软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2023,34(11):5143−5161 [doi: 10.13328/j.cnki.jos.006744]

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

基于汉语特征的中文对抗样本生成方法*

李相葛, 罗 红, 孙 岩

(北京邮电大学 计算机学院 (国家示范性软件学院), 北京 100876)

通信作者: 罗红, E-mail: luoh@bupt.edu.cn

摘 要: 深度神经网络容易受到来自对抗样本的攻击, 例如在文本分类任务中修改原始文本中的少量字、词、标点符号即可改变模型分类结果. 目前 NLP 领域对中文对抗样本的研究较少且未充分结合汉语的语言特征. 从中文情感分类场景入手, 结合了汉语象形、表音等语言特征, 提出一种字词级别的高质量的对抗样本生成方法 CWordCheater, 涵盖字音、字形、标点符号等多个角度. 针对形近字的替换方式, 引入 ConvAE 网络完成汉字视觉向量的嵌入, 进而生成形近字替换候选池. 同时提出一种基于 *USE* 编码距离的语义约束方法避免对抗样本的语义偏移问题. 构建一套多维度的对抗样本评估方法, 从攻击效果和攻击代价两方面评估对抗样本的质量. 实验结果表明, CWordAttacker 在多个分类模型和多个数据集上能使分类准确率至少下降 27.9%, 同时拥有更小的基于视觉和语义的扰动代价.

关键词: 中文情感分类; 对抗样本; 汉语特征

中图法分类号: TP18

中文引用格式: 李相葛, 罗红, 孙岩. 基于汉语特征的中文对抗样本生成方法. 软件学报, 2023, 34(11): 5143–5161. http://www.jos.org.cn/1000-9825/6744.htm

英文引用格式: Li XG, Luo H, Sun Y. Adversarial Sample Generation Method Based on Chinese Features. Ruan Jian Xue Bao/Journal of Software, 2023, 34(11): 5143–5161 (in Chinese). http://www.jos.org.cn/1000-9825/6744.htm

Adversarial Sample Generation Method Based on Chinese Features

LI Xiang-Ge, LUO Hong, SUN Yan

(School of Computer Science (National Pilot Software Engineering School), Beijing University of Posts and Telecommunications, Beijing 100876, China)

Abstract: Deep neural networks are vulnerable to attacks from adversarial samples. For instance, in a text classification task, the model can be fooled by modifying a few characters, words, or punctuation marks in the original text to change the classification result. Currently, studies of Chinese adversarial samples are limited in the field of natural language processing (NLP), and they fail to give due consideration to the language features of Chinese. This study proposes CWordCheater, a character-level and word-level high-quality method to generate adversarial samples covering the aspects of pronunciation, glyphs, and punctuation marks by approaching from the Chinese sentiment classification scenarios and taking into account the pictographic, alphabetic, and other language features of Chinese. The ConvAE network is adopted to embed Chinese visual vectors for the replacement modes of visually similar characters and further obtain the candidate pool of such characters for replacement. Moreover, a semantic constraint method based on universal sentence encoder (*USE*) distance is proposed to avoid the semantic offset in the adversarial sample. Finally, the study proposes a set of multi-dimensional evaluation methods to evaluate the quality of adversarial samples from the two aspects of attack effect and attack cost. Experiment results show that CWordAttacker can reduce the classification accuracy by at least 27.9% on multiple classification models and multiple datasets and has a lower perturbation cost based on vision and semantics.

Key words: Chinese sentiment classification; adversarial sample; Chinese feature

近年来深度学习在自然语言处理领域取得了大量突破性进展, 但有研究指出深度神经网络模型面对对抗样本的攻击时不够鲁棒[1]. 通过在输入样本中添加人类不易察觉的扰动, 可以误导神经网络使其输出错误的分类结

* 基金项目: 国家自然科学基金 (62172051, 61877005)
收稿时间: 2022-01-08; 修改时间: 2022-04-13; 采用时间: 2022-06-29; jos 在线出版时间: 2023-06-16
CNKI 网络首发时间: 2023-06-19

果[2]. 为了确保神经网络的可靠性和鲁棒性, 越来越多的研究工作开始关注对抗样本的生成和攻击方式. 目前在计算机视觉领域, 关于对抗样本的攻击和防御策略已经有充分的探索和实践[3]. 但是由于文本数据的离散性, 自然语言处理领域中对抗样本的生成和攻击仍具有挑战性, 它们需满足以下特征: (1) 能够使神经网络模型输出错误的分类结果; (2) 与原始文本相似度较高, 人类在阅读时能理解其原始语义.

