计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2022,31(11):238−245 [doi: 10.15888/j.cnki.csa.008820]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于改进人工蜂群算法的文本对抗样本生成①

杨　帆, 李邵梅, 金柯君

(战略支援部队信息工程大学, 郑州 450001)
通信作者: 杨　帆, E-mail: le2jh@foxmail.com

摘　要: 文本对抗样本的生成对于研究基于深度学习的自然语言处理系统的脆弱性, 提升这类系统的鲁棒性具有重要的意义. 本文对词级对抗样本生成中的重要步骤, 替换词的搜索展开研究, 针对现有算法存在的早熟收敛和有效性差的问题, 提出了基于改进人工蜂群搜索算法的文本对抗样本生成方法. 首先, 根据知网 HowNet 库中单词的义原标注筛选得到拟被替换词的搜索空间; 然后, 基于改进的人工蜂群算法搜索并定位替换词生成高质量的文本对抗样本. 本文针对当前主流的基于深度神经网络的文本分类模型, 在两个文本分类数据集上进行了攻击测试. 结果表明, 跟已有文本对抗样本生成方法相比, 本文提出的方法能以较高的攻击成功率误导文本分类系统, 并更多地保留语义和语法的正确性.

关键词: 文本分类; 对抗样本; 人工蜂群算法; 义原

引用格式: 杨帆,李邵梅,金柯君.基于改进人工蜂群算法的文本对抗样本生成.计算机系统应用,2022,31(11):238–245. http://www.c-s-a.org.cn/1003-3254/8820.html

Text Adversarial Samples Generation Based on Improved Artificial Bee Colony Algorithm

YANG Fan, LI Shao-Mei, JIN Ke-Jun

(PLA Strategic Support Force Information Engineering University, Zhengzhou 450001, China)

Abstract: The generation of text adversarial samples is of great significance for studying the vulnerability of deep learning-based natural language processing (NLP) systems and improving the robustness of such systems. This work studies the important steps in the generation of word-level adversarial samples and the search for replacement words. Considering the problems of premature convergence and poor effectiveness of existing algorithms, a text adversarial sample generation method is proposed, which is based on an improved artificial bee colony (ABC) search algorithm. Firstly, the search space of the words to be replaced is obtained by the screening of the sememe annotations of the words in the HowNet database. Then, the improved ABC algorithm is employed to search and locate the replacement words for the generation of high-quality text adversarial samples. Finally, attack tests are conducted on two text classification datasets for a comparison with the current mainstream text classification models based on deep neural networks (DNNs). The results demonstrate that compared with the existing text adversarial sample generation methods, the proposed method can mislead the text classification system with a higher success rate of attack and preserve semantic and grammatical correctness to a larger extent.

Key words: text classification; adversial examples; artificial bee colony (ABC) algorithm; sememe

1　引言

当前, 基于深度学习的自然语言处理技术在文本分类、情感分析、机器翻译等任务中广泛应用, 并取得了不错的应用效果. 但是, 研究表明[1,2], 当面对恶意

① 基金项目: 国家自然科学基金创新群体项目 (61521003)
收稿时间: 2022-03-04; 修改时间: 2022-04-02; 采用时间: 2022-04-25; csa 在线出版时间: 2022-08-26

构造的对抗性样本时, 基于深度神经网络的自然语言处理模型表现出了极大的脆弱性. 文本对抗样本主要是通过对文本中的字、词或者句子进行替换、插入、移除等加扰操作, 以生成人为不易觉察的攻击性文本, 实现对基于深度神经网络的自然语言处理系统的误导. 例如攻击者通过对评论文本加扰来干扰产品推荐系统; 通过在有害短信中使用变形字以规避检测等. 文本对抗样本的生成研究对于探索基于深度学习的自然语言处理技术的安全盲点, 提升其面对文本对抗攻击的鲁棒性, 具有重要的意义.

根据自然语言处理模型的输入特征, 可以从字符、词和句子 3 个层面对文本进行加扰, 对应地包含 3 类文本对抗样本生成方式: 字符级对抗样本、词级对抗样本和句子级对抗样本. 针对字符级对抗样本生成, Gao 等[3] 提出对重要单词进行字符增删、交换、替换操作, 以实现在付出最小扰动代价的情况下生成不易察觉的对抗样本, 但目前基于语法纠错的防御方法可以很好修正此类攻击; Eger 等[4] 提出选择字符中视觉效果相近的字符进行替换, 以实现人眼的不可察觉性. 句子级对抗样本, 主要是通过增加新的干扰语句或进行复述改写来实现, 例如 Jia 等[5] 提出在样本中插入注意力分散语句来欺骗阅读理解系统; Ribeiro 等[6] 提出语义等价的对抗攻击, 将输入改写为与之语义相同的句子. 但是句子级的扰动往往使得对抗样本和原始输入之间有巨大的差别, 很难控制所产生的对抗样本的质量, 所以无法保证其有效性.

