**引用格式：**邱士林, 刘启和, 周世杰, 等. 针对文本分类模型的高效硬标签对抗攻击方法 [J]. 电子科技大学学报, 2026, 55(1): 116-128.
QIU S L, LIU Q H, ZHOU S J, et al. Efficient hard-label adversarial attacks against natural language processing models[J]. Journal of University of Electronic Science and Technology of China, 2026, 55(1): 116-128.

# 针对文本分类模型的高效硬标签对抗攻击方法

邱士林*，刘启和，周世杰，曾 逸

（电子科技大学信息与软件工程学院, 成都 610054）

**摘要：**为了评估自然语言处理模型在真实应用场景下的对抗鲁棒性，硬标签设置下的黑盒对抗攻击技术逐渐引发关注。然而，受限于文本的离散性、反馈信息有限、查询次数限制等因素，现有硬标签对抗攻击方法通常存在查询次数多、对抗文本语义一致性低等问题，难以满足真实应用场景需求。因此，提出了一种高效的硬标签对抗攻击方法，该方法在对抗文本初始化阶段引入注意力机制，并在对抗文本语义优化阶段中提出了基于语义聚类的同义词搜索、基于语义梯度的动态扩展同义词搜索两个策略。实验结果表明，该方法能以少量查询来生成语义一致性高、自然流畅的高质量对抗文本。

**关键词：**对抗攻击；对抗样本；鲁棒性；自然语言处理；人工智能
**中图分类号：**TP391 **文献标志码：**A DOI: 10.12178/1001-0548.2024295

# Efficient hard-label adversarial attacks against natural language processing models

QIU Shilin*, LIU Qihe, ZHOU Shijie, and ZENG Yi

(School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China)

**Abstract:** Due to the necessity of verifying the robustness of natural language processing models against adversarial attacks in real-world application scenarios, black-box adversarial attack techniques under the hard-label setting have garnered increasing attention. However, due to the discrete nature of textual data, the limited information feedback from the victim model, and the constraints on the number of queries imposed by practical applications, existing hard-label adversarial attack methods usually suffer from excessive queries to the victim model and low semantic consistency of generated adversarial texts, rendering them inadequate for real-world applications. To this end, an efficient hard label adversarial attack method is proposed. In this method, an attention mechanism is introduced in the initialization stage of the adversarial text, while in the adversarial text semantic optimization stage, two strategies are proposed: the semantic clustering-based synonym search and the semantic gradient-based dynamic expansion synonym search. Experimental results demonstrate that the proposed method can efficiently generate high-quality adversarial text with high semantic consistency and natural fluency with a small number of queries.

**Key words:** adversarial attack; adversarial example; robustness; natural language processing; artificial intelligence

深度神经网络模型已被广泛应用于自然语言处理（natural language processing, NLP）领域，在情感分析[1]、机器翻译[2]、文本生成[3]、人机对话[4-5]等多种实际应用任务上表现出优秀的性能。

与此同时，文献 [6] 发现 NLP 模型面临严峻的对抗攻击威胁，攻击者通过向原文本添加微小噪声构造的对抗文本可操纵或误导深度神经网络模型，而这些对抗文本对人来说在语义上与原文本高度相似。

为了评估及提升 NLP 模型在面对对抗攻击威胁时的鲁棒性及安全性，研究者们提出了多种对抗

收稿日期：2024−10−28
基金项目：国家自然科学基金（62272089）
作者简介：邱士林，博士生，主要从事人工智能对抗安全方面的研究。
*通信作者 E-mail: shilinqiu@std.uestc.edu.cn

攻击方法[7]，由于文本数据的离散、易察觉和具有语义等特性，研究针对 NLP 模型的对抗攻击技术更具挑战性。

根据文本中被修改对象的语义粒度的不同，针对 NLP 模型的对抗攻击方法分为字符级攻击[8]、词级攻击[9]、句级攻击[10] 和多级攻击[11]，其中词级攻击方法通常基于同义词替换实现，与其他 3 类方法相比，其生成的对抗文本通常具有更高的语义一致性和文本质量。因此，词级攻击成为了当前文本对抗攻击技术的主流方法。

根据攻击者能够获取的目标模型信息的不同，针对 NLP 模型的对抗攻击方法分为白盒攻击和黑盒攻击。在白盒攻击[12-13] 过程中，攻击者对目标模型的训练数据、模型架构及参数具有完全访问权限，并利用这些丰富的信息生成对抗文本。而在黑盒攻击过程中，攻击者只能通过查询目标模型获得目标模型关于给定输入文本的输出结果。

根据目标模型反馈的信息类型的不同，黑盒攻击方法分为软标签攻击（基于分数）和硬标签攻击（基于决策），其中软标签对抗攻击利用目标模型输出的概率分布来生成对抗文本。如文献 [9] 先利用目标模型反馈的概率分布计算概率变化值、词显著性以及分类概率，然后采用同义词提取、词性检查、语义相似度检查 3 种策略来实现同义词替换扰动。此外，文献 [14] 采用注意力机制和局部敏感哈希策略来确定输入文本中每个词的重要性分数，并逐一对重要度高的词进行同义词替换扰动。但是，由于商用 API 等黑盒模型在实际应用中通常只提供给用户最终决策标签，软标签对抗攻击方法在真实应用场景中的适用性有限。

硬标签对抗攻击完全依赖目标模型反馈的最终决策标签来生成对抗文本，因而在实际场景中比软标签攻击具有更高的可行性与应用前景。文献 [15] 首先通过随机同义词替换策略确定具备对抗攻击能力的对抗文本，再以替换回原词的方式删除对抗文本中的冗余噪声，最后采用遗传算法优化对抗文本的语义。然而，这种基于遗传算法的优化方法在优化过程中需保留大量候选样本，需要向目标模型发起大量查询请求。为此，文献 [16] 将硬标签对抗攻击定义为词向量空间内的优化问题，并基于扰动矩阵相关的目标函数的梯度信息优化单一候选样本。文献 [17] 用词向量之间的差值来表征两个词之间的语义偏差，并通过将采样得到的同义词引入的语义偏差转换为连续词向量来估计目标模型决策边界的梯度，从而优化对抗文本语义。文献 [18] 在获得初始对抗文本后，首先通过对抗文本中的被扰动词与其对应原词之间的欧式距离确定词优化顺序，然后针对每个待优化词，通过随机选取同义词搜索锚点词，并通过比较语义相似度从所有同义词中选取更优替代词。与文献 [18] 相似，文献 [19] 根据被扰动词与其对应原词之间的余弦距离确定词优化顺序，通过随机选取同义词确定锚点词，然后通过随机选取锚点词的同义词来估计更新方向，从而优化对抗文本。虽然这些方法通过保留单一候选样本减少了向目标模型发起的查询请求次数，但却在一定程度上增加了得到次优对抗文本的可能性，因此，生成的对抗文本的语义相似度及扰动率有待进一步提升。

针对现有硬标签对抗攻击方法存在的上述问题，本文提出一种高效的硬标签词级对抗攻击方法。首先，将注意力机制引入到对抗文本初始化过程中，通过优先扰动对文本重要度高的词来降低扰动率；其次，在对抗文本语义优化过程中，将更优替代词搜索过程分为基于语义聚类的同义词搜索和基于语义梯度的动态扩展同义词搜索两个阶段，通过交替迭代两个过程，在扩展搜索空间的同时有效提升查询效率。实验结果表明，本文方法能够以少量查询高效地生成语义一致性高、自然流畅的高质量对抗文本。

## 1　本文方法

在硬标签场景下，给定原文本 $x=[w_1,w_2,w_3,w_4,\cdots,w_n]$ 和目标模型 $f$，对抗攻击者根据目标模型输出的关于 $x$ 的决策标签来生成与 $x$ 在语义上一致但使得 $f$ 产生错误决策的对抗文本 $x'$。

考虑到对抗文本的语义一致约束，即在攻击成功的情况下与原文本在语义上需保持一致，因此，最优对抗文本 $x^*$ 可由式 (1) 得到：

$$x^*=\mathrm{argmax}_{x'}\left[C(f,x,x')+S(x,x')\right] \tag{1}$$

式中，$x'$ 是对抗攻击方法生成的噪声文本；$C(f,x,x')$ 是对抗攻击分数，当 $f(x)\neq f(x')$，表明攻击成功，此时 $x'$ 称为对抗文本，$C(f,x,x')=0$；当 $f(x)=f(x')$，表明攻击失败，$C(f,x,x')=-\infty$；$S(x,x')$ 是 $x'$ 与 $x$ 之间的语义相似度分数。