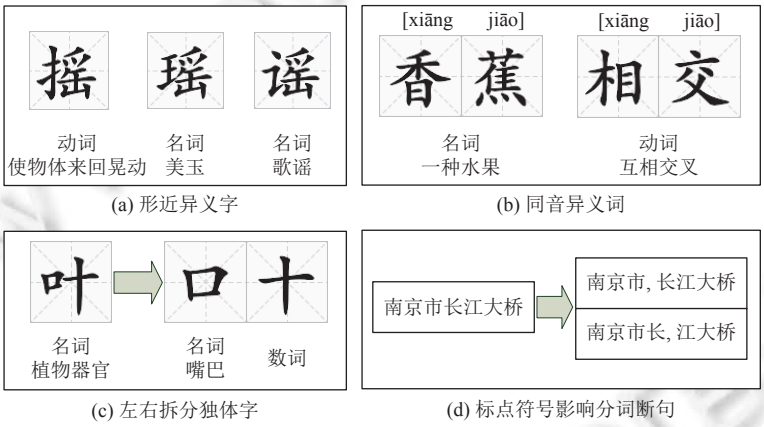
目前文本对抗样本生成方法主要包括基于语言特征的扰动和基于梯度的扰动.

● 基于梯度的攻击方法主要面向白盒攻击场景, 攻击者知晓模型的结构和所有参数, 通过梯度计算各个词语的扰动优先级并返回最优攻击选项. 其难点在于文本的数据空间是离散的, 难以使用梯度进行计算和优化.

● 基于语言特征的扰动方法主要面向黑盒攻击场景, 攻击者仅知晓被攻击模型的输出结果, 无法获悉模型的结构和参数; 这类攻击首先根据词语贡献度确定其扰动优先级, 然后使用语义计算器从待扰动词语的攻击候选池(例如近义词词典) 中进行搜索并返回最优攻击选项. 其攻击效果较为依赖于语言学先验知识, 需要针对不同语言单独设计规则.

对于英文文本数据, 常规的扰动方法是对单词的部分字母进行插入、删除、交换顺序、替换等操作. 而中文中的每个汉字都是不可拆分的最小独立单元, 若对更细粒度的偏旁部首使用英文字母的扰动方法, 容易生成非法字符输出. 同时与作为表音文字的英文相比, 汉语拥有多种鲜明的语言特征.

(1) 作为一种象形文字, 汉语中的字形包含多样的语义信息, 字形相近的汉字可以表达不同的语义信息, 例如图 1(a) 中“摇”“瑶”和“谣”属于形近异义字, 字形接近且语义不同. 而英文中由近似字母组合构成的单词大多拥有共同的词根, 表达的语义相近.



(a) 形近异义字 (b) 同音异义词

(c) 左右拆分独体字 (d) 标点符号影响分词断句

图 1 汉字的语言特征示例

(2) 作为一种意音文字, 汉语拥有 4 种拼音声调的组合, 其近音词数量远多于英语, 且读音相似的词语可以表达不同的语义信息, 例如图 1(b) 中“香蕉”和“相交”属于同音异义词, 读音相同且语义不同而英文中的近音词大多由相似字母组合构成, 源于同一词根, 表达的语义相近.

(3) 在横板文字编排习惯下, 部分左右结构汉字可以将偏旁部首拆分后独立形成汉字, 使语句语义发生变化, 但人类从左往右阅读时仍然能人工还原原始汉字, 例如图 1(c) 中“叶”字可以按左右拆分为“口”和“十”两个独体字, 拆分后语义发生变化而阅读时的视觉相似性较高. 而英文中由两个单词和连字符构成的复合词, 拆分后的语义仍然与原始文本相似.

(4) 汉语没有英文中的天然分词边界, 改变分词边界会显著影响文本的语义. 例如图 1(d) 中, 对“南京市长江大桥”插入标点, 可以形成不同语义的分词和断句.