已有研究表明, 在样本质量和攻击成功率方面, 词级对抗样本都是最优选择, 为此, 围绕词级对抗样本生成的研究也最多. Li 等[7] 提出使用原始的 BERT 模型来制作对抗性样本, 可以很好地保存原来语义; Ren 等[8] 提出了基于词显著性和分类概率共同决定的同义词替换攻击方法——PWWS 算法, 首先通过计算被攻击模型分类的分数, 定位输入中最易受攻击的单词, 接下来贪婪搜索出最适宜的替换词, 从而取得了很好的攻击效果; Alzantot 等[9] 提出, 词级对抗样本生成本质上是一个组合优化问题, 即在有限同义词表空间内进行搜索, 从而寻找能够使被攻击模型分类错误的最优解. 基于此, 国内外学者提出了基于不同的最优化搜索算法的词级对抗样本生成方法. 文献 [9] 采用遗传算法, 通过选择、交叉和变异操作, 并行搜索最优解. 但是在出现一个局部最优解 (即能够使受害者模型概率发生最大变化的最优同义词) 的情况下, 遗传算法搜索其他方向的概率极低, 容易陷入局部最优的情况, 因此攻击成功率不高. 为了克服这一弱点, Zang 等[10] 提出了基于义原的离散粒子群优化算法, 首先, 根据义原排除无效的单词组合, 得到可替代词的搜索空间, 从而保留了更多且高质量的对抗样本. 然后, 通过调整粒子的速度和位置, 在缩减后的空间中搜寻可以成功攻击受害者模型的对抗样本. 与遗传算法相比, 粒子具有飞跃性的特点使其具有找到全局最优解的能力. 但粒子群算法同时具有对参数的依赖性过大以及容易早熟收敛的缺点, 尤其在词级离散空间内表现更为明显, 因此对抗样本的生成效果也无法达到最佳.

为了进一步研究基于深度学习的自然语言处理系统的脆弱性, 提升这类系统的鲁棒性, 本文提出一种基于改进人工蜂群算法的文本对抗样本生成方法. 与贪婪搜索的方法相比, 收敛速度快; 与粒子群算法相比, 需要的控制参数少. 并且, 该方法不需要进行反向传播算法求解攻击梯度, 从而可以在不知道模型内部结构的情况下实现黑盒攻击.

2 义原及人工蜂群算法介绍

2.1 义原及知网 HowNet

根据文献 [11], 义原是语言学中最小的、不可分割的语义单位, 所有词语的含义可以由义原构成. 基于此, Dong 等[11] 设计和构建了包含 2 000 多个义原的语义描述体系 HowNet 知网库, 并为十几万个汉语和英语词所代表的概念标注了义原. 图 1 以单词苹果 (apple) 为例, 列举了其在知网中完整的义原树结构.

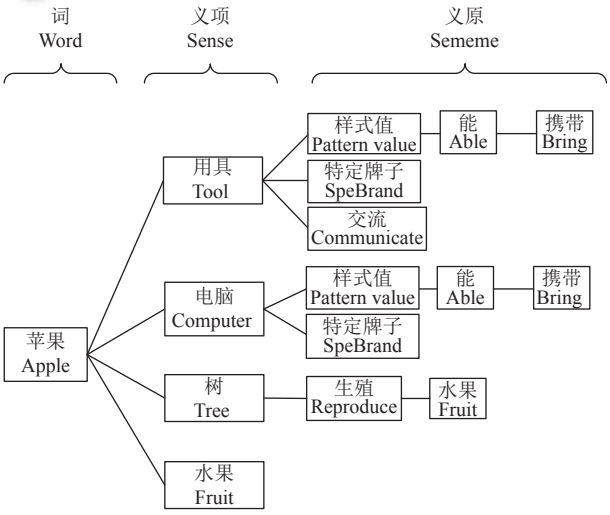

图 1 苹果 (apple) 义原结构示意图

由图 1 可见, 具有不同语义的单词可以用义原来进行组合表示, 保证了每个单词的语义完整性. HowNet 由于具有丰富的调用接口, 可以实现义原查询、词相似度计算等功能, 被广泛应用于各项自然语言处理任务中, 比如基于义原的词表示学习[12]、语言模型[13]、语义合成[14] 等. 本文主要基于 HowNet 中的义原标注搜索同义词, 对文本中的词进行替换实现对抗样本的生成.

2.2 人工蜂群算法

人工蜂群算法是由 Karaboga[15] 提出的一种基于群智能的全局优化算法, 其直观背景来源于蜂群的采蜜行为, 蜜蜂根据各自的分工进行不同的活动, 并实现蜂群信息的共享和交流, 从而找到问题的最优解. 标准的人工蜂群算法将蜜蜂分为 3 类: 雇佣蜂、观察蜂和侦察蜂, 将食物源视为问题的一个候选解, 不同种类的蜜蜂以不同的方式寻找食物源的过程即为搜索最优解的过程. 下面是其搜索步骤.

