevertheless, existing textual adversarial attack methods all focus on designing complex adversarial example generation strategies with a limited improvement of success rate, and the highly invasive modifications bring the decline of textual quality. Thus, a simple and effective method with high adversarial example quality is in demand. To solve this problem, the sememe-level sentence dilution algorithm (SSDA) and the dilution pool construction algorithm (DPCA) are proposed from a new perspective of improving the process of adversarial attack. SSDA is a new process that can be freely embedded into the classical adversarial attack workflow. SSDA first uses dilution pools constructed by DPCA to dilute the original examples, then generates adversarial examples through those diluted examples. It can not only improve the success rate of any adversarial attack methods without any limit of datasets or victim models but also obtain higher adversarial example quality compared with the original method. Through the experiments of different datasets, dilution pools, victim models, and textual adversarial attack methods, it is successfully verified the improvement of SSDA on the success rate and proved that dilution pools constructed by DPCA can further enhance the dilution ability of SSDA. The experiment results demonstrate that SSDA reveals more vulnerabilities of models than classical methods, and DPCA can help SSDA to improve success rate with higher adversarial example quality.

Key words: adversarial attack; machine learning; natural language processing (NLP); boundary value analysis; sememe

1 研究背景

机器学习已经广泛应用于现实场景, 并在多个领域有出色表现. 但机器学习模型在数据层、模型层及应用层等多个层面的鲁棒性问题也引起了学术界和工业界的广泛关注[1]. 对抗攻击就是在机器学习模型预测阶段的一种鲁棒性检验方法, 研究者通过精心构造的对抗样本使得机器学习模型预测出错[2], 发现模型的弱点, 其中受到攻击的模型也称为受害者模型. 通过对抗样本, 可以对受害者模型进行补充训练, 从而增强模型的泛化能力.

自然语言处理 (natural language processing, NLP) 领域的对抗攻击一般对文本样本展开. 文本对抗攻击方法通过对文本样本进行扰动, 在不改变样本实际分类的情况下, 使机器学习模型改变对该样本的预测结果.

不同于计算机视觉等领域中的样本保持连续性的特点, 自然语言处理领域的文本样本是离散的. 通常我们对图片进行一定程度地扰动不会影响人类的判断, 甚至可以做到人类无法察觉样本受到攻击, 但文本的细微变化就有可能完全逆转人类的理解, 导致攻击无效. 因此, 文本对抗攻击也面临着更大挑战, 研究者们在提高对抗攻击成功率的同时, 为了保证文本样本质量, 需要付出很大努力.

Jia 等人[3]早期通过语句插入等方式生成对抗样本, 成功使得多种已发布的问答系统失效, 揭示了文本对抗攻击的可研究性. Zhao 等人[4]通过构造一组逆变器与生成器对文本样本进行再编码, 实现语句级对抗攻击, 能够生成更接近人类真实表达的对抗样本, 提升了对抗样本的质量. Iyyer 等人[5]通过训练一个语法控制释意网络, 将原始输入按照指定语法改述生成对抗样本, 能够保证对抗样本符合语法规范, 并提高模型在语法变化上的鲁棒性. 也有一些方法迁移了其他领域的思想实现对抗攻击, 如 Eger 等人[6]开发了一种基于视觉相似性进行字符替换的策略 VIPER, 随机替换在视觉嵌入空间内的最相似近邻字符, 具有不错的攻击效率. Zang 等人[7]从语言学知识本身入手, 将语言学概念上的“义原”引入文本对抗攻击, 在多个数据集与模型上验证了有比经典的同义词替换法更好的攻击效果.

所谓义原, 即原子语义, 是语言学意义上的最小的、不可再分的语义单位, 通常视为词语的语义标签, 能够最准确地还原词语本意[8]. 语言学领域对义原的研究可以追溯至 20 世纪 20 年代. 我国学者董振东与董强花费了几十年时间构建了基于义原的大型中英文语言信息库 HowNet[9,10], 为 NLP 领域做出了巨大贡献. 基于 HowNet 开源的 OpenHowNet[11]使得更多 NLP 任务基于义原实现成为可能.

经典的对抗攻击方法着重于改进对抗样本生成方法, 对抗样本生成策略愈发复杂, 但能够实现的成功率提升也愈发有限. 本文引入义原相关研究, 将机器学习决策边界理论与传统软件测试领域的“边界值分析”迁移至文本对抗攻击领域, 提出义原级语句稀释法. 通过将语句稀释法植入传统的对抗攻击过程中, 实现全新的对抗攻击过程, 从改进对抗攻击过程的新角度提升了对抗攻击成功率, 并维持了生成的对抗样本的质量.

2 相关工作

本节将介绍本文涉及到的经典文本对抗攻击方法, 并对基本概念、使用的工具和相关知识进行说明.

2.1 文本对抗攻击方法

根据攻击的最小扰动粒度, 我们可以将文本对抗攻击方法分为语句级、词语级与字符级方法. 语句级的对抗样本需要维持样本的宏观语义, 其一般方法如句式变换插入[3]、再编码[4]、语法改述[5]等, SCPNs (syntactically controlled paraphrase networks)[5]是一种典型的通过语法改述生成对抗样本的方法. 语句级扰动通常导致极高的文本修改率, 样本质量难以控制, 且攻击成功率没有明显优势. 字符级方法在细粒度上操作语句, 如字符级替换与增删策略, 向量距离扰动等, 前文介绍的 VIPER[6]就是一种通过视觉相似性实行字符替换的方法. Ebrahimi 等人[12]提出的白盒字符级对抗攻击方法 HotFlip 基于梯度优化实行字符替换. 字符变化对词语及句子的影响难以预计, 同样容易引发文本质量下降, 目前 Pruthi 等人[13]已经提出了能够有效识别字符级错误的对抗防御模型. 还有一些方法结合了字符级与词语级方法的特点, 李进锋等人[14,15]提出的 TextBugger 在选择单词的语义最近邻进行替换后, 进一步会对语句添加预设的字符扰动, 有较高的攻击强度.

从近几年的研究成果来看, 词语级的对抗攻击方法在生成对抗样本质量、攻击成功率等方面往往具有更好的综合表现. Ren 等人[16]提出了基于分类概率变化进行词语级替换的 PWWS (probability weighted word saliency), 算法综合考虑了词语替换后模型分类概率的变化程度以及词语的显著性两个因素, 相较于同类方法攻击成功率有明显提升. Alzantot 等人[17]提出的基于遗传算法的词语级替换方法, 将遗传算法应用于对抗攻击, 在情感分析任务上以 14.7% 的较低文本修改率取得了 97% 的成功率, 本文将该方法简称为 Genetic. Zang 等人[7]提出了基于离散粒子群算法加速搜索的义原级词语替换方法, 首次使用从义原维度实现对抗攻击, 在多个数据集与模型上验证了义原相较经典同义词替换的优势, 对 BiLSTM 模型以 IMDB 数据集作为输入进行对抗攻击, 达到了 100% 的成功率, 而基于同义词替换方法能达到的最好结果是 98.7%.

替换与增删策略是词语级对抗攻击的通常方法: 将离散的句子视为一个可搜索的向量空间, 向量成员即每个位置的词语, 根据所采用的算法对向量成员检索后替换或增删词语, 直到成功攻击或攻击失败. 相较而言, 我们对词语进行同义替换或按照合理策略增删, 样本含义变化通常较小, 也易于控制, 可以自主设定迭代次数控制攻击的有效性, 便于在维持样本质量的情况下不断探寻更高的攻击成功率. Samanta 等人[18]通过在 IMDB 数据集与 Twitter 数据集上对重要词语进行递进的增删和替换, 验证了这一类词语级攻击策略的有效性.