本文从中文情感分类场景入手, 结合汉语上述语言特征, 提出了一种针对中文文本分类任务的字、词级别的对抗样本生成方法——CWordCheater, 通过相近字形、字音替换等方式保留了人类阅读文本内容的连贯性, 在黑盒条件下能够干扰模型分类结果. 同时引入基于 *USE* 编码的语义相似度约束, 降低对抗样本的语义偏移问题. 主

要贡献包括以下 4 点.

(1) 结合汉语的语言特征, 提出了一种涵盖了字音、字形、标点符号等特征的综合攻击策略, 对中文文本进行扰动攻击.

(2) 针对汉字象形文字的特点, 提出了一套基于图像特征的字词替换方法, 采用无监督方法构建了汉字形近字替换候选池, 进一步扩充了基于字形的对抗样本生成.

(3) 在对抗样本生成过程中, 嵌入了一种与被攻击模型无关的基于 *USE* 编码距离的语义约束方法, 有效避免了对抗样本的语义偏移问题.

(4) 构建了一套多维度的对抗样本评估方案, 从攻击效果和攻击代价两方面评估生成对抗样本的质量.

实验结果表明, 当 *USE* 语义约束阈值取 0.2 时, CWordCheater 在外卖评论等 3 个数据集上能使 4 种常见的文本分类模型准确率至少下降 27.9%, 同时在对抗样本的语义相似度和视觉相似度上达到 83% 和 86.7%, 扰动效果和扰动代价均优于两个对比的基线方法.

本文第 1 节梳理文本领域中对抗样本的研究工作. 第 2 节详细介绍对抗样本生成方案. 第 3 节展示实验方法、数据并分析实验结果. 最后总结全文.

1 相关工作

1.1 情感分类任务

情感分类任务被用广泛用于电商购物评价等场景, 通常句子级别的情感分类任务中会使用基于深度学习方法的文本分类模型, 主要包括基于 CNN 类、基于 RNN 类、基于注意力机制类和基于预训练语言模型等方法. 基于 CNN 类方法的代表是 TextCNN[4], 它将图像领域的卷积神经网络用于文本分类任务, 使用多个卷积核通过卷积操作提取文本特征; 基于 RNN 类方法的代表是长短记忆网络 (long short-term memory, LSTM)[5], 在文本分类中引入双向 LSTM 来提取句子中每个词语及其上下文特征, 以此提升对长文本序列的处理能力. 基于注意力机制类的方法中应用最广泛的是 Transformer 网络[6], 它使用注意力机制中的多头自注意力机制取代了传统的 RNN 编码结构, 根据词语之间的相似性得到各个词语之间权重矩阵, 对其进行注意力加权编码, 从而提升分类准确性; 目前最流行的预训练的语言模型是 BERT[7], 它在上游对输入的语句向量通过 Transformer 网络进行双向编码, 在下游任务中通过 Masked LM 等多任务联合学习更新网络参数, 得到的预训练模型仅需在具体任务的数据集上进行微调即可用于文本分类任务, 在训练数据较少时仍能保持良好的分类性能. 许多工作会在上述 4 类模型的基础上进行组合或变形, 得到改进后的分类方法.

同时 Xing 等人的研究指出基于深度学习的情感分类模型的鲁棒性较弱[8], 修改部分词语时会显著改变语句的情感倾向. 因此在情感分类任务中进行对抗攻击对于研究模型的鲁棒性具有一定价值, 本文将以上述 4 类常见的深度学习文本分类模型为对象, 研究对抗攻击效果.

1.2 文本对抗攻击

随着深度学习方法的广泛应用, 针对神经网络对抗性攻击和鲁棒性研究引起了广泛关注[9]. 对抗样本的研究兴起于图像领域, 这些工作通过改变图片中的少许像素点, 以微小的扰动代价成功干扰分类模型的输出[10]. 但是由于图像和文本数据的结构差异, 离散型的文本数据不适合沿用图像领域的对抗样本生成方式[11].