本文聚焦于硬标签场景下基于同义词替换的词级对抗攻击，本文方法的总体框架如图 1 所示。此方法的攻击过程包括对抗文本初始化和对抗文本语

义优化两个阶段，其中前者的目标是获得具备攻击能力的初始对抗文本，而后者的目标是优化对抗文本的语义，提高对抗文本与原文本之间的语义一致性及文本质量。

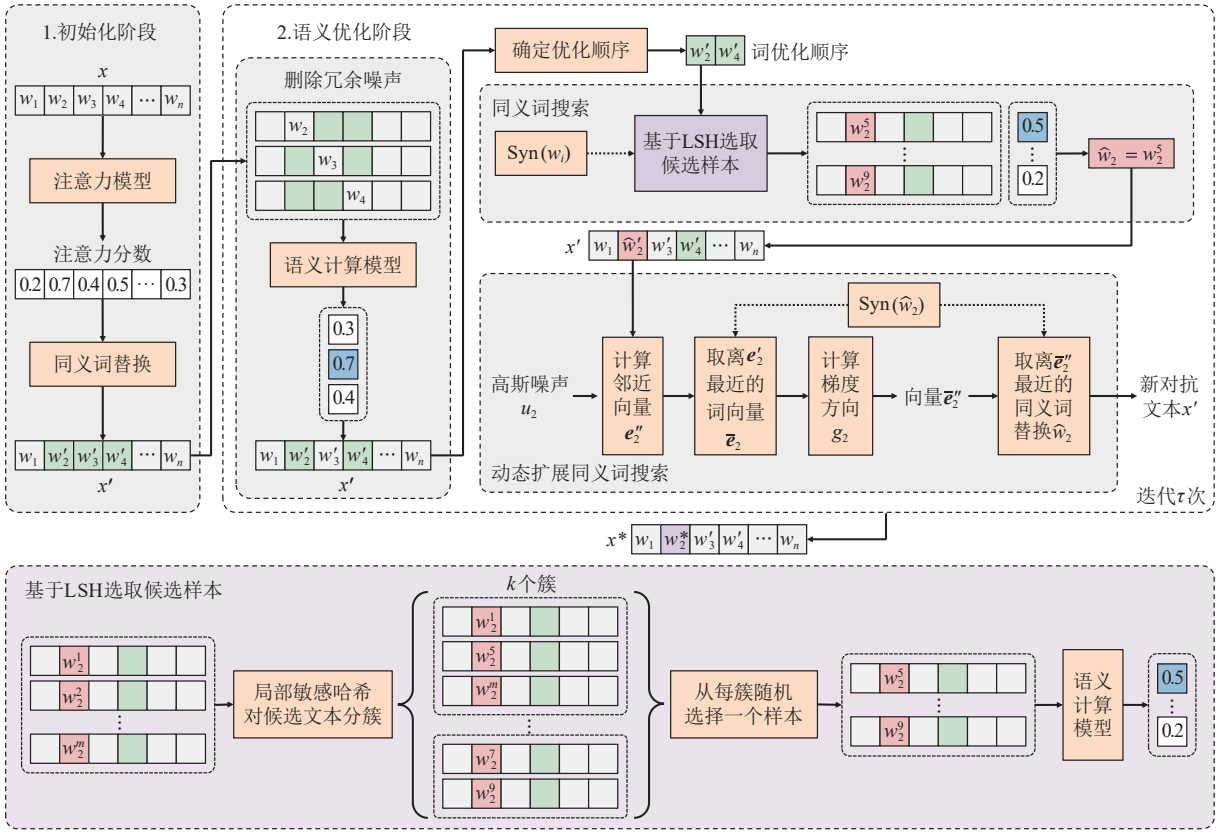


图 1 本文攻击方法总体框架图

### 1.1 对抗文本初始化策略

考虑到现有方法[15, 18-19]采用的初始化策略随机性大、容易引起不必要的查询请求，本文在初始化对抗文本阶段引入注意力机制，并以词注意力分数为概率分布对原文本中的词进行非重复采样，使得对原文本重要度高的词优先被扰动，以此减少所生成对抗文本中的被扰动词个数。

具体来说，对于原文本$x$中词性为名词、动词、形容词或副词的$w_i$，首先通过注意力模型计算每个$w_i$的注意力分数$a_i$，再根据式 (2) 对注意力分数进行归一化：

$$\bar{a}_i = \frac{a_i + 1}{\sum_{i=1}^{n}(a_i + 1)} \tag{2}$$

然后，以归一化后的注意力分数$\bar{a}_i$为概率分布对这些词进行非重复采样。最后，按照采样得到的排序结果，逐一用$w_i$的同义词集合$\mathrm{Syn}(w_i) = \{w_i^1, w_i^2, \cdots, w_i^m\}$中的一个随机同义词替换$w_i$，直至目标模型产生错误输出。由此，得到初始对抗文本$x' = [w_1, w_2', w_3', w_4', \cdots, w_n]$，其中，$w_i'$是替换$w_i$的同义词，也是对应于$w_i$的噪声词。

由于上述过程只考虑了所生成文本是否具备对抗攻击能力，而没有考虑其与原文本的语义一致性，因此，需进一步优化初始对抗文本以使其与原文本具有更高的语义一致性。

### 1.2 对抗文本语义优化策略

对抗文本的语义优化一般可通过在保证对抗文本攻击能力的情况下，将对抗文本中的$w_i'$替换为$w_i$或$w_i$的其他同义词。其中，将$w_i'$替换为$w_i$是最直接且有效的策略，而从$w_i$的同义词集合$\mathrm{Syn}(w_i)$选取更合适的词替换$w_i'$则能更好地平衡所生成文本的攻击能力和语义一致性。

本文以迭代方式交替执行上述两种策略，在每一次迭代过程中，首先尝试将对抗文本中的$w_i'$尽可能替换为$w_i$，以删除对抗文本中的冗余噪声；再逐一为对抗文本中的每个$w_i'$选取更优替代词。

在为$w_i'$选取更优替代词的过程中，遍历$w_i$的所

有同义词会导致对目标模型发起大量查询请求，这在查询次数有限的实际应用场景下并不适用。因此，在更优替代词搜索过程中，本文交替执行基于语义聚类的同义词搜索和基于语义梯度的动态扩展同义词搜索两个过程，从而实现在减少查询次数的同时，生成语义一致性高、自然流畅的高质量对抗文本。

总体而言，本文的对抗文本语义优化过程是一个迭代过程，每一次迭代分为以下 3 个步骤：首先，以迭代方式将对抗文本中的被扰动词$w_i'$尽可能替换为原词；然后，根据每个噪声词与其对应原词的词向量距离确定所有噪声词$w_i'$的优化顺序；最后，按照优化顺序，针对每个$w_i'$，先通过基于语义聚类的同义词搜索策略从$\mathrm{Syn}(w_i)$选取$w_i'$的更优替代词$\widehat{w}_i$，再通过基于语义梯度的动态扩展同义词搜索策略从$\mathrm{Syn}(\widehat{w}_i)$选取$\widehat{w}_i$的更优替代词$\overline{w}_i$，并用$\overline{w}_i$替换当前对文本中的$w_i'$。

#### 1.2.1　删除冗余噪声

本文采用与文献 [19] 相同的策略来删除对抗文本中的冗余噪声，即以迭代方式逐一选取使得生成的文本具备攻击能力且具有最高语义相似度的$w_i$替换$w_i'$。

具体来说，对于当前对抗文本$x'$，首先分别以$w_i$替换$x'$中的$w_i'$，得到候选样本集$M=\{x_1,x_2,\cdots,x_h\}$，其中，$h$是$x'$中的噪声词的总个数。再根据式 (3)，从$M$中选取具备攻击能力且具有最高语义相似度分数的候选样本作为新对抗文本$x'$：

$$x'=\mathrm{argmax}_{x_j\in M}\left[C(f,x,x_j)+S(x,x_j)\right] \tag{3}$$

若找到新样本，则重复上述过程；若没有满足条件的新样本，则停止迭代过程，并将当前$x'=[w_1,w_2',w_3,w_4',\cdots,w_n]$作为后续优化对象。