为了更高效的开展对抗攻击实验, 清华大学 THUNLP 实验室开发的专门用于文本对抗攻击及防御实验的开源工具 OpenAttack[19]作为文本对抗攻击的有效工具. OpenAttack 目前已集成了许多 NLP 对抗攻击领域主流的攻击方法, 包括前文提及的 SCPNs, PWWS, Genetic, VIPER 等.

2.2 义 原

前文对义原的基本概念进行过介绍. “基于义原”与“基于同义词”是两个不同的概念, 在“基于义原”的情况下, 词语可能有多个义原标签, 也拥有了更多维度的搜索空间, 可以找到许多在同义词维度下难以发现的替换规则, 图 1 为“基于义原”进行词语替换的一个例子. “基于同义词”仍然是在完整的词语级层面探索词语, 搜索空间维度单一, 难以覆盖所有可能替换的情况. 除了前文提及的义原在对抗攻击领域的出色表现, 近年来 NLP 领域的研究已经有越来越多应用义原开展的研究[20–24].

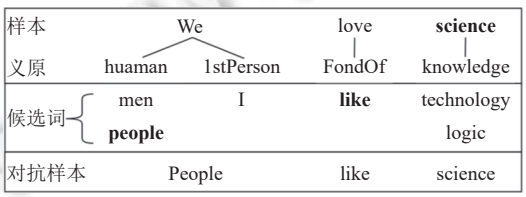

图 1 基于义原的词替换

义原的有效性已经得到了许多验证, 应当尽可能探索其在 NLP 领域能做出的贡献. 本文提出义原级语句稀释法正是义原的一种全新应用, 我们在义原维度构建一个适用于实现语句稀释的义原级词典, 通过所构造的词典完成稀释方法的实际运行, 从而生成更易实现对抗攻击的新数据集.

目前计算机领域对义原研究也已经有了丰富的资源支持. 由 HowNet 开源的 OpenHowNet 义原词库构建了包含 2 000 多个义原的语义描述体系, 并为数十万个汉语和英语单词所代表的含义标注了义原[9,10]. Qi 等人[11]也在近年为这个新词库进行了详细介绍. 刘阳光等人[25]最新的研究提出了针对已标注义原进行一致性检验的方法, 并通过一致性检验对明显有误的数据进行了修复, 进一步提升了义原语料库的数据质量.

3 义原级语句稀释法

本文提出义原级语句稀释法用于强化文本对抗攻击方法能力, 通过在对抗攻击过程中加入稀释过程, 生成更易被成功攻击的稀释样本, 从而提高对抗攻击的成功率. 这种强化方法的概念迁移自机器学习的决策边界理论以及传统软件测试领域中的“边界值分析”法, 机器学习模型与软件有共通性: 位于模型决策边界附近的样本, 更容易导致模型执行分类任务出错. 机器学习领域对模型决策边界也有类似于软件测试中边界的定义, 表示机器学习模型决策空间中会因微小扰动改变输出的分界位置. 郭书杰[26]提出了一种在图像识别领域利用两种不同维度查找法快速寻找模型决策边界的方法.

本文精确定义了用于自然语言处理分类任务的边界表达式、边界距离以及对抗攻击成功率等基本概念, 并根据文本分类场景, 将所定义边界称为分类边界. 基于这些概念提出了适用于自然语言模型的分类边界理论, 并使用经典的自然语言分类模型及数据集, 对现有的多种维度对抗攻击方法在同一数据集不同边界距离子集上的攻击效果进行了横向对比, 验证分类边界理论的有效性.

基于本文定义的分类边界, 进一步设计并实现了基于分类边界理论的义原级稀释池构建算法 DPCA 与义原级语句稀释法 SSDA. DPCA 能够通过预设的种子词集自动搜索义原词典空间, 建立用于稀释文本的稀释池. SSDA 通过 DPCA 建立的稀释池迭代寻找可令样本更靠近模型输入边界的稀释词, 生成不改变原意但更易被攻击的稀释样本, 从而提高对抗攻击成功率.

3.1 分类边界

定义 1. 二分类模型. 令:

$$M(t)=\begin{cases}-1, \text{样本}t\text{为负极性}\\ 1, \text{样本}t\text{为正极性}\end{cases},$$

表示一个二分类模型, 其中 t 代表输入的文本样本, 模型依据其极性分类打上正极性或负极性标签. 用 $M(t)=1$ 简化代表样本 t 的标签为正极性, $M(t)=-1$ 代表样本 t 的标签为负极性. 在分类模型的内部, 通常会有一组用于评估分类模型对输入样本 t 属于某一标签的确信程度的函数, 本文统称其为评估函数. 某一个标签的评估函数值越高, 则模型判断样本属于该标签的可能性也越高. 不同的模型其评估函数实现形式也不同, 例如在 Naïve Bayes 模型中, 这一评估函数表现为后验概率, 取值介于 0–1 之间.

定义 2. 评估函数. 令:

$$\begin{cases}Negative\_Prediction(t)\in[0,1], \text{简写为}NegPre(t)\\ Positive\_Prediction(t)\in[0,1], \text{简写为}PosPre(t)\end{cases},$$

分别表示二分类模型 M 相信 $M(t)=-1$ 与相信 $M(t)=1$ 的评估函数. $NegPre(t)$ 与 $PosPre(t)$ 将返回一个介于 0 至 1 的评估值, 最终 M 判断得到的标签由二者返回值的比较结果决定. 当 $NegPre(t)>PosPre(t)$ 时, 模型给出 $M(t)=-1$ 的标签, 反之模型认为 $M(t)=1$.

定义 3. 边界距离. 令 $B(t)=|PosPre(t)-NegPre(t)|$ 表示样本 t 相较于模型 M 边界的距离, 称 $B(t)$ 为模型 M 的边界距离. $B(t)$ 的含义即模型 M 对最终预测结果的确信程度. 显然, 当 $B(t)=0$ 时, 二分类模型做出的决策是完全随机的, 由此给出分类边界的定义.

定义 4. 分类边界. 当边界距离 $B(t)=0$ 时, 该位置即二分类模型 M 的分类边界. 在该位置, M 将无法准确判断输入样本标签的位置. 如果希望使输入的样本 t 尽可能接近模型的分类边界, 需要做的就是尽可能地减小 $B(t)$ 的值. 下面给出分类边界的例子, 以便读者更清晰理解分类边界的概念.

例 1. 假设 t_s 和 t_w 是两条文本样本. 不失一般性, 令 $B(t_s)\geqslant B(t_w)$. 本例中假设 t_s 为一条长度为 10 的文本样本, 其评估函数 $PosPre(t_s)=0.8$, $NegPre(t_s)=0.2$, 标签为正极性; t_w 为一条长度为 20 的文本样本, 其评估函数 $PosPre(t_s)=0.4$, $NegPre(t_s)=0.6$, 标签为负极性. 在上述定义下 t_w 比 t_s 更靠近分类边界. 图 2 使用一个共用纵坐标轴的合并坐标系展示 t_w 和 t_s 之间的关系, 纵坐标轴处于 $B(t)=0$ 处, 即定义的分类边界, 纵坐标轴左侧为正极性样本坐标系, 右侧为负极性样本坐标系. 可以看到 t_w 较 t_s 更靠近分类边界, 也就更容易受到攻击而逾越边界.

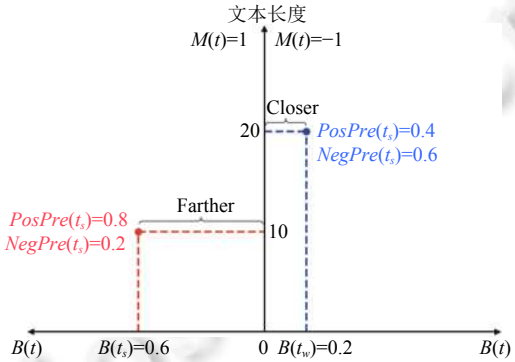

图 2 分类边界

3.2 分类边界理论

基于分类边界的定义, 进一步定义用于强化对抗攻击过程的分类边界理论.