(1) 初始化阶段. 在搜索空间中随机生成 SN 个解, 种群中第 i 个解表示为$X_i = [X_{i,1}, X_{i,2}, \cdots, X_{i,D}]$, 其中 D 为优化问题的维数.

(2) 雇佣蜂阶段. 每只雇佣蜂在X_i附近进行搜索, 找到更好的解X_i':

$$X'_{id} = X_{id} + \varphi_{id}(X_{id} - X_{kd}) \tag{1}$$

其中, $i = 1,2,\cdots,SN$, $d = 1,2,\cdots,D$, φ_{id}为 [−1, 1] 之间的随机数, $k \neq i$. 贪婪比较新生成的解X_i'和X_i, 如果X_i'的适应度值更高, 则替换X_i.

(3) 观察蜂阶段. 每只观察蜂采用轮盘赌规则选择候选解, 首先计算食物源被选中的概率:

$$P_i = \frac{fit_i}{\sum_{i=1}^{SN} fit_i} \tag{2}$$

其中, fit_i是候选解X_i的适应度值. 然后计算每个食物源的累计概率P_i, 并随机产生 [0, 1] 之间的一个随机数 r, 依次比较累计概率$P_1, P_2 - P_i$与 r 的值, 选中第一个大于 r 的食物源作为候选解. 此时观察蜂转换为雇佣蜂, 对新的候选解附近进行贪婪搜索替换, 直至找到更优解.

(4) 侦察蜂阶段. 当雇佣蜂和观察蜂在指定步数内没有完成新解的替换, 则丢弃原蜜源, 雇佣蜂转换为侦察蜂, 通过以下公式搜索新解:

$$X_{id} = X_d^{\min} + r(X_d^{\max} - X_d^{\min}) \tag{3}$$

其中, r是区间 [0, 1] 上的随机数, $X_d^{\min}$和$X_d^{\max}$是第d维解的下界和上界.

由上述搜索过程可以看出, 人工蜂群算法可以通过雇佣蜂、观察蜂和侦察蜂的转换, 实现全局和局部搜索, 通过侦察蜂来避免陷入局部最优解, 更适用于本文的词级对抗样本生成中的搜索问题.

3 基于人工蜂群算法的文本对抗样本生成

3.1 基于义原标注的词替换方法

文献 [10] 指出, 具有相同义原表示的词具有相同的含义, 可以互为替代. 具体举例如下: 图 1 中的“苹果”一词具有“用具”“电脑”“树”和“水果”4 种语义信息. 此时需应用义原标注来进一步区分. 当单词“苹果”的义项为“用具”时, 在 HowNet 知网库中查找到具有相同义原“能携带”“特定牌子”“交流”的可替换词有“OPPO”“华为”等, 如此便通过义原寻找到了最贴近“苹果”用具语义的替换词. 如果无法查找到完全相同的义原, 单词便没有可替换词语. 对于输入的原始句子, 首先将单词还原为多义原形式, 以此来寻找更多的替换, 之后再将单词由义原形式还原为单词形式来避免语法错误.

为了提高搜索效率, 保证对抗样本的语义语法特征一致, 在本文基于义原标注的词替换方法中, 对替换词的选择进行了限制: 一是需与原词具有相同的词性标签; 二是两个单词在 HowNet 知网库中需要具有相同的义项才可以进行替换.

与基于同义词和词向量的替换方法相比, 义原标注具有以下优点: 一方面, 其精度更高, 替代词更为准确, 而使用同义词或词向量时会不可避免得引入许多不合适的、低质量的替代词, 从而破坏原始输入的语义和语法特征; 另一方面, 由于义原标注的单词类型更全, 可以查找到更多的替代词, 从而保留更多潜在的对抗样本, 扩大了搜索空间.

3.2 基于改进人工蜂群算法的文本对抗样本生成

对于一个句子, 使用基于义原标注的方法找到每个词的替代词后, 接下来应用基于人工蜂群的算法进行对抗样本的搜索. 与原始人工蜂群算法不同, 词对抗样本的整个搜索空间由句子每个词的替代词组成, 是离散空间, 因此需要对人工蜂群算法进行离散化处理.

每个食物源对应为一个句子, 食物源的每个维度对应为一个单词. 食物源的适应度值是目标标签的预测概率, 由被攻击的模型得出, 目标标签即为攻击的期望分类结果. 以二分类任务为例: 如果原始输入的真实

标签为“0”, 则对抗攻击的目标标签就为“1”, 反之亦然.

基于上述改进后的人工蜂群算法, 生成文本对抗样本的主要步骤如下.