经过上述过程，与初始阶段得到的对抗文本相比，优化后的对抗文本尽可能多地保留了原词，因而在语义相似度及扰动率方面都有所提升。

#### 1.2.2　确定优化顺序

由于对抗文本中存在多个$w_i'$，因此，首先需要对这些词的优化顺序进行排序。为了保持所生成对抗文本的多样性，本文首先计算每个$w_i'$与其对应$w_i$在 Counter-Fitted 词向量空间的 $L_2$距离 $d_i=\|\boldsymbol{e}_i'-\boldsymbol{e}_i\|_2$，其中，$\boldsymbol{e}_i'$是$w_i'$的词向量，$\boldsymbol{e}_i$是$w_i$的词向量。然后，以式 (4) 计算所得值为概率分布，对所有噪声词$w_i'$执行非重复采样，从而确定优化顺序。

$$p_i=\frac{\exp(d_i+1)}{\sum_{i=1}^{n}\exp(d_i+1)} \tag{4}$$

#### 1.2.3　基于语义聚类的同义词搜索

对于当前对抗文本$x'$中的待优化词$w_i'$，为了避免搜索锚点词的过程遍历$w_i$的所有同义词，本文采用局部敏感哈希方法对高维空间中的句级向量进行聚类分簇，以此减少对目标模型的查询次数。

具体而言，对于待优化词$w_i'$，首先以$\mathrm{Syn}(w_i)$中的每个同义词分别替换对抗文本$x'$中的$w_i'$，得到候选文本集$\overline{M}=\{x_1',\cdots,x_t',\cdots,x_m'\}$，其中，$x_t'$是以$\mathrm{Syn}(w_i)$中的第$t$个同义词$w_i^t$替换$x'$中的$w_i'$得到的候选文本。

采用通用句子编码器[20]（universal sentence encoder, USE）将$\overline{M}$中的候选文本编码为$l$维向量，得到句级向量集$\overline{V}=\{\boldsymbol{v}_1,\boldsymbol{v}_2,\cdots,\boldsymbol{v}_m\}$。

然后，利用局部敏感哈希中的随机投影法（random projection method, RPM）计算$\overline{V}$中每个向量的哈希值。该过程首先随机生成$R$个$l$维单位向量$\{\boldsymbol{r}_1,\boldsymbol{r}_2,\cdots,\boldsymbol{r}_q,\cdots,\boldsymbol{r}_R\}$，其中，$\boldsymbol{r}_q$是第$q$个单位向量，再根据式 (5) 构建一组哈希函数，对于每个$\boldsymbol{v}_s$，其哈希值由拼接所有哈希函数的输出确定。

$$\mathrm{hash}_{\boldsymbol{r}_q}(\boldsymbol{v}_s)=\begin{cases}0 & \boldsymbol{r}_q\cdot\boldsymbol{v}_s<0\\ 1 & \boldsymbol{r}_q\cdot\boldsymbol{v}_s\geqslant 0\end{cases} \tag{5}$$

由此，具有相同哈希值的句级向量对应的候选文本被划分为同一簇，从而得到$K$个簇$\{c_1,c_2,\cdots,c_K\}$，其中，每个簇$c_k$包含具有相似句级向量的多个候选文本。

最后，从每个簇$c_k$中随机选择一个候选文本$x^{(k)}$，得到候选文本集$\widehat{M}=\left\{x^{(1)},x^{(2)},\cdots,x^{(K)}\right\}$，根据式 (6) 从$\widehat{M}$中选取具备攻击能力且具有最高语义相似度的候选文本作为新对抗文本$x'$，其第$i$个词即为$w_i'$的更优替代词$\widehat{w}_i$：

$$x'=\mathrm{argmax}_{x^{(k)}\in\widehat{M}}\left[C(f,x,x^{(k)})+S(x,x^{(k)})\right] \tag{6}$$

通过利用局部敏感哈希对候选文本进行语义聚类，上述过程所需的查询次数从同义词总数$m$减少至哈希簇总数$K$。由于$K$显著小于$m$，因此，上述过程的查询效率得到有效提升。

#### 1.2.4　基于语义梯度的动态扩展同义词搜索

考虑到固定且有限的同义词搜索空间易导致对抗文本语义一致性难以有效提升，本文将针对$w_i'$的更优替代词搜索空间动态扩展至$\widehat{w}_i$的同义词空

间，并基于语义梯度来搜索用于替换$\widehat{w}_i$的更优替代词$\overline{w}_i$。

给定噪声词$\widehat{w}_i$和原词$w_i$，它们在$d$维连续空间的词向量分别为$\widehat{\boldsymbol{e}}_i$和$\boldsymbol{e}_i$，首先从标准高斯分布中随机采样一个$d$维噪声向量$\boldsymbol{u}_i$，并根据式 (7) 得到$\widehat{\boldsymbol{e}}_i$的邻近向量$\boldsymbol{e}_i''$，其中，$\beta$是用于控制噪声大小的超参数：

$$\boldsymbol{e}_i'' = \widehat{\boldsymbol{e}}_i + \beta \cdot \boldsymbol{u}_i \tag{7}$$

给定由$\mathrm{Syn}(\widehat{w}_i)$中所有词的词向量构成的词向量集合$E_i = \left\{\boldsymbol{e}_i^1, \boldsymbol{e}_i^2, \cdots, \boldsymbol{e}_i^m\right\}$，根据式 (8) 得到与$\boldsymbol{e}_i''$的差值的$L_2$范数值最小的词向量$\overline{\boldsymbol{e}}_i$：

$$\overline{\boldsymbol{e}}_i = \mathrm{argmin}_{\boldsymbol{e}_i^j \in E_i} \| \boldsymbol{e}_i^j - \boldsymbol{e}_i'' \|_2 \tag{8}$$

然后，用$\overline{\boldsymbol{e}}_i$对应的同义词替换当前对抗文本$x'$中的$\widehat{w}_i$，得到候选文本$x''$，然后根据式 (9) 计算梯度方向$\boldsymbol{g}_i$：

$$\boldsymbol{g}_i = -\frac{S(x, x'') - S(x, x')}{\beta} \cdot \boldsymbol{u}_i \tag{9}$$

式中，$S(x, x'')$和$S(x, x')$分别是$x''$和$x'$与原文本之间的语义相似度分数。

最后，根据式 (10) 得到新向量$\overline{\boldsymbol{e}}_i''$：

$$\overline{\boldsymbol{e}}_i'' = \widehat{\boldsymbol{e}}_i + \gamma \cdot \boldsymbol{g}_i \tag{10}$$

式中，$\gamma$为超参数。从$\mathrm{Syn}(\widehat{w}_i)$中选取满足以下两个条件的同义词作为$\overline{w}_i$：1）其对应的词向量$\boldsymbol{e}_i^j$与$\overline{\boldsymbol{e}}_i''$的$L_2$距离最小；2）用其替换$x'$中的$\widehat{w}_i$，生成的文本与原文之间的语义相似度最大。最后，用$\overline{w}_i$替换$x'$中的$\widehat{w}_i$。

# 2 实验结果与分析

## 2.1 实验设置

### 2.1.1 数据集与目标模型

本文在文本分类和自然语言推理两个任务中评估所提出攻击方法。

对于文本分类任务，本文使用 4 个公开数据集，即 SST-2[21]、Movie Review（MR）[22]、AG's News（AG）[23] 和 Yahoo Answers（Yahoo）[23]，目标模型采用文献 [15] 使用的 3 个模型（BERT[24]、WordCNN[25] 和 WordLSTM[26]）。此外，为分析本文方法在攻击大语言模型时的表现，本文使用大语言模型 GPT-3.5 和 Claude 作为目标模型。

对于自然语言推理任务，本文使用 SNLI[27]、mMNLI[28] 和 mmMNLI[28] 这 3 个数据集，并采用 BERT[24] 作为目标模型。mMNLI 和 mmMNLI 是 MNLI 数据集[28] 的两个不同版本的测试集，其中，mMNLI 的文本与 MNLI 训练集的文本体裁一致，而 mmMNLI 的文本体裁与 MNLI 训练集不同。在对抗攻击过程中，攻击方法只修改每个样本的“假设”文本。

对 BERT、WordCNN 和 WordLSTM 模型实施攻击时，本文从每个数据集的测试集中随机选择 1 000 个样本作为测试数据，其详细信息如表 1 和表 2 所示。此外，本文从 SST-2 数据集的测试集中随机选取 100 个样本作为测试数据，用于对 GPT-3.5 和 Claude 实施对抗攻击。

**表 1 文本分类测试数据的详细信息**

| 数据集 | 平均文本长度 | 文本内容 | 分类总数 | 原始准确率/% | | |
|---|---|---|---|---|---|---|
| | | | | BERT | WordCNN | WordLSTM |
| SST-2 | 18.5 | 电影评论 | 2 | 93.60 | 80.90 | 82.60 |
| MR | 20.2 | 电影评论 | 2 | 85.00 | 76.50 | 78.00 |
| AG | 41.4 | 新闻文章 | 4 | 93.00 | 90.40 | 90.20 |
| Yahoo | 101.5 | 问答对 | 10 | 79.10 | 71.10 | 73.70 |