定义 5. 对抗攻击方法. 令 $A(t,t')$ 表示一种对抗攻击方法. $A(t,t')$ 将样本 t 转化为一个对抗样本 t', 根据对抗攻击方法的不同, t' 最终的内容也是未知的.

定义 6. 对抗攻击成功. 假设 $A(t,t')$ 成功将 t 转化为对抗样本 t', 若 $M(t')=-M(t)$ 成立, 可以断言 $A(t,t')$ 成功在样本 t 上攻击了模型 M.

如果将单个样本 t 替换为一个数据集 T, 将对抗攻击方法 A 应用于 T, 那么可以视为所有 T 中的所有样本都生成新的对抗样本数据集 T'. 对整个数据集执行的攻击可以记为 $A(t,t')$.

定义 7. 对抗攻击成功率. 设:

$$T_A'=\{t'|A(t,t'),\ t\in T \text{ and } M(t')=-M(t)\},$$

T_A' 即原始数据集中执行对抗攻击成功的对抗样本构成的子集. 再令:

$$T'=\{t'|A(t,t'),\ t\in T\},$$

T' 表示整个受攻击的原始数据集. 通过成功攻击的对抗样本数据集与原始数据集, 令:

$$E(A(T,T'))=\frac{|T_A'|}{|T'|},$$

$E(A(T,T'))$ 即可表示对抗攻击方法 A 的成功率, 即成功攻击的样本在总样本数中所占的比例.

定义 8. 分类边界理论. 设数据集 $T_s=\{t_s^i\}_{i=1}^n$, $T_w=\{t_w^j\}_{j=1}^n$, 不妨设 T_s 与 T_w 内的元素均按照边界距离 B 的大小从小到大排列. 若 $B(t_s^i)\geqslant B(t_w^j)$, $\forall i\in\{1,2,\ldots,n\}$ and $j=i$ 成立, 那么 $E(A(T_s,T_s'))\leqslant E(A(T_w,T_w'))$ 也将成立.

即若 T_s 中的每个元素均能在 T_w 中不重复地找到一个边界距离更近的元素, 那么使用 T_w 进行对抗攻击将得到比 T_s 更高的成功率.

例 2: 本例选用 NLP 领域常用的 SST-2 数据集对理论进行实验验证. SST-2 数据集中的测试集包含 1 821 条样本, 以 Naïve Bayes 模型作为受害者模型计算这些样本的边界值后, 以 0.65 为边界值分界线, 将该数据集按照边

界值划分为样本数量尽可能均等的两部分. 边界值大于 0.65 合计 885 条样本, 设为数据集 T_s, 边界值小于等于 0.65 合计 936 条样本, 按照分类边界理论, 需要保证对比数据集大小相等, 避免偶然性, 因此随机抽取其中 885 条样本, 设为数据集 T_w, 再令 $S = T_s \cup T_w$, 表示所有实验样本的集合.

图 3 为 S 的边界值 $B(S)$ 分布散点图, 坐标轴定义与例 1 相同. 所有 T_w 中的点用蓝色点表示, T_s 中的点用橙色点表示, 图中 $B(S)=0$ 的位置即为分类边界, 可以看到 T_s 比 T_w 更远离边界. 我们分别选取了语句级的 SCPNs、词语级的 PWWS 和 Genetic、字符级的 VIPER 和 HotFlip、词语级字符级结合的 TextBugger 共 6 种不同维度的对抗攻击方法, 以 T_s 和 T_w 作为输入, 对 Naïve Bayes 模型实施对抗攻击. 对抗攻击成功率 E 的对比结果见表 1, 图 4 对该结果进行了可视化展示.

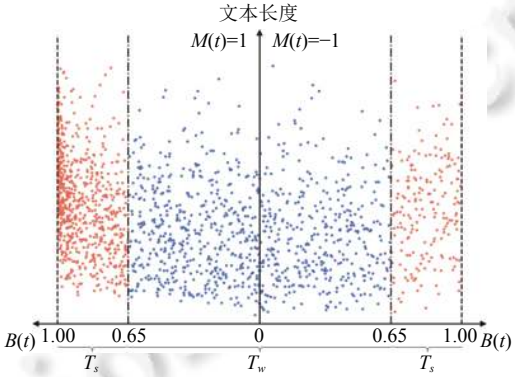

图 3 Naïve Bayes 计算 SST-2 数据集的边界值

表 1 不同分类边界距离的数据集的攻击效果 (%)

输入	SCPNs	PWWS	Genetic	VIPER	HotFlip	TextBugger
T_s	35.03	95.02	61.92	91.07	44.07	94.23
T_w	74.46	98.31	88.36	96.49	84.63	95.93
提升率	**39.43**	**3.29**	**26.44**	**5.42**	**40.56**	**1.7**

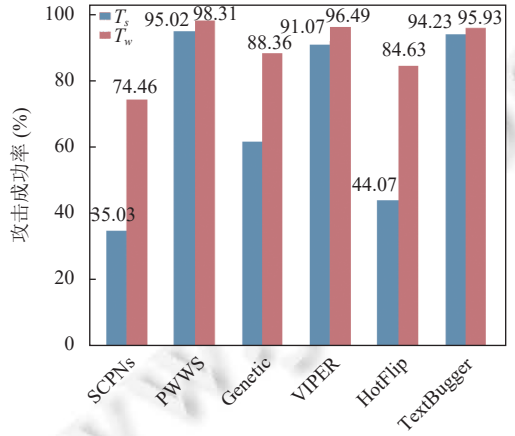

图 4 不同分类边界距离的数据集的攻击效果

实验显示, 6 种对抗攻击方法在 T_w 上展开的攻击成功率均高于 T_s, 即使在 T_s 上已经达到最高成功率 95.02% 的 PWWS 方法, 在 T_w 上仍能取得更高的攻击成功率 98.31%. 而 SCPNs 与 HotFlip 方法上, T_w 的攻击成功率甚至较 T_s 高出 39.43% 与 40.56%, SCPNs 对 T_w 的攻击成功率超过 T_s 的两倍.

这一实验涵盖了语句级、词语级、字符级、词语级字符级结合等多维度对抗攻击方法, 验证了分类边界理论的效果. 如果能在维持样本有效性的前提下, 通过既定方法改变数据集整体的边界距离, 依据统计学理论, 只要数据集的规模足够大, 那么其对抗攻击成功率也会相应提高. 这一迫近边界的过程, 即为稀释过程.

3.3 语句稀释过程

本节设计并实现了用于执行稀释过程的 SSDA 算法, 以及稀释过程所需使用的稀释池构建算法 DPCA. 算法保证原始样本与稀释样本在同一个模型下, 二者的分类标签都将保持一致.

Samanta 等人[18]验证了词语替换和词语增删策略在文本对抗攻击中发挥的有效作用, 其核心思想之一是在高贡献形容词处增删副词. 利用副词操作语句的思想来源于自然语言规律: 副词在语句中大部分时候起强调作用, 而非决定语句含义. 副词能够操纵语句使其远离或迫近分类边界, 又很少直接逾越边界, 因此是实现语句稀释的重要工具. 然而不加区分的选择任意副词不能完全保证对抗样本有效性, 表 2 列举了 3 组语句实例说明副词对语句的影响.