(1) 初始化阶段. 由于对抗样本需要尽可能接近原始输入, 因此无法采取随机初始化的方式. 本文借鉴遗传方法[9] 中的变异思想, 对原始输入中的一个或多个单词进行义原词替换, 生成初始 CS 个食物源, 计算其初始适应度值同时进行最优解判断: 如果预测标签等于目标标签, 则直接返回替换词作为成功对抗样本; 如果预测标签不等于目标标签, 则保存适应度高的食物源.

(2) 雇佣蜂阶段. 以初始化阶段中适应度最高的食物源为依托, 在邻域内变异生成 CS 个新的食物源, 计算适应度值再次进行最优解判断, 此时需保存所有的适应度值.

(3) 观察蜂阶段. 为了避免陷入局部最优, 本文不再采取雇佣蜂阶段中的贪婪比较法, 而是采用基于适应度值占比率的轮盘赌方法来生成新的观察蜂. 根据式 (2) 计算步骤 (2) 雇佣蜂阶段中每个食物源被选中的概率和其累计概率, 生成 [0, 1] 之间的随机数, 选中累计概率第一个大于此随机数的食物源作为候选解. 观察蜂在候选解邻域进行贪婪搜索, 直至预测标签等于目标标签.

(4) 如果雇佣蜂和观察蜂阶段都没有搜索到使预测标签发生变化的最优词, 则进入侦察蜂阶段: 重新进行初始化生成新的食物源.

上述算法的描述见算法 1.

算法 1. 基于人工蜂群的词搜索算法

输入: 原始样本, 食物源数量 CS, 最大循环次数 MCN, 食物源耗尽上限 $limit$, 循环计数 $cycle = 0$
输出: 最优食物源 X

```
1) 初始化:
初始化试验次数 n = 0;
对原始样本变异生成 CS 个食物源;
计算每个食物源的适应度值 prob;
最优解判断:
  if 预测标签 = 目标标签:
    输出此时最优解 X
  else:
    保存 prob 最大位置的食物源 X_i
2) 雇佣蜂阶段:
  当 cycle < MCN 时:
    对食物源 X_i 变异生成 CS 个新的食物源;
    计算其适应度值 prob;
    最优解判断:
    if 预测标签 = 目标标签:
      输出此时最优解 X
    else:
      n = n + 1
      保存所有食物源 prob
3) 观察蜂阶段:
      根据轮盘赌规则生成 CS 个新的食物源;
      计算其适应度值 prob;
      最优解判断:
      if 预测标签 = 目标标签:
        输出此时最优解 X
      else:
        保存 prob 最大位置的食物源 X_i
        n = n + 1
4) 侦察蜂阶段:
      当 n > limit 时:
      返回步骤 1) 重新初始化
      cycle = cycle + 1
  end
```

4 实验分析

4.1 数据集

为了充分评估本文所提出的攻击方法, 实验选用 SST-2[16] 和 AG-NEWS 两个文本分类数据集. SST-2 为电影评论数据集, 分为 6 920 条训练样本, 872 条验证样本和 1 821 条测试样本. AG-NEWS 为新闻文本数据集, 包含 4 个类别超过 2 000 个新闻源的新闻文章, 数据集采用了标题和描述字段, 每个类别分别拥有 30 000 个训练样本和 1 900 个测试样本. 两个数据集都是针对单个句子进行文本分类任务, 其中 SST-2 数据集平均句子长度仅为 17 个单词, 而 AG-NEWS 数据集平均句子长度为 68 个单词, 搜索空间增大, 类别数更多, 更具挑战性. 表 1 展示了数据集的具体统计信息.

实验中, 每次随机选取了 1 000 条 SST-2 数据和 500 条 AG-NEWS 数据作为原始样本, 被选中数据在加扰前均可以被分类模型正确预测.

表 1 数据集详细统计信息

数据集	类别	训练集	测试集	验证集	句子平均长度
SST-2	2	6 920	1 821	872	17
AG-NEWS	4	120 000	7 600	0	68

4.2 被攻击模型与对比方法

本文选用文本分类中常用的 Bi-LSTM[17] 和 BERT[18] 作为被攻击模型. 实验中的 Bi-LSTM 模型使用 300 维的 Word2Vec 词向量, 隐层设置为 128 维;

BERT 模型选用 base 版.

本文选用以下词级攻击方法作为基线模型来进行对比:

(1) PWWS[8]: 基于贪婪搜索的同义词替换攻击方法

(2) Genetic[9]: 基于遗传算法的同义词替换攻击方法

(3) PSO[10]: 基于义原的粒子群优化攻击方法

经过实验测试, 本文算法中的食物源数量 *CS*、最大循环次数 *MCN* 和食物源耗尽上限 *limit* 均设为 20.

4.3 文本对抗样本的评测指标

为了验证本文提出的攻击方法的有效性, 需要从不同角度进行性能评估, 包括攻击有效性、攻击效率和对抗样本质量, 选用表 2 所示的详细指标来进行评测.