**表 2 自然语言推理测试数据的详细信息**

| 数据集 | “前提”平均长度 | “假设”平均长度 | 分类总数 | BERT原始准确率/% |
|---|---|---|---|---|
| SNLI | 15.43 | 8.39 | 3 | 82.20 |
| mMNLI | 21.27 | 11.44 | 3 | 85.10 |
| mmMNLI | 22.86 | 12.43 | 3 | 89.10 |

### 2.1.2 对比方法

本文使用 4 种硬标签词级对抗攻击方法作为基线方法，其中，HLBB[15] 基于遗传算法搜索对抗文本；LeapAttack[17] 通过在词向量空间估计决策边界的梯度来优化对抗文本；SSPAttack[18] 和 HQA-Attack[19] 采用“确定过渡词-搜索更优替代词”策略来搜索对抗文本。

### 2.1.3 评价指标

本文采用语义相似度、扰动率两个指标来评估对抗文本与原文本之间的语义一致性。其中，语义相似度（Sim）是原文本与对抗文本经 USE 编码得到的句级向量之间的余弦相似度，扰动率（Pert）是对抗文本中被修改的词的总数与文本总词数的百分比。语义相似度越大、扰动率越小，表明生成的对抗文本的语义一致性越高。

本文采用困惑度、语法错误数两个指标来评估对抗文本质量。其中，困惑度（PPL）衡量对抗文本的自然度及流畅度，由 GPT-2 模型[29] 计算得

到，语法错误数（GErr）是对抗文本中的语法错误总数与原文本中的语法错误总数的差值，由 LanguageTool（https://languagetool.org/）评估得到。语法错误数越小，困惑度越小，表明对抗文本质量越高。

本文采用不同查询次数限制下生成的对抗文本的语义相似度和扰动率来衡量对抗攻击方法的查询效率。此外，采用相同查询次数限制下生成的对抗文本的语义相似度、扰动率、困惑度、语法错误数及攻击成功率（Suc）来衡量对抗攻击方法的攻击能力。其中，攻击成功率是成功攻击目标模型的对抗文本总数与无攻击时被目标模型正确分类的文本总数的百分比。

#### 2.1.4　实验细节设置

本文采用文献 [15] 和文献 [19] 中的部分实验设置，包括使用 Spacy 库标注词性、使用 USE 计算语义相似度分数、以 Counter-Fitted 空间的词向量为依据为每个词选取 50 个同义词。此外，本文使用层次注意力网络[30]（hierarchical attention networks）计算注意力分数，对抗文本语义优化阶段的最大迭代次数$\tau=100$，超参数$R=5$，$\beta=0.5$，$\gamma=0.3$。在对 BERT、WordCNN、WordLSTM 实施攻击时，每次攻击的最大查询次数为 2 000 次，而在对 GPT-3.5 实施攻击时，每次攻击的最大查询次数为 350 次。

### 2.2　实验结果分析

#### 2.2.1　语义一致性分析

在最大查询次数为 2 000 次时，本文方法和基线方法针对文本分类和自然语言推理模型生成的对抗文本的语义相似度和扰动率如表 3 和表 4 所示。其中，每行中的语义相似度最大值和扰动率最小值通过加粗进行标注。与基线方法相比，本文方法能够在相同查询次数限制下生成语义一致性更高的对抗文本。

**表 3　攻击文本分类模型时生成的对抗文本的语义相似度和扰动率比较**　　%

| 数据集 | 目标模型 | HLBB | | LeapAttack | | SSPAttack | | HQA-Attack | | 本文方法 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Sim | Pert | Sim | Pert | Sim | Pert | Sim | Pert | Sim | Pert |
| SST-2 | BERT | 85.25 | 15.46 | 82.29 | 18.64 | 86.20 | 14.16 | 89.18 | 11.58 | **89.97** | **10.29** |
| | WordCNN | 86.49 | 14.74 | 84.76 | 16.63 | 87.59 | 13.60 | 90.29 | 11.25 | **90.61** | **10.23** |
| | WordLSTM | 86.40 | 15.35 | 84.47 | 16.72 | 87.26 | 13.87 | 90.31 | 11.56 | **90.72** | **10.12** |
| MR | BERT | 86.95 | 13.88 | 84.33 | 16.58 | 88.07 | 12.64 | 90.72 | 10.25 | **91.29** | **9.37** |
| | WordCNN | 87.69 | 13.73 | 86.09 | 15.44 | 88.73 | 12.86 | 91.48 | 10.56 | **91.61** | **9.64** |
| | WordLSTM | 86.70 | 14.28 | 85.00 | 16.15 | 88.04 | 13.08 | 91.20 | 10.74 | **91.49** | **9.84** |
| AG | BERT | 84.06 | 15.87 | 82.12 | 18.62 | 87.31 | 12.24 | 90.25 | 9.93 | **90.59** | **9.27** |
| | WordCNN | 90.33 | 12.18 | 88.24 | 14.87 | 91.69 | 9.99 | **94.10** | 8.04 | 94.07 | **7.65** |
| | WordLSTM | 85.43 | 15.40 | 82.66 | 18.66 | 87.65 | 12.66 | 90.42 | 10.16 | **90.80** | **9.48** |
| Yahoo | BERT | 90.38 | 7.17 | 89.12 | 8.67 | 90.88 | 6.27 | 92.72 | 5.26 | **92.73** | **5.21** |
| | WordCNN | 91.31 | 8.03 | 90.51 | 9.46 | 92.07 | 6.87 | 93.98 | 5.38 | **94.01** | **5.35** |
| | WordLSTM | 88.84 | 8.78 | 87.80 | 10.27 | 89.86 | 7.35 | **91.64** | 6.48 | **91.64** | **6.44** |
| 平均值 | | 87.49 | 12.91 | 85.62 | 15.06 | 88.78 | 11.30 | 91.36 | 9.27 | **91.63** | **8.57** |

**表 4　攻击自然语言推理模型时生成的对抗文本的语义相似度和扰动率比较**　　%

| 数据集 | 目标模型 | HLBB | | LeapAttack | | SSPAttack | | HQA-Attack | | 本文方法 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Sim | Pert | Sim | Pert | Sim | Pert | Sim | Pert | Sim | Pert |
| SNLI | BERT | 68.76 | 18.46 | 67.94 | 20.25 | 73.69 | 17.21 | 74.42 | 15.78 | **74.47** | **15.39** |
| mMNLI | | 81.19 | 14.80 | 79.76 | 16.70 | 84.43 | 13.17 | 85.60 | 11.87 | **85.67** | **11.40** |
| mmMNLI | | 82.97 | 13.88 | 80.36 | 16.21 | 85.87 | 12.70 | 86.57 | 11.44 | **86.91** | **10.87** |
| 平均值 | | 77.64 | 15.71 | 76.02 | 17.72 | 81.33 | 14.36 | 82.20 | 13.03 | **82.35** | **12.55** |

表 3 表明，在文本分类任务中，虽然本文方法在使用 AG 攻击 WordCNN 设置下生成的对抗文本的语义相似度比 HQA-Attack 方法低 0.03%，但扰动率却小了 0.39%。与 HQA-Attack 方法相比，本文方法的扰动率在 SST_2、MR 数据集时小了约 1%，使用 AG 数据集时小了 0.4%～0.7%，在使用 Yahoo 数据集时小了约 0.05%。这表明，本文方法能够有效地为短文本生成扰动率更小的对抗文本。从平均值来看，本文方法的语义相似度平均值和扰动率平均值分别为 91.63% 和 8.57%，比基线方法中表现最佳的 HQA-Attack 方法的语义相似度平均值高 0.27%，且比其扰动率平均值小 0.7%。

在自然语言推理任务中，本文方法使用SNLI、mMNLI、mmMNLI所生成的对抗文本的语义相似度和扰动率均优于所有基线方法。其中，在使用mmMNLI数据时，本文方法生成的对抗文本的语义相似度为86.91%，扰动率为10.87%，比LeapAttack方法的语义相似度高了6.55%，并且比其扰动率小了5.34%。从平均值来看，本文方法的语义相似度平均值为82.35%，比HQA-Attack方法高了0.15%，本文方法的扰动率平均值为12.55%，比HQA-Attack方法小了0.48%。

### 2.2.2 文本质量分析

在最大查询次数为2 000次时，本文方法和基线方法针对文本分类和自然语言推理模型生成的对抗文本的困惑度和语法错误数如表5和表6所示。其中，每行中的困惑度最小值和语法错误数最小值通过加粗进行标注。与基线方法相比，本文方法能够在相同查询次数限制下生成文本质量更高的对抗文本。