表 2 使用副词操纵语句实例

实例组编号	实例编号	实例内容	实例标签
1	1.1	I feel sad.	Label: –1
	1.2	I never feel sad.	Label: 1
2	2.1	I feel sad.	Label: –1
	2.2	I sometimes feel sad.	Label: –1
3	3.1	I am incredibly sorry.	Label: –1
	3.2	I am sorry.	Label: –1

3 组实例的共同点是同组中两条语句都相差一个副词, 但标签变化情况不同. 第 1 组实例中, “never”蕴含否定含义, 逆转了语句标签, 导致对抗样本不具备有效性, 而第 2 组实例则属于副词的一般作用, 对语句极性增强或弱化, 第 3 组实例表明, 如果副词能够增强或弱化语句极性, 那么将副词删除也会起到同样的效果. 因此, 本文将从副词集合中选择一部分副词构成副词子集, 选择策略为: 用于表达极性大于 0% 的副词集合. 用表 2 中的 1.2 语句为例, 句中“never”表达了极性为 0% 的消极含义, 导致语句标签逆转, 但“sometimes”和“incredibly”表达的极性均大于 0%. 该策略可解释为: 确保使用副词子集中任何副词对语句进行任何增删, 都不会逆转语句标签, 该子集成为构造稀释池的数据源.

稀释池设计为两个独立子池, 分别是增稀释池 (pool for addition, PfA) 与删稀释池 (pool for deletion, PfD). PfA 中包含语句可插入副词, 一般为程度弱化副词, PfD 中包含语句可删除副词, 一般为程度强化副词. 当需要往句子中增加副词时, 从 PfA 中取一个可以弱化语句含义的副词, 当希望删除句子中增强语句的副词时, 通过 PfD 判断哪一些词语可以被删除, 从而达到稀释语句并逼近边界的目的. “稀释池”是一种形象的比喻, 因为稀释过程就像将强极性的文本样本丢进文本构成的清水池中浸泡, 降低该文本样本的极性.

基于现有的义原词典 OpenHowNet 构造稀释池, 本文提出稀释池构建算法 DPCA. DPCA 算法在执行之前将先预置一些种子词语, 由人工挑选少量常见的程度副词构建种子集, 并划分为增稀释池种子集 (seeds for addition, SfA) 与删稀释池种子集 (seeds for deletion, SfD). 接下来在义原关系词库中分别搜索 SfA 与 SfD 的所有同义原词语, 加入对应的稀释池当中, 与种子集取并集后构成完整的稀释池 PfA 与 PfD, 因此 PfA 与 PfD 是强依赖于预置的种子集的. 算法中所使用的义原关系词库即 OpenHowNet, 简写为变量 OHN.

搜索过程的实现利用队列临时存储种子序列, 通过广度优先搜索算法遍历 OHN 词典, 直到最终结果收敛. DPCA 算法自动从 OHN 词典中基于同义原递归搜索了所有种子词语的同义原词, 成功搭建了稀释池. 在实际实现中, 我们也对最终构建的稀释池进行了人工检查, 以避免同义原情况下因词语多义性而加入一些会明显导致语句不通顺的词语. 因此最后搭建的稀释池是结合了自动化算法与人工筛选的, 具有较高的质量. 最终 PfA 与 PfD 分别包括 73 个与 125 个同义原候选词.

算法 1. 稀释池构建算法 (dilution pool construction algorithm).

前置定义:

1. 义原词库 OpenHowNet, 定义为变量 OHN, OHN[w] 回词语 w 的一张同义原词表.
2. 种子队列 Seeds_queue, Seeds_queues.pop() 返回并删除队头元素, Seeds_queues.push() 在队尾插入元素.

算法实现:

```
1. OpenHowNet OHN
2. Seeds_queue SfA, SfD
3. Lists PfA, PfD
4. String TEMP
5. i ← 1
6. function build_pool (Seeds_queue)
7.     result_list
8.     while Seeds_queues is not empty
9.         TEMP = OHN[Seed_queues.pop()]
10.            for w in TEMP
11.                if w is an adverb and w is not in Seed_queues and w is not in result_list
12.                    Add w into result_list
13.                else
14.                    Pass
15.        return result_list
16.    PfA = build_pool(SfA) ∪ SfA
17.    PfD = build_pool(SfD) ∪ SfD
```

基于 PfA 与 PfD 实现稀释算法 SSDA, SSDA 利用稀释池中的词语对源数据进行稀释, 保证:

(1) 原始数据集经稀释后边界距离减小.

(2) 维持原样本标签不变.

算法 2. 义原级语句稀释法 (sememe-level sentence dilution algorithm).

前置定义:

1. 文本数据集 T, $T=\{t_i|0<i<n\}$, 其中 $t_i=w_0w_1\ldots w_m$ $(1\leqslant m)$, w_j 表示一个单词.
2. 二分类模型 M, $M(t_i)$ 表示模型 M 对样本 t_i 的分类标签, 用 t_i' 暂存对 t_i 的修改.
3. 模型 M 下的边界距离 $B(t_i)$, 表示样本 t_i 到模型 M 的边界距离.
4. 构建完成的稀释池 PfA 与 PfD.

算法实现:

```
1. Find B(t_i) for each example t_i ∈ T
2. Tokenize each word w_j ∈ t_i and get its characteristic
3. i ← 1
4. for each t_i ∈ T
5.     j ← 1
6.     for each w_j ∈ t_i
7.         if w_j is an Adverb and w_j is belong to PfD
```

$t_i' \leftarrow t_i - w_j$
if $B(t_i) > B(t_i')$ and $M(t_i) = M(t_i')$
$t_i \leftarrow t_i'$
Continue
else if w_j is an Adjective
for each Adverb in PfA
$t_i' \leftarrow t_i + \{\text{Adverb inserted in front of } w_j\}$
Only record the t_i' with the minimum $B(t_i')$ and $M(t_i) = M(t_i')$
$t_i \leftarrow t_i'$
$j \leftarrow j+1$
$i \leftarrow i+1$

SSDA 在允许语句被稀释前额外增加了 $M(t_i) = M(t_i')$ 的关系判断, 这是区别义原级语句稀释法并非简单的对语句进行二次攻击的关键点. 稀释过程期望尽可能地迫使语句靠近模型分类边界, 但不能逾越边界, 否则将由于样本分类标签被提前改变, 导致后续对抗攻击再次将标签逆转, 将使生成的对抗样本无效. 经典对抗攻击方法, 目的是直接改变分类标签, 即使得 $M(t_i) = -M(t_i')$, 与稀释过程有本质不同.

4 文本对抗攻击能力强化

本节介绍稀释过程如何嵌入经典文本对抗攻击过程中, 实现对对抗攻击过程的改进. 本文设置了由 2 种不同文本数据集以及 2 种不同自然语言机器学习模型构成的合计 4 组交叉对比实验, 每组实验包含对 6 种已发表的主流对抗攻击方法成功率提升效果的验证, 以及使用 2 种不同稀释池时稀释前后对抗样本质量的对比, 即每组实验包含 3 个对照组, 总实验次数合计 12 次. 实验结果显示, 稀释过程有效提高了对抗攻击方法的成功率, 并且稀释后产生的对抗样本在多项文本质量指标上都取得了较原对抗样本更优的表现.

4.1 文本对抗攻击过程改进

本文将稀释池、稀释方法以及算法运行过程中的输入输出数据嵌入进传统的对抗攻击过程中, 形成一条新的对抗攻击工作流, 实现对经典对抗攻击过程的改进. 图 5 展示了经典对抗攻击过程, 图 6 展示了语句稀释过程嵌入传统对抗攻击过程形成的新流程.