表 2 评测指标详细信息

分类	评测指标
攻击有效性	攻击成功率
攻击效率	平均攻击耗时
对抗样本质量	平均扰动率 平均语义相似度 平均语法错误 句子困惑度

下面对评测指标进行详细说明:

(1) 攻击成功率 (success rate): 攻击成功率=攻击成功样本数量/总样本数量;

(2) 平均攻击耗时 (running time): 每个样本的平均生成时间, 包括使用义原进行替换词查找的时间和人工蜂群算法的搜索时间;

(3) 平均扰动率 (modified rate): 平均扰动率=替换单词数量/输入总单词数量;

(4) 平均语义相似度 (semantic similarity): 对原样本和生成样本进行句子编码[19] 后计算句向量的余弦相似度;

(5) 平均语法错误 (grammatical errors): 使用 Python 中的 Language-Tool 工具获得每个句子的语法错误数, 在攻击成功的样本中取其平均值;

(6) 句子困惑度 *PPL*: 句子序列$s=w_1,w_2,\cdots,w_N$的困惑度 *PPL* 值计算公式如下:

$$\begin{aligned} PPL(s) &= P(w_1w_2\cdots w_N)^{-\frac{1}{N}} \\ &= \sqrt[N]{\frac{1}{p(w_1w_2\cdots w_N)}} \\ &= \sqrt[N]{\prod_{i=1}^{N}\frac{1}{p(w_i|w_1w_2\cdots w_{i-1})}} \end{aligned} \tag{4}$$

其中, $p(w_i)$表示第 i 个词出现的概率, $p(w_i|w_1w_2\cdots w_{i-1})$表示基于前 $i-1$ 个词计算出第 i 个词的概率. 可以看出, 当 *PPL* 越小, $p(w_i)$则越大, 即本句话中每个词出现的概率越高, 流利度越好. 实验中直接调用 GPT-2 模型[19] 来计算.

4.4 实验结果与分析

4.4.1 义原有效性对比结果

为了证明基于义原标注的词替换方法的有效性, 选用 1 000 条被 BERT 模型正确预测的 SST-2 数据, 分别采用基于义原 HowNet 和基于同义词 WordNet 库的本文方法进行扰动, 使用攻击成功率指标进行评价, 实验结果如表 3 所示.

表 3 义原有效性实验结果

攻击方法	攻击成功率
基于义原HowNet的人工蜂群算法	0.872
基于同义词WordNet的人工蜂群算法	0.638

由实验结果可知, 基于义原 HowNet 的方法攻击成功率远高于同义词替换的方法. 这是因为使用 HowNet 可以寻找到更多更精准的替换词, 搜索空间变大, 增大了搜索到使模型分类错误的对抗样本的成功率.

4.4.2 与其他方法对比实验结果

从表 4 的实验结果可以看出, 本文提出的基于人工蜂群算法除攻击耗时外在其余指标上几乎均取得了最优的性能, 充分证明了其攻击的有效性和对样本质量的提升效果. 下面对各项指标进行具体分析.

(1) 攻击成功率: 如表 4 中攻击成功率一列所示, 本文提出的算法在所有数据集和文本分类模型上均取得了最高的攻击成功率. 此外, 4 种文本对抗样本生成方法在 SST-2 数据集上的攻击成功率均要高于 AG-NEWS 数据集, 一是因为 SST-2 数据集句子平均长度要少于 AG-NEWS, 对少量词进行修改, 其语义变化有可能会非常巨大, 从而使分类器分类错误的概率更高; 二是因为 SST-2 为二分类数据集, 分类器为粗粒度的划分, 精度不高, 对其实现攻击较为容易.

(2) 样本质量: 如表 4 中第 4–7 列所示, 为了更直观展示生成样本质量情况, 将结果绘制如图 2 和图 3 所示. 其中, PSO 和本文均采用的是基于义原标注的词替换方法, 而另外两种为基于同义词库的方法; 图 2 描述的是 AG-NEWS 数据集的样本情况, 图 3 描述的是 SST-2 数据集的样本情况. 下面对长度不同的两类文本数据集进行具体分析.

表 4 数据集实验结果

数据集	攻击方法	攻击成功率	平均扰动率	语义相似度	语法错误	困惑度*PPL*	耗时 (ms)
SST-2 (BiLSTM)	PWWS	0.506	0.288	0.817	4.498	837.78	**8.019**
	Genetic	0.829	0.302	0.795	4.590	914.76	57.524
	PSO	0.773	0.305	0.805	**4.286**	992.82	45.996
	人工蜂群	**0.894**	**0.257**	**0.859**	4.364	**837.17**	89.334
SST-2 (BERT)	PWWS	0.773	0.243	0.850	4.618	818.11	**5.623**
	Genetic	0.848	0.264	0.825	4.613	1 013.8	55.559
	PSO	0.734	0.263	0.821	4.425	1 399.7	57.398
	人工蜂群	**0.872**	**0.211**	**0.871**	**4.419**	**739.96**	91.775
AG-NEWS (BiLSTM)	PWWS	0.604	0.381	0.843	14.718	486.48	**14.030**
	Genetic	0.702	0.405	0.827	14.44	597.85	117
	PSO	0.588	0.299	0.926	**12.691**	353.92	155.11
	人工蜂群	**0.818**	**0.271**	**0.926**	12.784	**282.17**	187.03
AG-NEWS (BERT)	PWWS	0.57	0.450	0.823	14.116	478.24	**19.330**
	Genetic	0.762	0.441	0.813	14.129	587.84	155.96
	PSO	0.517	0.459	0.922	11.647	368.16	240.63
	人工蜂群	**0.792**	**0.340**	**0.928**	**10.349**	**328.1**	234.18