**表5 攻击文本分类模型时生成的对抗文本的语法错误数和困惑度比较**

| 数据集 | 目标模型 | HLBB | | LeapAttack | | SSPAttack | | HQA-Attack | | 本文方法 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | PPL | GErr | PPL | GErr | PPL | GErr | PPL | GErr | PPL | GErr |
| SST-2 | BERT | 1 661.50 | 0.27 | 834.49 | 0.33 | 2 000.07 | 0.24 | 1 884.47 | 0.23 | **788.13** | **0.21** |
| | WordCNN | 2 365.21 | 0.31 | 1 315.08 | 0.27 | 390.37 | 0.25 | 330.19 | 0.23 | **277.34** | **0.22** |
| | WordLSTM | 1 995.52 | 0.20 | 751.59 | 0.27 | 570.96 | 0.20 | **555.02** | **0.19** | 558.43 | **0.19** |
| MR | BERT | 2 287.31 | 0.25 | 2 997.56 | 0.27 | 1 581.15 | 0.24 | 1 587.16 | 0.20 | **1 569.69** | **0.18** |
| | WordCNN | 1 402.94 | 0.25 | 627.13 | 0.28 | 488.36 | 0.26 | 449.28 | 0.23 | **328.35** | **0.22** |
| | WordLSTM | 553.68 | 0.25 | 1 648.37 | 0.29 | 369.08 | 0.21 | 589.74 | 0.18 | **253.49** | **0.16** |
| AG | BERT | 365.80 | 0.50 | 437.49 | 0.57 | 238.50 | 0.30 | 187.63 | 0.30 | **169.53** | **0.29** |
| | WordCNN | 261.11 | 0.41 | 319.09 | 0.50 | 159.65 | 0.41 | 135.62 | 0.39 | **134.07** | **0.38** |
| | WordLSTM | 317.65 | 0.39 | 436.26 | 0.55 | 210.76 | 0.37 | 178.74 | 0.35 | **176.28** | **0.34** |
| Yahoo | BERT | 142.02 | 0.76 | 164.52 | 0.98 | 104.84 | 0.84 | **67.60** | **0.81** | 89.35 | 0.84 |
| | WordCNN | 109.24 | 0.86 | 154.31 | 1.00 | 80.36 | 0.77 | 63.07 | 0.82 | **62.26** | **0.75** |
| | WordLSTM | 127.96 | 1.10 | 199.30 | 1.23 | 93.06 | **0.85** | 86.28 | 1.12 | **85.94** | 1.02 |
| 平均值 | | 965.83 | 0.46 | 823.77 | 0.55 | 523.93 | 0.41 | 509.57 | 0.42 | **374.41** | **0.40** |

**表6 攻击自然语言推理模型时生成的对抗文本的语法错误数和困惑度比较**

| 数据集 | 目标模型 | HLBB | | LeapAttack | | SSPAttack | | HQA-Attack | | 本文方法 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | PPL | GErr | PPL | GErr | PPL | GErr | PPL | GErr | PPL | GErr |
| SNLI | BERT | 2 167.26 | 0.35 | 3 515.45 | 0.38 | 1 567.48 | 0.37 | 1 616.89 | 0.36 | **1 341.61** | **0.33** |
| mMNLI | | 1 243.74 | 0.34 | 1 791.15 | 0.37 | 790.50 | 0.31 | 808.49 | 0.28 | **700.03** | **0.26** |
| mmMNLI | | 1 541.57 | 0.32 | 983.08 | 0.35 | 1 105.15 | 0.31 | 1 006.22 | 0.30 | **893.50** | **0.28** |
| 平均值 | | 1 650.86 | 0.34 | 2 096.56 | 0.37 | 1 154.38 | 0.33 | 1 143.87 | 0.31 | **978.38** | **0.29** |

在文本分类任务中，虽然在使用SST-2攻击WordLSTM、使用Yahoo攻击BERT及WordLSTM下，本文方法生成的对抗文本的困惑度、语法错误数略大于SSPAttack方法或HQA-Attack方法，但在其他设置下，本文方法都优于基线方法。从平均值来看，本文方法生成的对抗文本的困惑度平均值为374.41，比HQA-Attack方法的困惑度小了135.16，本文方法在所生成对抗文本中新引入的平均语法错误数为0.40个，比SSPAttack方法小了0.01。

在自然语言推理任务中，本文方法在所有设置下生成的对抗文本的困惑度及语法错误数均优于4种基线方法。其中，在使用mMNLI数据集时，本文方法的语法错误数为0.26个，比LeapAttack方法少了0.11。从平均值来看，本文方法的困惑度平均值为978.38，比HQA-Attack方法小了165.49，本文方法的语法错误数为0.29个，比HQA-Attack方法小了0.02。

### 2.2.3 查询效率分析

在最大查询次数分别为100、300、500、700、1 000次时，本文方法和基线方法在文本分类和自然语言推理任务下攻击BERT模型时所生成的对抗文本的语义相似度和扰动率如图2和图3所示。

从图2可以看出，随着查询次数限制增大，本文方法和基线方法生成的对抗文本的语义相似度都有所提升，且使用较长文本时的语义相似度增长幅

度更大。本文方法在所有查询次数限制下生成的对抗文本的语义相似度比 HLBB 和 LeapAttack 方法高了 4%～6%。与 SSPAttack 方法相比，当使用 SST-2、MR、mMNLI 数据集时，本文方法在所有查询次数限制下的语义相似度提高了 2%～3%。与 HQA-Attack 方法相比，在使用 AG、Yahoo 数据集时，本文方法在查询次数限制为 100 次时的语义相似度提高了约 2%，这表明，与基线方法相比，本文方法在较小查询次数限制下能够为较长文本生成具有更高语义相似度的对抗文本。

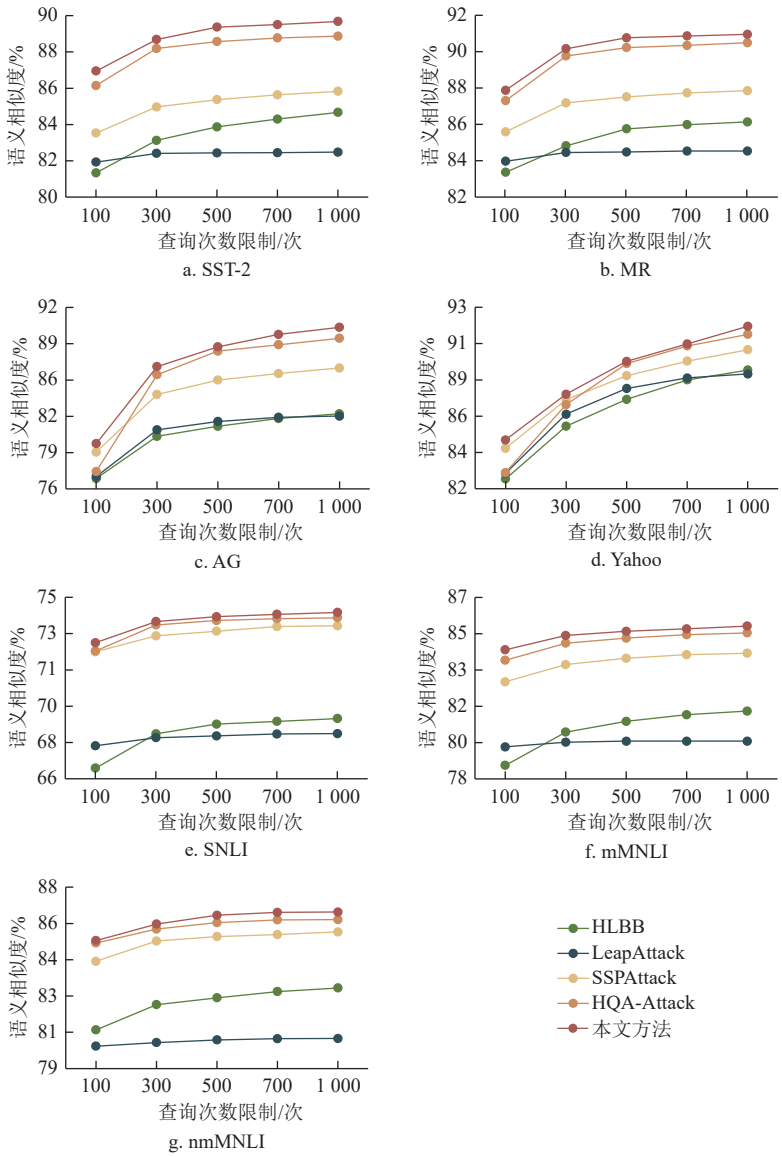


图 2　不同查询次数限制下攻击 BERT 模型时生成的对抗文本的语义相似度比较

从图 3 可以看出，当最大查询次数限制从 100 增加到 1 000 时，本文方法和基线方法所生成的对抗文本的扰动率都有所下降，且使用 AG 和 Yahoo 数据集生成的对抗文本的扰动率下降幅度较其他 4 个数据集更大。当使用 SST-2、MR、SNLI、mMNLI、mmMNLI 这 5 个文本长度较短的数据集时，本文方法在所有查询次数限制下所生成的对抗文本的扰动率，均小于 4 种基线方法。与基线方法相比，本文方法能够以少量查询为短文本生成具有更小扰动率的对抗文本。当使用 AG 和 YAhoo 两个具有较长文本的数据集时，本文方法在所有查询次数限制下的扰动率都小于 HLBB、LeapAttack、HQA-Attack 这 3 个基线方法。与 SSPAttack 方法相比，当查询次数限制增加至 300 次时，本文方法比 SSPAttack 方法具有更小的扰动率。与基线方法相比，本文方法能够在查询次数限制增大时为较长文本生成具有更小扰动率的对抗文本。