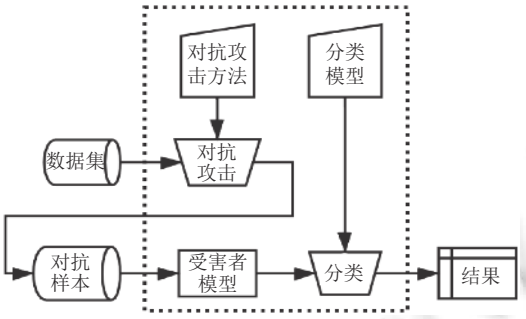

图 5 经典对抗攻击过程

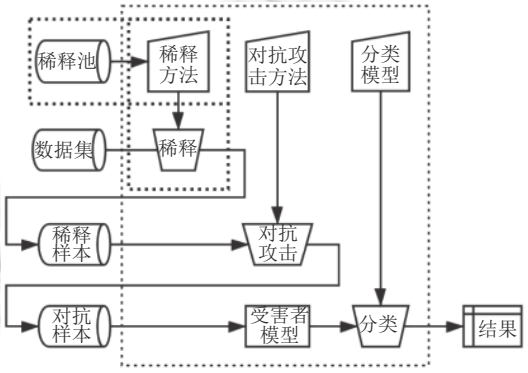

图 6 应用 SSDA 的新对抗攻击过程

在新的对抗攻击过程中, DPCA 将通过预置种子集构建稀释池, SSDA 通过稀释池对原始数据集进行稀释, 生成新的稀释样本, 由稀释样本替代原始数据集作为对抗攻击方法的输入, 产生新对抗样本, 验证在新对抗样本下对受害者模型的对抗攻击成功率以及对抗样本文本质量.

为便于理解整个过程中数据的变化过程, 表 3 展示了在新的对抗攻击过程下, SST-2 数据集中的一条样本经 SSDA 完整稀释后输入对抗样本生成方法 SCPNs 得到对抗样本的过程.

表 3 文本样本输入新对抗攻击过程

攻击阶段	样本变化阶段	文本样本
稀释过程	原始样本	This extremely unfunny film clocks in at 80 minutes, but feels twice as long.
	PfD稀释	This ~~extremely~~ unfunny film clocks in at 80 minutes, but feels twice as long.
	PfA稀释	This really unfunny film clocks in at 80 minutes, but feels twice as long.
对抗样本生成	SCPNs	I have a really bad movie in 80 minutes.

该过程对经典过程的改进体现在:

(1) 稀释样本相较原始样本更接近分类边界, 根据分类边界理论, 将更容易被成功攻击, 从而提升对抗攻击成功率.

(2) 稀释过程不依赖于下游具体使用了哪种对抗攻击方法或是受害者模型, 也不依赖于上游输入的数据集, 由此实现了稀释过程与上下游模块的解耦, 易于快速嵌入.

4.2 文本对抗攻击实验

为了体现 SSDA 生成的稀释样本能有效强化对抗攻击能力, 我们选用了在二分类任务中常用的 SST-2 与 IMDB 数据集作为基准数据集. 选用经典的 Naïve Bayes 模型以及常用于处理情感分析任务的 Vader[27]共同作为受害者模型进行对照实验, 其中 Vader 模型由 NLTK 开源库实现. 选择了 6 种不同维度的主流对抗攻击方法作为对比方法, 验证 SSDA 能够对对抗攻击方法起到的强化作用. 实验中所有对抗攻击方法基于 OpenAttack 实现. 实验过程中为更全面的评估不同方法的对抗攻击能力, 我们采用了多项指标进行综合评估, 所涉及指标定义如下.

定义 9. 词修改率 (word modified rate, WMR).

词修改率 WMR 在词语级统计两个句子不一致的比率. 当两个句子在同一位置处词语不一致, 即视为一处修改, 包括不相等或其中一段文本词语为空字符串而另一段文本非空的情况. 总修改数与最短句的长度之比即为 WMR. 因此, WMR 值越低, 表明两句话之间被修改的词语越少, 在词语维度越相似. 本文中 WMR 的比较对象是输出的对抗样本与原始样本.

定义 10. 语法错误 (grammatical errors, GE).

本实验所有输入均为英文, 语法错误 GE 即句子中存在的英文语法错误数. 语法错误规则设计及实现基于 LanguageTool 的开源版本, 涵盖所有基本英文语法要求. 因此, GE 值越低, 表明对抗样本引入的错误语法数越少.

定义 11. 编辑距离 (edit distance, ED).

编辑距离 ED 即 Levenshtein 距离[28], 是指将一段源文本转化为目标文本的最少编辑操作次数, 其中编辑操作包括: 用一个字符替换另一个字符、增加一个字符及删除一个字符. 对抗样本的 ED 值越低, 说明与原输入样本在字符维度越相似. ED 与 WMR 都用于体现语句相似性, 但二者观察相似性的维度不同. 本文中 ED 的比较对象是输出的对抗样本与原始样本.

定义 12. 困惑度 (perplexity, *PPL*).

假设由 n 个词组成的语句 $S=w_1,w_2,\ldots,w_n$, 其中 w_i $(1\leqslant i\leqslant n)$ 表示一个单词. S 的困惑度 *PPL* 可以表示为:

$$PPL(S)=P(w_1,w_2,\ldots,w_n)^{-\frac{1}{n}},$$

其中, $P(w_1,w_2,\ldots,w_n)$ 表示 S 在给定语言模型下生成的概率, 公式中的几何平均正则化处理是为了避免长句在概率连乘的过程中越来越小, 导致长短句计算结果不公平. 因此, 若句子在给定语言模型下生成的概率越高, 表明句子模式越易被理解, 从而 *PPL* 值将更低.

以上 4 项指标, 结合定义 7 对抗攻击成功率 E, 可以综合评价不同方法的攻击能力. 具体而言, 若 E 提升, 表明新方法取得更高的对抗攻击成功率, 而 WMR、GE、ED 及 *PPL* 下降表明新方法产生的对抗样本在对应指标上取得更好的表现. 表 4 展示了 SSDA 对 6 种现有对抗攻击方法进行强化, 在 SST-2 与 IMDB 数据集下对 Vader 实施对抗攻击后各项指标的平均值. 表 5 提供了完整实验数据. 每组实验按所使用的稀释池划分为 3 个对照组, 不同对

照组所使用稀释池分别如下: (1) 不添加稀释过程的原始数据集进行对抗攻击 (未稀释). (2) 仅使用种子集 SfA 与 SfD 稀释原始数据集进行对抗攻击 (种子稀释). (3) 使用 DPCA 建立的完整稀释池稀释原始数据集进行对抗攻击 (DPCA 稀释).

表 4 SSDA 作用于 Vader 模型

指标	SST-2				IMDB			
	未稀释	种子稀释	DPCA稀释	提升率 (%)	未稀释	种子稀释	DPCA稀释	提升率 (%)
WMR (%)	61.88	57.94	56.53	**−5.35**	80.98	77.07	73.12	**−7.87**
GE	6.07	6.22	6.16	**+1.51**	14.43	13.68	13.28	**−8.00**
ED	8.61	8.61	8.39	**−2.55**	26.28	24.89	23.61	**−10.16**
PPL	586.01	620.87	517.17	**−11.75**	281.26	323.63	249.99	**−11.12**
E (%)	52.91	58.21	66.08	**+13.17**	50.51	54.99	65.32	**+14.81**