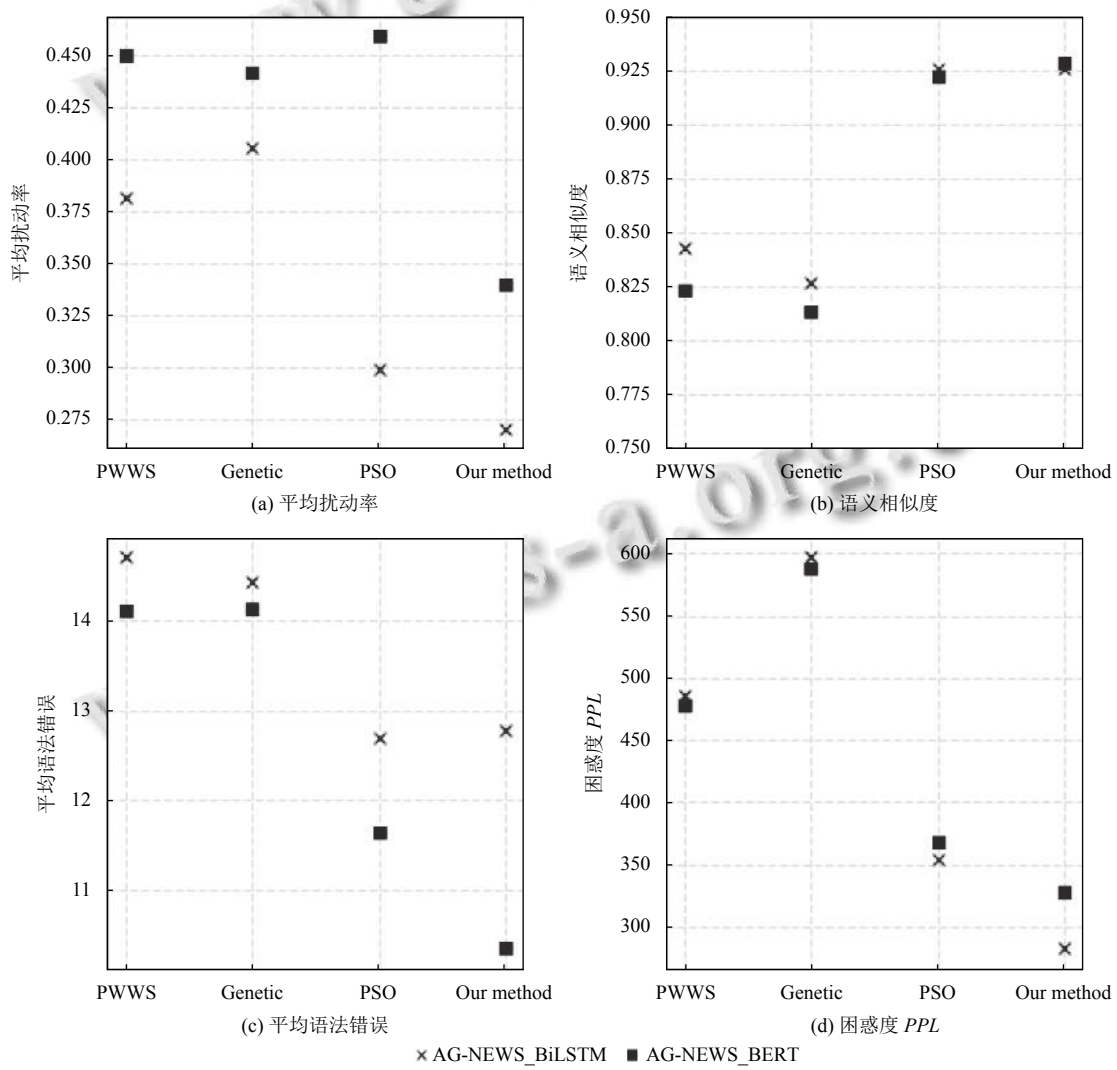

图 2 不同方法在 AG-NEWS 数据集生成的对抗样本质量对比图

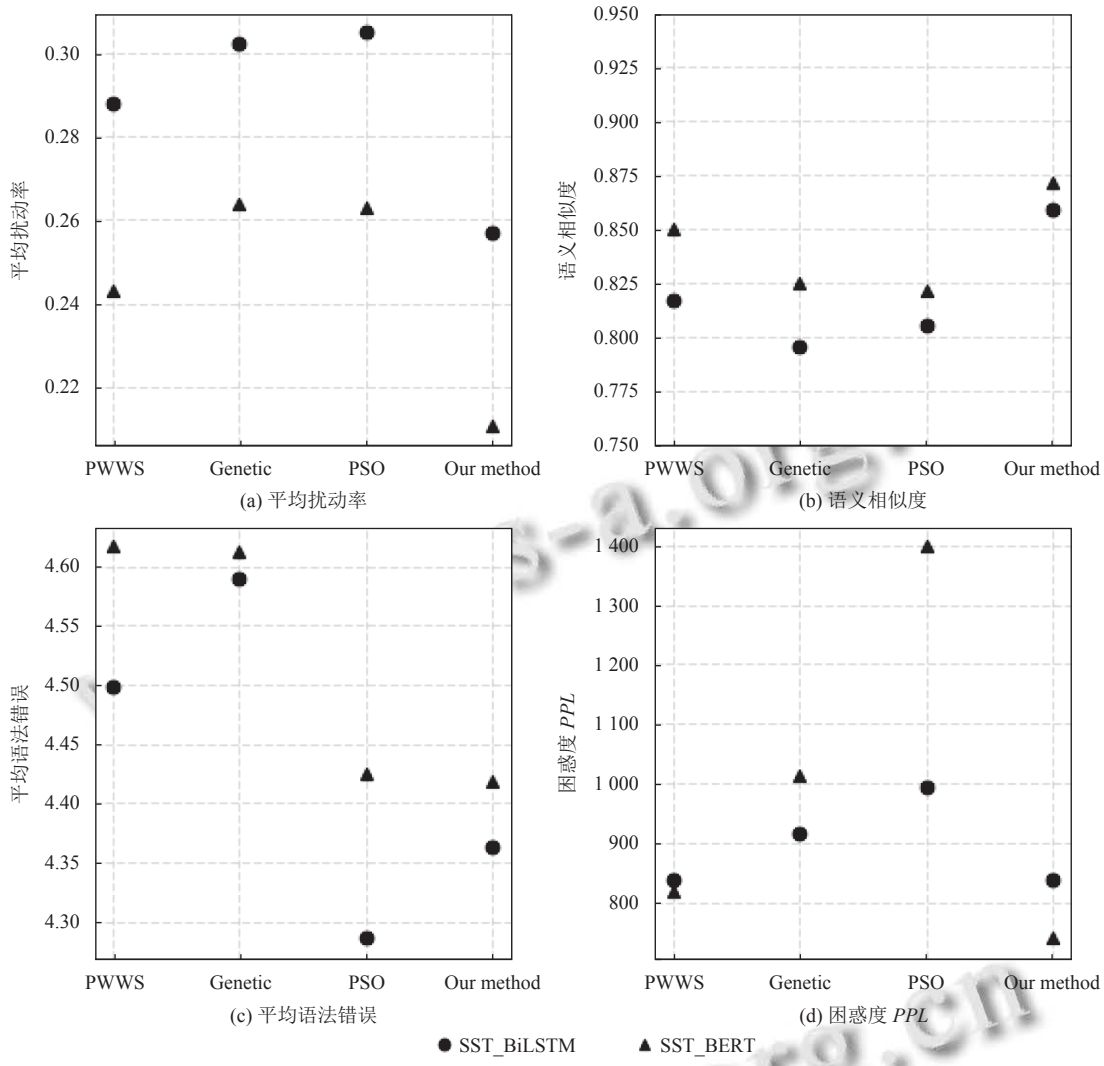

图 3 不同方法在 SST-2 数据集生成的对抗样本质量对比图

1) 短文本 SST-2 数据集: 本文方法生成的对抗样本在 4 个评测指标中基本全为最优, 可以实现对原始句子的最低扰动, 同时兼具最高的语义相似度, 较低的语法错误及句子困惑度, 生成样本的质量最好.

2) 长文本 AG-NEWS 数据集: 整体扰动率较高, 说明长文本在更改少量单词的情况下很难实现攻击, 但本文方法在 4 种方法中取得了最低的扰动率; 从语义相似度来看, 基于义原标注的方法可以使样本与原句达到 90% 以上的句子相似度, 进一步证明了义原方法的有效性; 从语法错误和困惑度图中可以看出, 基于义原标注的方法可以有效减少语法错误和句子困惑度, 增强文本可读性.