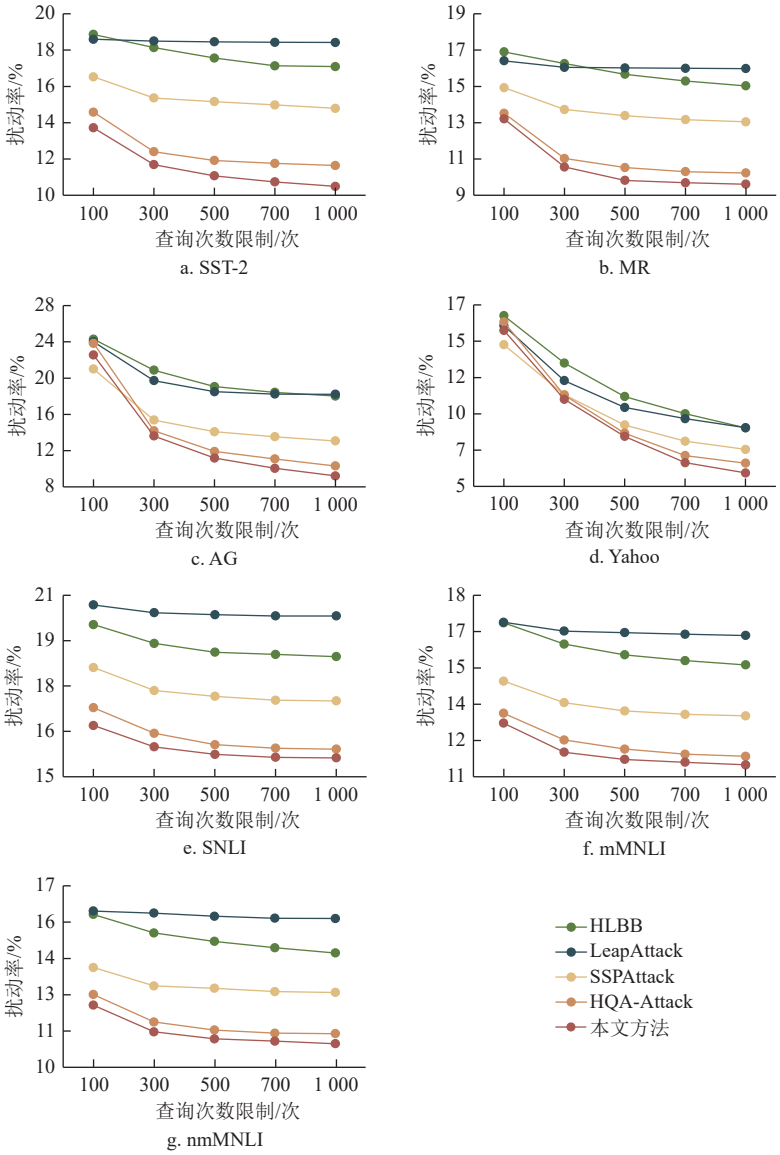


图 3 不同查询次数限制下攻击 BERT 模型时生成的对抗文本的扰动率比较

### 2.2.4 攻击能力分析

本节对比了最大查询次数为 2 000 次时，本文方法和基线方法使用 MR 和 Yahoo 数据集攻击 BERT 模型时所生成的对抗文本的语义一致性、文本质量以及攻击成功率如表 7 所示。其中，每行中的最优值通过加粗进行标注。

**表 7 对 BERT 模型的攻击能力比较**

| 方法 | MR | | | | | Yahoo | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Sim/% | Pert/% | PPL | GErr | Suc/% | Sim/% | Pert/% | PPL | GErr | Suc/% |
| HLBB | 86.95 | 13.88 | 2 287.31 | 0.25 | 98.59 | 90.38 | 7.17 | 142.02 | 0.76 | 99.12 |
| LeapAttack | 84.33 | 16.58 | 2 997.53 | 0.27 | 98.47 | 89.12 | 8.67 | 164.52 | 0.98 | 99.12 |
| SSPAttack | 88.07 | 12.64 | 1 581.15 | 0.24 | 98.59 | 90.88 | 6.27 | 104.84 | 0.84 | 99.24 |
| HQA-Attack | 90.72 | 10.25 | 1 587.16 | 0.2 | 98.59 | 92.72 | 5.26 | **67.6** | **0.81** | 99.12 |
| 本文方法 | **91.29** | **9.37** | **1 569.69** | **0.18** | **98.82** | **92.73** | **5.21** | 89.35 | 0.84 | **99.37** |

在语义一致性方面，使用 MR 和 Yahoo 数据集时，本文方法的语义相似度为 91.29%、92.73%，比 HQA-Attack 方法分别提高了 0.57%、0.01%，本文方法的扰动率为 9.37%、5.21%，比 HQA-Attack 方法分别减小了 0.88%、0.05%。

在文本质量方面，本文方法使用 MR 数据集生成的对抗文本的困惑度为 1 569.69，语法错误数为 0.18 个，比 HQA-Attack 方法分别减小了 17.47 和

0.02。在使用 Yahoo 数据集时，本文方法生成的对抗文本的困惑度和语法错误数仅比 HQA-Attack 方法略差，而优于其他 3 种基线方法。

表 7 表明，在相同查询次数限制下，本文方法能够在保持攻击成功率的情况下，大幅度提升了所生成对抗文本与原文本之间的语义一致性。

### 2.2.5　消融实验分析

为了分析本文的不同优化策略在优化对抗文本语义一致性及文本质量、提高查询效率方面的贡献度。本节对比了在最大查询次数限制为 1 000 次的情况下，本文方法和 4 种对比策略使用 SST-2 数据集攻击 BERT 模型所生成的对抗文本的表现，如表 8 所示。其中，4 种对比策略与本文方法的不同之处在于：“策略 A”在随机扰动过程中不使用注意力模型，而是采用随机扰动方式来生成初始对抗文本；“策略 B”没有冗余噪声删除过程，直接以当前对抗文本作为同义词搜索过程的优化对象；“策略 C”没有同义词搜索过程，直接以当前对抗文本作为动态扩展同义词搜索过程的优化对象；“策略 D”没有动态扩展同义词搜索过程，直接以当前对抗文本作为下一轮迭代中冗余噪声删除过程的优化对象。

**表 8　不同策略攻击 BERT 模型的对抗文本比较**

| 方法 | 指标 | | | | |
|---|---|---|---|---|---|
| | GErr | PPL | Pert/% | Sim/% | 查询次数 |
| 策略A | **0.21** | 1 863.39 | 10.68 | 88.86 | 289.65 |
| 策略B | 0.24 | 645.76 | 11.24 | 88.25 | 344.25 |
| 策略C | 0.27 | **523.87** | 12.57 | 87.80 | **205.75** |
| 策略D | 0.23 | 1 935.36 | 11.37 | 89.21 | 273.72 |
| 本文方法 | **0.21** | 683.47 | **10.49** | **89.68** | 302.11 |

从表 8 中可以看出，与策略 A 相比，本文方法生成的对抗文本的困惑度减小 1 179.92，扰动率减小 0.19%，语义相似度提高 0.82%；与策略 B 相比，本文方法的语法错误数减小 0.03，扰动率减小 0.75%，语义相似度提高 1.43%，查询次数减少 42.14；与策略 C 相比，本文方法的语法错误数减小 0.06，扰动率减小 2.08%，语义相似度提高 1.88%；与策略 D 相比，本文方法的语法错误数减小 0.02，困惑度减小 1 251.89，扰动率减小 0.88%，语义相似度提高 0.47%。