表 5 SSDA 作用于 Vader 模型完整数据

对抗样本生成方法	指标	SST-2			IMDB		
		未稀释	种子稀释	DPCA稀释	未稀释	种子稀释	DPCA稀释
SCPNs	WMR (%)	143.62	135.31	133.55	268.81	270.07	256.50
	GE	3.21	3.42	3.46	4.2	4.17	4.21
	ED	13.07	14.17	14.06	56.98	57.42	56.65
	PPL	768.56	495.66	473.50	182.71	190.16	177.68
	E (%)	47.28	50.91	55.68	61.29	61.50	69.44
PWWS	WMR (%)	16.09	14.97	13.29	9.35	8.36	6.40
	GE	2.96	3.13	3.04	4.19	4.33	4.08
	ED	2.78	2.71	2.38	4.00	3.66	2.61
	PPL	560.55	680.84	541.42	308.40	309.39	250.31
	E (%)	78.69	84.07	85.39	45.73	45.79	46.06
TextBugger	WMR (%)	112.79	99.93	100.07	113.64	92.75	89.34
	GE	8.32	7.53	7.47	15.66	13.08	12.03
	ED	14.67	12.90	12.84	31.85	24.56	21.96
	PPL	918.78	938.78	898.73	421.91	405.10	369.14
	E (%)	52.22	56.12	62.05	30.51	38.93	46.71
Genetic	WMR (%)	12.59	11.23	11.12	4.14	3.89	3.62
	GE	2.83	3.05	2.87	5.49	5.31	5.42
	ED	2.18	2.15	2.12	2.55	2.40	2.27
	PPL	393.65	473.41	470.16	185.34	222.37	191.21
	E (%)	38.50	45.09	63.65	34.50	44.11	62.85
HotFlip	WMR (%)	24.43	23.98	20.11	20.84	18.72	14.81
	GE	2.88	3.0	2.94	6.01	5.91	5.75
	ED	4.75	4.90	4.18	13.98	12.56	9.98
	PPL	797.72	864.42	718.60	455.96	684.85	380.69
	E (%)	52.44	64.74	70.84	47.57	55.78	70.09
VIPER	WMR (%)	61.79	62.19	61.01	69.12	68.63	68.03
	GE	16.2	17.17	17.1777	51.05	49.26	48.18
	ED	14.22	14.85	14.77	48.29	48.72	48.17
	PPL	285.13	272.08	279.54	133.22	129.93	130.89
	E (%)	48.05	48.33	58.10	83.48	83.82	96.76

表 4 实验结果从两方面体现了 DPCA 与 SSDA 在提升对抗攻击成功率上的有效性.

首先, 无论是仅使用种子集作为稀释池, 还是基于 DPCA 构造的完整稀释池, SSDA 均能够有效提高对抗攻击成功率, 基于种子集稀释池进行稀释, 在 SST-2 数据集上对 Vader 进行对抗攻击, 成功率的提升率达到 5.3%, 基于 DPCA 构造的稀释池则达到了 13.17%, 而在 IMDB 数据集上前者达到 4.48%, 后者达到 14.81%, 4 组实验再次验

证了分类边界理论及 SSDA 的有效性.

第二, 使用 DPCA 构建的稀释池进行稀释过程, 能够使成功率进一步提升, 提升率在 SST-2 数据集上增长约 2.5 倍, 而在 IMDB 数据集上增长约 3.3 倍, 说明 DPCA 所构造的稀释池对 SSDA 算法起了很大增益作用, DPCA 能够有效帮助 SSDA 算法在更大的空间中搜索候选稀释池, 从而生成更多样化的对抗样本.

表 6 展示了对 Naïve Bayes 实施对抗攻击后各项指标的平均值, 所有实验方法、实验过程与表 4 相同. 表 7 提供了完整实验数据.

表 6　SSDA 作用于 Naïve Bayes 模型

指标均值	SST-2				IMDB			
	未稀释	种子稀释	DPCA稀释	提升率 (%)	未稀释	种子稀释	DPCA稀释	提升率 (%)
WMR (%)	56.77	54.10	52.18	**−4.59**	78.53	76.55	73.55	**−4.98**
GE	5.56	5.62	5.54	**−0.24**	13.35	13.02	12.63	**−5.34**
ED	7.53	7.37	7.07	**−6.15**	22.49	21.98	21.29	**−5.35**
PPL	619.23	742.78	627.30	**+1.30**	260.73	281.94	267.26	**+2.50**
E (%)	79.83	83.77	87.22	**+7.39**	69.13	71.69	75.11	**+5.98**

表 7　SSDA 作用于 Naïve Bayes 模型完整数据

对抗样本生成方法	指标	SST-2			IMDB		
		未稀释	种子稀释	DPCA稀释	未稀释	种子稀释	DPCA稀释
SCPNs	WMR (%)	142.59	145.58	138.62	304.95	309.08	299.36
	GE	3.04	3.09	3.18	4.12	4.10	4.05
	ED	12.52	13.83	13.09	55.84	58.37	57.14
	PPL	547.93	708.02	635.02	185.06	177.17	237.47
	E (%)	55.52	58.08	63.43	65.33	65.39	67.87
PWWS	WMR (%)	15.34	13.85	13.25	6.54	5.77	5.28
	GE	2.77	2.88	2.81	4.05	4.29	3.99
	ED	2.57	2.38	2.26	3.16	2.88	2.65
	PPL	542.97	743.61	600.27	328.47	386.41	346.52
	E (%)	96.49	97.52	98.57	47.95	47.75	48.00
TextBugger	WMR (%)	84.49	72.07	68.99	70.00	59.16	53.03
	GE	6.79	6.09	5.62	10.23	9.40	8.61
	ED	10.49	8.88	8.04	15.84	12.66	11.02
	PPL	1 155.50	1 266.15	1 079.40	381.51	388.31	332.84
	E (%)	93.41	93.88	94.01	70.46	75.23	80.18
Genetic	WMR (%)	16.02	13.62	13.10	5.67	4.78	4.46
	GE	2.83	2.88	2.84	5.74	5.93	5.50
	ED	2.78	2.39	2.35	3.47	2.98	2.80
	PPL	513.07	635.09	519.31	195.88	242.88	222.94
	E (%)	75.40	82.02	87.97	74.03	77.83	83.10
HotFlip	WMR (%)	20.11	16.40	15.81	14.41	11.04	9.85
	GE	2.72	2.84	2.79	5.94	6.17	5.69
	ED	3.55	3.00	2.91	9.39	7.25	6.49
	PPL	653.48	800.34	636.18	340.12	366.72	331.50
	E (%)	64.85	73.91	82.04	57.99	64.82	72.35
VIPER	WMR (%)	62.09	63.10	63.34	69.61	69.44	69.32
	GE	15.20	15.95	16.02	50.01	48.21	47.96
	ED	13.28	13.73	13.77	47.26	47.73	47.64
	PPL	302.45	303.45	293.63	133.35	130.12	132.28
	E (%)	93.30	97.19	97.31	99.03	99.14	99.14

表 6 的实验结果显示, Naïve Bayes 的对抗攻击实验结果与 Vader 的实验结果一致, 在 SST-2 数据集上, SSDA 基于种子集稀释平均提升成功率 3.94%, 基于 DPCA 构造的稀释池进行稀释平均提高成功率 7.39%, 在 IMDB 数据集上前者平均提高 2.56, 后者平均提高 5.975%. DPCA 所构造的稀释池对提升率的增益在 SST-2 数据集下达到 1.87 倍, 在 IMDB 数据集下达到 2.33 倍.

成功率 E 提升表明 DPCA 与 SSDA 的有效性. 进一步, 我们对其余的文本质量评价指标进行分析, 以验证稀释过程是否能够在提升成功率的同时保障对抗样本质量. 图 7–图 10 分别是 WMR、GE、ED 及 *PPL* 这 4 项指标在 4 组实验中平均值的变化趋势. 尽管所有 DPCA 稀释后产生的对抗样本在 4 项指标上都表现出更优或接近原对抗样本的性能, 但加入种子稀释的结果形成更丰富对比后, 我们能从中发现更多信息.