(3) 攻击耗时: 如表 4 中耗时一列所示, 由于 PWWS 方法直接定位到被攻击词, 大大缩小了搜索空间, 所以即使采用贪婪搜索的方式, 速度仍然较快. 而其余均为基于同义词或义原的词替换方法, 需要在整个搜索空间内进行组合优化, 并对每一个解作出判断, 由此攻击时间大大延长. 本文提出的人工蜂群算法的搜索效率相较遗传算法和粒子群优化算法来说, 攻击耗时又有增加, 主要原因是 3 个蜂种的转化增加了处理时间.

5 结论

为了进一步探索当前文本分类模型的脆弱性问题, 本文提出了一种基于人工蜂群算法的对抗样本生成, 拓展了文本领域词级对抗样本生成方法. 该方法不需要进行反向传播算法求解攻击梯度, 从而可以在不知道模型内部结构的情况下实现黑盒攻击. 在经典文本分类模型上的实验结果表明, 采用义原替换结合人工蜂群搜索的方法, 可以生成质量更好的文本对抗样本,

且攻击成功率高, 缺点是牺牲了一部分搜索效率. 未来可在此基础上研究相应的防御对策来增强自然语言处理模型的鲁棒性, 消除模型在实际部署应用中所面临的对抗攻击风险.

参考文献

1 Goodfellow IJ, Shlens J, Szegedy C. Explaining and harnessing adversarial examples. Proceedings of the 3rd International Conference on Learning Representations. San Diego: ICLR, 2015.

2 Szegedy C, Zaremba W, Sutskever I, *et al*. Intriguing properties of neural networks. arXiv: 1312.6199, 2013.

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

4 Eger S, Şahin GG, Rücklé A, *et al*. Text processing like humans do: Visually attacking and shielding NLP systems. Proceedings of 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Minneapolis: Association for Computational Linguistics, 2019. 1634–1647.

5 Jia R, Liang P. Adversarial examples for evaluating reading comprehension systems. Proceedings of 2017 Conference on Empirical Methods in Natural Language Processing. Copenhagen: Association for Computational Linguistics, 2017. 2021–2031.

6 Ribeiro MT, Singh S, Guestrin C. Semantically equivalent adversarial rules for debugging NLP models. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics. Melbourne: Association for Computational Linguistics, 2018. 856–865. [doi: 10.18653/v1/P18-1079]

7 Li LY, Ma RT, Guo QP, *et al*. BERT-ATTACK: Adversarial attack against BERT using BERT. Proceedings of 2020 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2020. 6193–6202.

8 Ren SH, Deng YH, He K, *et al*. Generating natural language adversarial examples through probability weighted word saliency. Proceedings 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]

9 Alzantot M, Sharma Y, Elgohary A, *et al*. Generating natural language adversarial examples. Proceedings of 2018 Conference on Empirical Methods in Natural Language Processing. Brussels: Association for Computational Linguistics, 2018. 2890–2896.

10 Zang Y, Qi FC, Yang CH, *et al*. Word-level textual adversarial attacking as combinatorial optimization. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, 2020. 6066–6080.

11 Dong ZD, Dong Q. HowNet and the Computation of Meaning. Hackensack: World Scientific Publishing Co., 2006.

12 Niu YL, Xie RB, Liu ZY, *et al*. Improved word representation learning with sememes. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. Vancouver: Association for Computational Linguistics, 2017. 2049–2058. [doi: 10.18653/v1/P17-1187]

13 Gu YH, Yan J, Zhu H, *et al*. Language modeling with sparse product of sememe experts. Proceedings of 2018 Conference on Empirical Methods in Natural Language Processing. Brussels: Association for Computational Linguistics, 2018. 4642–4651. [doi: 10.18653/v1/D18-1493]

14 Zhang L, Qi FC, Liu ZY, *et al*. Multi-channel reverse dictionary model. Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34(1): 312–319. [doi: 10.1609/AAAI.V34I01.5365]

15 Karaboga D. An idea based on honey bee swarm for numerical optimization. Technical Report, Kayseri: Erciyes University, 2005.

16 Socher R, Perelygin A, Wu J, *et al*. Recursive deep models for semantic compositionality over a sentiment treebank. Proceedings of 2013 Conference on Empirical Methods in Natural Language Processing. Seattle: Association for Computational Linguistics, 2013. 1631–1642.

17 Conneau A, Kiela D, Schwenk H, *et al*. Supervised learning of universal sentence representations from natural language inference data. Proceedings of 2017 Conference on Empirical Methods in Natural Language Processing. Copenhagen: Association for Computational Linguistics, 2017. 670–680. [doi: 10.18653/v1/D17-1070]

18 Devlin J, Chang MW, Lee K, *et al*. BERT: Pre-training of deep bidirectional transformers for language understanding. Proceedings of 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Minneapolis: Association for Computational Linguistics, 2019. 4171–4186. [doi: 10.18653/v1/N19-1423]

19 Radford A, Wu J, Child R, *et al*. Language models are unsupervised multitask learners. https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf. (2019-02-14).

(校对责编: 牛欣悦)