由此可见，本文提出的基于注意力机制的随机扰动策略、基于最高语义相似度的冗余噪声删除策略、基于语义聚类的同义词搜索策略、基于语义梯度的动态扩展同义词搜索策略都能在一定程度上提升所生成对抗文本的语义一致性。此外，基于注意力机制的随机扰动策略和基于语义梯度的动态扩展同义词搜索策略能够提升对抗文本的流畅度，基于最高语义相似度的冗余噪声删除策略能够减少生成过程所需的查询次数，而基于语义聚类的同义词搜索策略能够有效减少对抗文本中的语法错误数。因此，本文方法中的 4 种优化策略协同互补，从而有效提升了对抗文本的语义一致性及生成过程的查询效率。

### 2.2.6　攻击大语言模型的表现分析

为了评估本文方法在实际应用场景下的表现，对比了在最大查询次数为 350 次时，本文方法和基线方法使用 100 个 SST-2 文本攻击 GPT-3.5 和 Claude 模型所生成的对抗文本的语义一致性和文本质量，如表 9 所示。其中每行的最优值通过加粗进行标注。

**表 9　攻击大语言模型时生成的对抗文本表现比较**

| 方法 | GPT-3.5 | | | | | Claude | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Sim/% | Pert/% | PPL | GErr | 攻击成功样本数 | Sim/% | Pert/% | PPL | GErr | 攻击成功样本数 |
| HLBB | 76.76 | 21.83 | 921.50 | 0.37 | 67 | 81.90 | 16.65 | 706.40 | 0.21 | 47 |
| LeapAttack | 81.79 | 16.54 | 538.82 | 0.55 | 59 | 82.08 | 17.16 | 545.53 | 0.31 | 45 |
| SSPAttack | 81.91 | 17.59 | 633.63 | 0.26 | **80** | 86.22 | 13.75 | 399.73 | 0.18 | **51** |
| HQA-Attack | 82.24 | 16.69 | 702.16 | 0.29 | 72 | 89.15 | 11.81 | 393.40 | 0.15 | **51** |
| 本文方法 | **82.60** | **16.19** | **528.01** | **0.22** | 79 | **89.28** | **10.15** | **369.37** | **0.14** | 50 |

在攻击 GPT-3.5 模型时，与基线方法的最优值相比，本文方法的扰动率比 LeapAttack 方法减小了 0.35%，语义相似度比 HQA-Attack 方法提高了 0.36%，语法错误数比 SSPAttack 方法减少了 0.04 个，困惑度比 LeapAttack 方法减小了 10.81。

在攻击 Claude 模型时，与基线方法中表现最优的 HQA-Attack 方法相比，本文方法生成的对抗文本的扰动率小了 1.66%，语义相似度提高了 0.13%，语法错误数减少了 0.01 个，困惑度减小了 24.03。

由此可见，在相同查询次数限制下，与基线方法相比，本文方法生成的对抗文本对大语言模型具有更强的攻击能力。

此外，本节对比了在最大查询次数分别为50、100、200、300、350次时，本文方法和基线方法攻击GPT-3.5、Claude模型所生成的对抗文本的扰动率和语义相似度如图4所示。

在攻击GPT-3.5模型时，与HLBB方法相比，本文方法在所有查询次数限制下的扰动率减小了约5%～6%，语义相似度提高了约5%。与LeapAttack、SSPAttack、HQA-Attack方法相比，本文方法在最大查询次数为50次时的扰动率和语义相似度大幅优于3种基线方法。

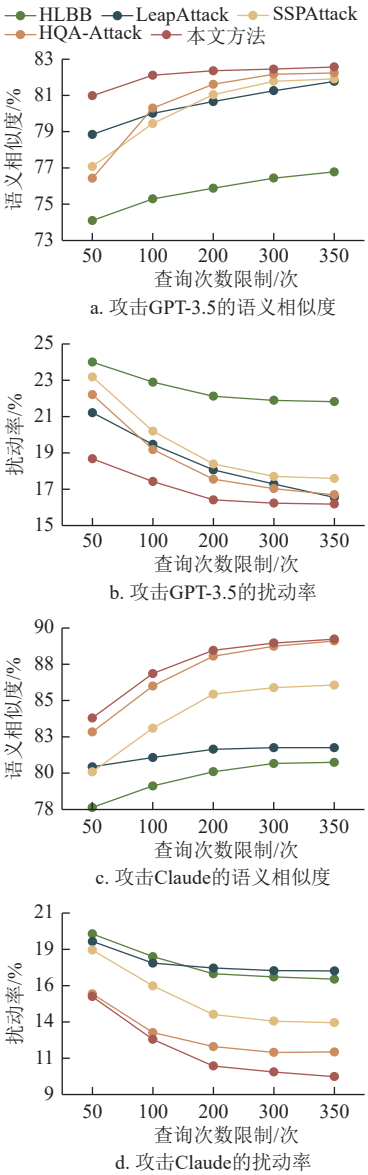


图4 不同查询次数限制下攻击大语言模型时生成的对抗文本表现比较

在攻击Claude模型时，本文方法在所有查询次数限制下生成的对抗文本的扰动率比HLBB、LeapAttack方法减小了约4%～7%，语义相似度提高了5%～7%。虽然HQA-Attack方法的扰动率和语义相似度与本文方法相近，但在所有查询次数限制下，本文方法都优于HQA-Attack方法。

由此可见，与基线方法相比，本文方法在不同查询次数限制下都能生成对大语言模型更具攻击性的对抗文本。

### 2.2.7 对抗文本示例分析

本文方法和4种基线方法使用MR数据攻击BERT模型时生成的对抗文本实例如图5和图6所示。其中，“Pred”是目标模型对输入文本的预测结果，“Sim”是对抗文本与原文本之间的语义相似度，“Qrs”是生成该对抗文本所消耗的查询次数，对抗文本中加粗标注的词是替换对应原词的同义词。在图5中，本文方法消耗了22次查询，生成的对抗文本与原文本之间的语义相似度为88.28%。虽然HLBB和LeapAttack方法只查询了27次目标模型，但它们生成的对抗文本的语义相似度仅为83.04%，比本文方法降低了5.24%。值得注意的是，虽然本文方法和SSPAttack、HQA-Attack方法生成了相同的对抗文本（用形容词“entranced”替换形容词“enamored”），但本文方法产生的查询消耗仅为基线方法的四分之一。

| |
|---|
| 原文本 (Pred=Negative)<br>Davis is so enamored of her own creation that she can't see how insufferable the character is. |
| HLBB生成的对抗文本 (Pred=Positive，Sim=83.04%，Qrs=27)<br>Davis is so **enthused** of her own creation that she can't see how insufferable the character is. |
| LeapAttack生成的对抗文本 (Pred=Positive，Sim=83.04%，Qrs=27)<br>Davis is so **enthused** of her own creation that she can't see how insufferable the character is. |
| SSPAttack生成的对抗文本 (Pred=Positive，Sim=88.28%，Qrs=79)<br>Davis is so **enthused** of her own creation that she can't see how insufferable the character is. |
| HQA-Attack生成的对抗文本 (Pred=Positive，Sim=88.28%，Qrs=74)<br>Davis is so **enthused** of her own creation that she can't see how insufferable the character is. |
| 本文方法生成的对抗文本 (Pred=Positive，Sim=88.28%，Qrs=22)<br>Davis is so **enthused** of her own creation that she can't see how insufferable the character is. |

图5 使用MR数据生成的对抗文本示例1

在图6中，本文方法生成的对抗文本与原文本之间的语义相似度为92.17%，且仅消耗了8次查询次数，查询效率远高于4种基线方法。在文本质量方面，本文方法将原文本中的助动词（“may”）修改为副词（“conceivably”），确保了文本的语法正确性，而HLBB方法将助动词（“may”）修改为形容词（“risque”），导致

文本产生语法错误。此外，在文本语义一致性方面，本文方法只修改了一个词，扰动率表现优于 HLBB 和 SSPAttack 方法，虽然语义相似度比 LeapAttack 方法低了 1.2%，但查询次数仅为 LeapAttack 方法的七分之一，因而在查询受限场景下的适用性更高。