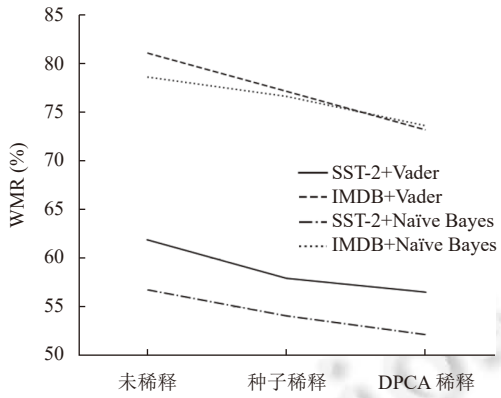

图 7 不同稀释池产生的 WMR 结果

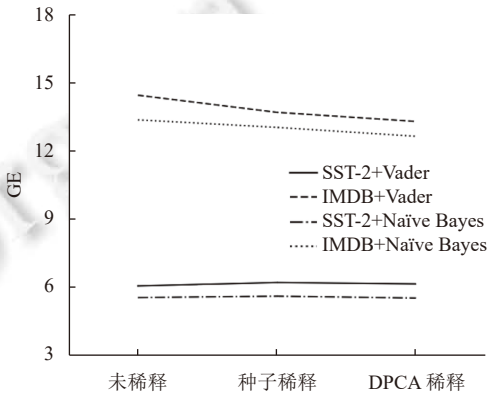

图 8 不同稀释池产生的 GE 结果

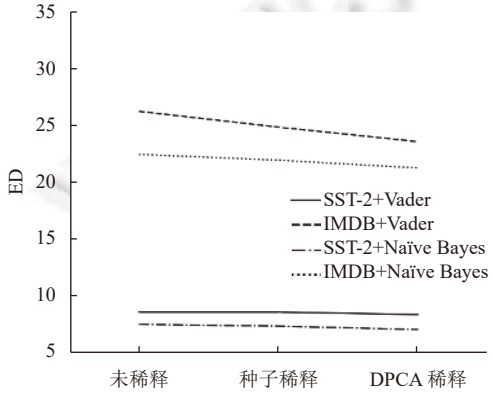

图 9 不同稀释池产生的 ED 结果

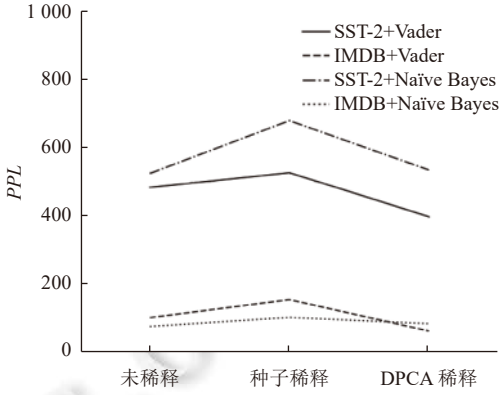

图 10 不同稀释池产生的 *PPL* 结果

从图 7 中词修改率 WMR 的结果分析, 所有实验组下, 从未稀释到 DPCA 稀释的过程中 WMR 都呈现逐步降低的趋势. 这一现象说明, 尽管稀释过程在对抗攻击正式开始前就向输入样本引入了稀释词, 但结果并没有提高最终词修改率, 这是因为稀释过程使得对抗样本生成策略可以减少对原语句的修改而完成对抗攻击, 从而稀释池构建越全面, 效果越好, WMR 也就越低.

从图 8 中语法错误 GE 的结果分析, 在以 IMDB 作为数据集的实验组中, 随着稀释池完整, GE 呈下降趋势, 而以 SST-2 作为数据集的实验组中则呈略微上升后再略微下降的趋势, 整体呈现稳定. SST-2 数据集与 IMDB 的主要区别在于前者为短文本数据集, 后者为长文本数据集, 因此稀释后再攻击的扰动成分比例在 SST-2 数据集中更大. 当仅使用种子集进行稀释时, 可搜索的稀释池空间有限, 不能保证找到最优稀释词, 也就不能保证必然降低后续对抗样本生成过程中引入的语法错误. 这一缺陷能够随着稀释池的完善而缓解, GE 整体呈现轻微的先升后减的

趋势, 但在 SST-2 的实验组中 GE 最终的均值还是略微提升.

从图 9 中编辑距离 ED 的结果分析, 以 IMDB 作为数据集的实验组中 ED 下降的趋势较为明显, 而以 SST 为数据集的实验组中仅呈略微下降趋势. ED 与 WMR 的结果趋势保持一致, 因为二者本身只是相似性在不同文本维度上的体现. 对比 WMR 与 ED 的结果, 我们可以发现稀释过程在降低词维度的修改上更加有效, 而在字符维度则相对稳定.

从图 10 中困惑度 *PPL* 的结果分析. *PPL* 与其他评价指标的显著不同在于尽管其整体趋势呈下降, 但我们在以种子集作为稀释池时, *PPL* 都呈现出明显的上升趋势. 这一现象表明不充分的稀释池将导致对抗样本生成过程中引入更多不确定性. 根据定义 12, *PPL* 表现为给定语言模型在生成该样本时的不确信程度. 与 GE 类似, 当找到的稀释词非最优解时, 不能有效减少后续对抗样本生成策略的修改量. 语句模型在计算生成概率时还需要额外考虑稀释词上下文的生成概率, 造成的不确定性更大, 从而导致 *PPL* 上升. 在 DPCA 构建了更完整的稀释池后, *PPL* 的抖动也得到了有效缓解.

以上对照实验可以说明, SSDA 生成的稀释样本, 在多种对抗攻击方法下都更容易生成能够成功攻击受害者模型的对抗样本, 显著提高了对抗攻击成功率, 表现了 SSDA 不依赖于具体对抗样本生成策略的高度解耦能力与出色的对抗攻击强化能力. 从文本评价指标分析, 稀释样本生成的对抗样本在 4 组对照实验中, 4 项评价指标均有至少 3 项相较于原对抗样本拥有更优值, 呈现出更优的文本质量. 进一步, 通过不同规模稀释池对文本质量以及对抗攻击成功率的对比实验, 我们还验证了 DPCA 在稀释过程中的积极作用, 仅仅使用低规模种子集, 并不能保证有效覆盖整个义原空间可达的词语, 从而可能遗漏最优解. 实验结果说明 DPCA 通过义原空间扩展种子集至完整的稀释池, 能够帮助 SSDA 在提升成功率的同时有效提升文本质量.

整个实验成功搭建了通过 DPCA 构建稀释池, 基于 DPCA 稀释池进行 SSDA 稀释过程, 最后进行对抗样本生成的全新对抗攻击解决方案, 显著提升经典对抗攻击方法的攻击强度以及生成对抗样本的文本质量, 圆满达成了本文提出的“高成功率”且“高质量”的文本对抗攻击目标.

5 结 论

本文基于机器学习领域的决策边界理论与传统软件测试领域的“边界值分析”法提出了分类边界理论, 对分类边界理论开展的实验结果显示, 逼近模型分类边界的样本更容易被成功攻击. 在此理论基础上, 我们进一步实现了改变样本边界距离的有效方法: 义原级语句稀释法 SSDA 与稀释池构建算法 DPCA. 通过对 6 种经典的对抗攻击方法与数据集的交叉对比实验, 结合多项文本质量评价指标, 我们验证了义原级语句稀释法 SSDA 能够有效提升对抗攻击成功率, 并输出高质量的文本对抗样本. 而对多规模稀释池稀释效果的对比也验证了稀释池构建算法 DPCA 则能够构建可用性更高的稀释池, 帮助 SSDA 最大化其稀释效果.

这些成功的实验或许可以给对抗攻击方法的研究一些启迪: 过去改进对抗攻击的思路着重于花费大量精力设计各式各样复杂的对抗样本生成策略, 植入稀释过程为研究者们提供了从过程的维度考虑的新思路. 此外, 是否有更多间接方式能够结合现有方法, 或是使用其他有效的组合攻击手段, 在保证对抗样本质量的情况下提升对抗攻击成功率, 仍有很大的研究空间.