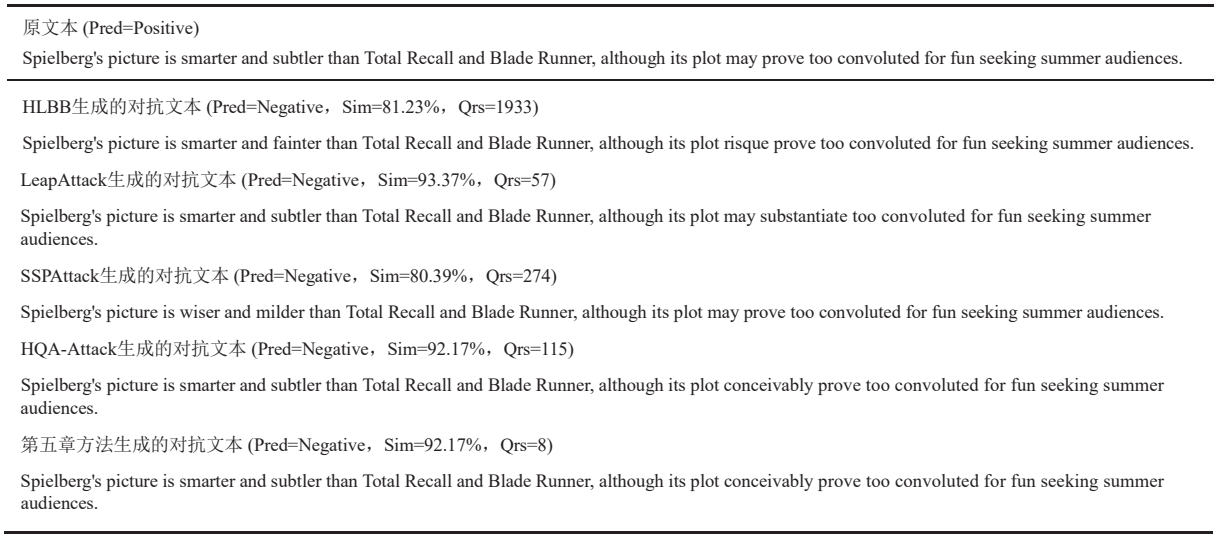


图 6　使用 MR 数据生成的对抗文本示例 2

由此可见，相较于基线方法，本文方法在查询效率方面的表现显著优于 4 种基线方法，同时，生成的对抗文本的文本质量及语义一致性表现优于大部分基线方法。

## 3　结束语

针对硬标签场景下现有对抗攻击方法存在的查询次数多、对抗文本语义一致性低的问题，本文提出了一种高效的硬标签对抗攻击方法。该方法在对抗文本语义优化过程中，通过基于语义聚类的同义词搜索和基于语义梯度的动态扩展同义词搜索两个阶段的交替迭代，为每个噪声词搜索更优替代词。

在未来的研究中，可进一步探索面向多模态任务及大语言模型的文本对抗攻击技术，从而验证这些模型在实际应用场景下的对抗鲁棒性；探索融合基于决策和基于迁移方法优势的文本对抗攻击技术，从而提升文本对抗攻击技术在实际应用场景下的实用性和有效性。

## 参 考 文 献

[1] BORDOLOI M, BISWAS S K. Sentiment analysis: A survey on design framework, applications and future Scopes[J]. Artificial Intelligence Review, 2023, 56(11): 12505-12560.

[2] ZHANG B, HADDOW B, BIRCH A. Prompting large language model for machine translation: A case study [C]//International Conference on Machine Learning. [S.l.]: PMLR, 2023: 41092-41110.

[3] LI X, THICKSTUN J, GULRAJANI I, et al. Diffusion-lm improves controllable text generation[J]. Advances in Neural Information Processing Systems, 2022, 35: 4328-4343.

[4] ZHANG W N, CUI Y M, ZHANG K Y, et al. A static and dynamic attention framework for multi turn dialogue generation[J]. ACM Transactions on Information Systems, 2023, 41(1): 1-30.

[5] LYU A, LI J P, XIE S F, et al. Envisioning future from the past: Hierarchical duality learning for multi-turn dialogue generation[C]//Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics. Stroudsburg, PA: [s.n.], 2023: 7382-7394.

[6] PAPERNOT N, MCDANIEL P, SWAMI A, et al. Crafting adversarial input sequences for recurrent neural networks [C]//Proceedings of the 2016 IEEE Military Communications Conference. [s.l.]: IEEE, 2016: 49-54.

[7] QIU S L, LIU Q H, ZHOU S J, et al. Adversarial attack and defense technologies in natural language processing: A survey[J]. Neurocomputing, 2022, 492: 278-307.

[8] GAO J, LANCHANTIN J, SOFFA M L, et al. Black-box generation of adversarial text sequences to evade deep learning classifiers[C]//Proceedings of the IEEE Security and Privacy Workshops. [S.l.]: IEEE, 2018: 50-56.

[9] JIN D, JIN Z J, ZHOU J T, et al. Is BERT really robust? A strong baseline for natural language attack on text classification and entailment[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34(5): 8018-8025.

[10] WANG T, WANG X, QIN Y, et al. Cat-gen: Improving robustness in NLP models via controlled adversarial text generation[EB/OL]. [2024-10-05]. https://arxiv.org/pdf/

2010.02338.
[11] ZHENG X Q, ZENG J H, ZHOU Y, et al. Evaluating and enhancing the robustness of neural network-based dependency parsing models with adversarial examples [C]//Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Stroudsburg, PA: [s.n.], 2020: 6600-6610.
[12] CHENG M H, YI J F, CHEN P Y, et al. Seq2Sick: Evaluating the robustness of sequence-to-sequence models with adversarial examples[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34(4): 3601-3608.
[13] ATANASOVA P. Generating label cohesive and well-formed adversarial claims[M]//Accountable and Explainable Methods for Complex Reasoning over Text. Cham: Springer Nature Switzerland, 2020: 65-79.
[14] MAHESHWARY R, MAHESHWARY S, PUDI V. A strong baseline for query efficient attacks in a black box setting[EB/OL]. [2024-09-10]. https://arxiv.org/pdf/2109.04775.
[15] MAHESHWARY R, MAHESHWARY S, PUDI V. Generating natural language attacks in a hard label black box setting[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2021, 35(15): 13525-13533.
[16] YE M C, MIAO C L, WANG T, et al. TextHoaxer: Budgeted hard-label adversarial attacks on text[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2022, 36(4): 3877-3884.
[17] YE M C, CHEN J H, MIAO C L, et al. LeapAttack: Hard-label adversarial attack on text via gradient-based optimization[C]//Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. New York: ACM, 2022: 2307-2315.
[18] LIU H, XU Z, ZHANG X T, et al. SSPAttack: A simple and sweet paradigm for black-box hard-label textual adversarial attack[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2023, 37(11): 13228-13235.
[19] LIU H, XU Z, ZHANG X, et al. HQA-attack: Toward high quality black-box hard-label adversarial attack on text[J]. Advances in Neural Information Processing Systems, 2023, 36: 51347-51358.
[20] CER D, YANG Y F, KONG S Y, et al. Universal sentence encoder for English[C]//Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. Stroudsburg, PA: [s.n.], 2018: 169-174.
[21] SOCHER R, PERELYGIN A, WU J, et al. Recursive deep models for semantic compositionality over a sentiment treebank[C]//Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. Stroudsburg, PA: [s.n.], 2013: 1631-1642.
[22] PANG B, LEE L. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales[EB/OL]. [2024-09-17]. https://arxiv.org/pdf/cs.CL/0506075.
[23] ZHANG X, ZHAO J, LECUN Y. Character-level convolutional networks for text classification[J]. Advances in Neural Information Processing Systems, 2015, 28:1-9.
[24] DEVLIN J. Bert: Pre-training of deep bidirectional transformers for language understanding[EB/OL]. [2024-09-24]. https://arxiv.org/pdf/1810.04805.
[25] KIM Y. Convolutional neural networks for classification[EB/OL]. [2024-09-03]. https://arxiv.org/pdf/1408.5882.
[26] HOCHREITER S, SCHMIDHUBER J. Long short-term memory[J]. Neural Computation, 1997, 9(8): 1735-1780.
[27] BOWMAN S R, ANGELI G, POTTS C, et al. A large annotated corpus for learning natural language inference [C]//Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Lisbon: ACL, 2015: 632-642.
[28] WILLIAMS A, NANGIA N, BOWMAN S. A broad-coverage challenge corpus for sentence understanding through inference[EB/OL]. [2024-08-22]. https://arxiv.org/pdf/1704.05426.
[29] RADFORD A, WU J, CHILD R, et al. Language models are unsupervised multitask learners[J]. Open AI blog, 2019, 1(8): 9.
[30] YANG Z C, YANG D Y, DYER C, et al. Hierarchical attention networks for document classification[C]// Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Stroudsburg, PA: [s.n.], 2016: 1480-1489.

责任编辑 税 红