最后, 本文所提出的方法也有持续进步的空间. 例如, 实验结果通过对比在不同规模的稀释池下进行实验的指标提升率, 验证了基于 DPCA 搭建的稀释池对 SSDA 的稀释效果有积极的影响. 随着新兴的义原相关研究发展, 我们的 DPCA 也能持续补充更多易于实现稀释的种子词语以及义原字典, 建立更加完备的稀释池. 文本对抗攻击最终的目的是提升自然语言处理任务的机器学习模型鲁棒性, 因此, SSDA 同样可以在结合更丰富的人类语言学知识的基础上继续改进, 加入更多符合人类语言规律的规则, 将样本稀释为更加符合人类真实表达的稀释样本, 进一步提升样本质量. 我们期待可以有更多学者参与这方面的研究.

References:

[1] Ji SL, Du TY, Li JF, Shen C, Li B. Security and privacy of machine learning models: A survey. Ruan Jian Xue Bao/Journal of Software,

2021, 32(1): 41–67 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6131.htm [doi: 10.13328/i.cnki.jos.006131]

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

[3] Jia RB, Liang P. Adversarial examples for evaluating reading comprehension systems. arXiv:1707.07328, 2017.

[4] Zhao ZL, Dua D, Singh S. Generating natural adversarial examples. arXiv:1710.11342, 2018.

[5] Iyyer M, Wieting J, Gimpel K, Zettlemoyer L. Adversarial example generation with syntactically controlled paraphrase networks. In: Proc. of the 2018 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Vol. 1 (Long Papers). New Orleans: Association for Computational Linguistics, 2018. 1875–1885. [doi: 10.18653/v1/N18-1170]

[6] Eger S, Şahin GG, Rücklé A, Lee JU, Schulz C, Mesgar M, Swarnkar K, Simpson E, Gurevych I. Text processing like humans do: Visually attacking and shielding NLP systems. 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. 1634–1647. [doi: 10.18653/v1/N19-1165]

[7] 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. Online: Association for Computational Linguistics, 2020. 6066–6080. [doi: 10.18653/v1/2020.acl-main.540]

[8] Bloomfield L. A set of postulates for the science of language. Language, Linguistic Society of America, 1926, 2(3): 153–164. [doi: 10.2307/408741]

[9] Dong ZD, Dong Q. HowNet and the Computation of Meaning. Singapore: World Scientific, 2006: 303. [doi: 10.1142/5935]

[10] Dong ZD, Dong Q. HowNet—A hybrid language and knowledge resource. In: Proc. of Int'l Conf. on Natural Language Processing and Knowledge Engineering. Beijing: IEEE, 2003. 820–824. [doi: 10.1109/NLPKE.2003.1276017]

[11] Qi FC, Yang CH, Liu ZY, Dong Q, Sun MS, Dong ZD. OpenHowNet: An open sememe-based lexical knowledge base. arXiv:1901.09957, 2019.

[12] Ebrahimi J, Rao AY, Lowd D, Dou DJ. HotFlip: White-box adversarial examples for text classification. arXiv:1712.06751, 2018.

[13] Pruthi D, Dhingra B, Lipton ZC. Combating adversarial misspellings with robust word recognition. arXiv:1905.11268, 2019.

[14] Li JF, Ji SL, Du TY, Li B, Wang T. TextBugger: Generating adversarial text against real-world applications. arXiv:1812.05271, 2018.

[15] Li JF. Research on adversarial attack and defense against natural language processing system [MS. Thesis]. Hangzhou: Zhejiang University, 2020 (in Chinese with English abstract).

[16] 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]

[17] Alzantot M, Sharma Y, Elgohary A, Ho BJ, Srivastava M, Chang KW. Generating natural language adversarial examples. arXiv:1804.07998, 2018.

[18] Samanta S, Mehta S. Towards crafting text adversarial samples. arXiv:1707.02812, 2017.

[19] Zeng GY, Qi FC, Zhou QR, Zhang TJ, Ma ZX, Hou BR, Zang Y, Liu ZY, Sun MS. OpenAttack: An open-source textual adversarial attack toolkit. arXiv:2009.09191, 2020.

[20] Niu YL, Xie RB, Liu ZY, Sun MS. Improved word representation learning with sememes. In: Proc. of the 55th Annual Meeting of the Association for Computational Linguistics (Vol. 1: Long Papers). Vancouver: Association for Computational Linguistics, 2017. 2049–2058. [doi: 10.18653/v1/P17-1187]

[21] Fu XH, Liu G, Guo YY, Wang ZQ. Multi-aspect sentiment analysis for Chinese online social reviews based on topic modeling and hownet lexicon. Knowledge-based Systems, 2013, 37: 186–195. [doi: 10.1016/j.knosys.2012.08.003]

[22] Qi FC, Huang JJ, Yang CH, Liu ZY, Chen X, Liu Q, Sun MS. Modeling semantic compositionality with sememe knowledge. arXiv: 1907.04744, 2019.

[23] Qin YJ, Qi FC, Ouyang SC, Liu ZY, Yang C, Wang YS, Liu Q, Sun MS. Improving sequence modeling ability of recurrent neural networks via sememes. IEEE/ACM Trans. on Audio, Speech, and Language Processing, 2020, 28: 2364–2373. [doi: 10.1109/TASLP.2020.3012060]

[24] Zhang L, Qi FC, Liu ZY, Wang YS, Liu Q, Sun MS. Multi-channel reverse dictionary model. arXiv:1912.08441, 2019.

[25] Liu YG, Qi FC, Liu ZY, Sun MS. Research on consistency check of sememe annotations in HowNet. Journal of Chinese Information Processing, 2021, 35(4): 23–34 (in Chinese with English abstract). [doi: 10.3969/j.issn.1003-0077.2021.04.004]

[26] Guo SJ. Black box adversarial examples generation method based on fast boundary attack. Computer Systems & Applications, 2020, 29(12): 216–221 (in Chinese with English abstract). [doi: 10.15888/j.cnki.csa.007684]

[27] Hutto C, Gilbert E. VADER: A parsimonious rule-based model for sentiment analysis of social media text. Proc. of the Int'l AAAI Conf. on Web and Social Media, 2014, 8(1): 216–225.

[28] Levenshtein VI. Binary codes capable of correcting deletions, insertions, and reversals. Doklady Akademii Nauk SSSR, 1965, 163(4): 845–848.

附中文参考文献:

[1] 纪守领, 杜天宇, 李进锋, 沈超, 李博. 机器学习模型安全与隐私研究综述. 软件学报, 2021, 32(1): 41–67. http://www.jos.org.cn/1000-9825/6131.htm [doi: 10.13328/i.cnki.jos.006131]

[15] 李进锋. 面向自然语言处理系统的对抗攻击与防御研究 [硕士学位论文]. 杭州: 浙江大学, 2020.

[25] 刘阳光, 岂凡超, 刘知远, 孙茂松. HowNet义原标注一致性检验方法研究. 中文信息学报, 2021, 35(4): 23–34. [doi: 10.3969/j.issn.1003-0077.2021.04.004]

[26] 郭书杰. 基于快速边界攻击的黑盒对抗样本生成方法. 计算机系统应用, 2020, 29(12): 216–221. [doi: 10.15888/j.cnki.csa.007684]

叶文滔(1997－), 男, 硕士, 主要研究领域为机器学习, 自然语言处理, 文本对抗攻击, 蜕变测试.

陈仪香(1961－), 男, 博士, 教授, CCF 杰出会员, 主要研究领域为物联网与信息物理融合系统, 实时软件系统, 软件形式化方法与可信评估, 软硬件协同设计与优化技术.

张敏(1977－), 女, 博士, 教授, CCF 专业会员, 主要研究领域为复杂系统的量化分析与验证, AI 系统的测试与分析验证.