目前英文场景下的对抗样本多采用白盒和黑盒攻击相结合的攻击方式. Jia 等人在 QA 问答任务的白盒场景下引入句子级别的文本对抗攻击, 通过删除、修改原始语句和插入混淆语句的方式使得模型输出错误结果, 证明了文本分类模型脆弱的鲁棒性[12]. Paperno 等人从词语级别对文本分类模型进行攻击, 使用快速梯度标志法 (FGSM) 来找寻词语级别的对抗样本, 提出了插入、修改、删除 3 种攻击方式, 在黑盒和白盒场景下验证了基于梯度生成对抗样本的可行性[13]. Gao 等人提出了基于词语重要性的 DeepWordBug 算法, 在黑盒场景下通过模型输出结果设计出了词语重要性计算函数, 对关键词进行插入、删除、替换、交换顺序的扰动攻击[14]. Li 等人沿用了基于词语重要性攻击的思想, 通过在白盒场景下使用雅可比矩阵, 在黑盒场景下使用删除分数的方式提高了关键词筛选效率[15].

中文语言特征与英文存在较大差异, 汉字由偏旁和部首构成, 不能简单地使用英文任务中插入、替换、删除和

交换字母来扰动单词的方法; 与此同时, 汉字拥有象形文字和拼音等语言特征可以挖掘利用. Wang 等人设计了基于词语 TF-IDF 得分和删除分数的 WordHanding 算法来计算词语重要性, 使用同音词替换来生成对抗样本, 在 CNN 和 LSTM 网络上验证了该方法的有效性, 并通过 WMD 距离来评估对抗样本和原始样本的相似性[16]. Wang 等人提出了一种针对 BERT 语言模型的中文字符级别的攻击方法, 将离散文本映射到高维空间中, 依托于 BERT 强大的语言表征能力从高维空间中搜索返回语义最接近的字符[17]. Tong 等人针对汉字的语言知识提出了基于繁体字的形近字替换和汉字改写为拼音的替换方式, 对简单的深度学习模型具有良好的攻击效果[18]. Zeng 等人引入了基于字典的形近字替换方法, 生成具有视觉相似性的形近字对抗样本, 在影响模型输出结果的同时保留了读者阅读时的连贯性[19].

上述研究的攻击方法存在以下问题.

(1) 大量使用基于规则和字典的攻击方法, 攻击效果十分依赖人工先验知识和字典规则的质量. 例如对于形近字替换攻击, 字典中可能缺少独体字、冷门字、繁体字等替换组合.

(2) 攻击方法较为单一, 未充分结合汉字的表音和象形等语言特征. 例如没有考虑汉字可以按左右结构拆分以及标点符号可以改变分词边界的情况.

(3) 对生成的对抗样本的评估方法不够完善. 例如, 使用人工打分主观性较强且成本昂贵; 传统的编辑距离、WMD 距离等语句相似性指标难以反映中文场景下对抗样本的语义相似性和视觉相似性.

(4) 在生成对抗样本时没有考虑语义偏移问题. Michlel 等人[20]的研究指出, 大多数现有的文本攻击方案没有考虑到扰动前后输入的语义等价性, 对抗样本在改变模型分类结果的同时语义也发生了变化, 人类在阅读时难以理解其原始语义, 这样的对抗样本是无效的.

本文将针对前两点问题, 设计一套结合中文字形、读音、标点符号等特征的对抗样本生成策略. 针对第 3 点问题构建一套多维度的对抗样本评估方案, 从攻击效果和攻击代价两方面评估生成对抗样本的质量, 无需人工评分. 特别是针对第 4 点对抗样本的语义偏移问题, 深入研究了添加语义相似度约束的方法. 对于词级别的对抗样本, 现有研究主要通过使用同义词替换和添加 POS 词性检查器来保证语义相似性[21], 但这种词级别的语义约束容易导致对抗样本语法不通顺、语义不清晰. 另一种语义约束的思路是对整句话进行向量编码, 进而比较原始文本和对抗样本的语义向量距离. 目前主流的句编码方式有两种, 分别是 DAN 模型和 *USE* 模型[22]. DAN 模型通过句子中各个词语的词向量相加取平均的方式得到句向量, 这种方法的缺点是没有考虑到句子中各个词语对于语义的贡献度不同. 而 *USE* 模型使用基于多头注意力机制的 Transformer 模型对输入句子进行编码, 利用多任务联合学习训练获得词向量权重. 本文将设计一种与被攻击模型无关的基于 *USE* 的句编码网络对对抗样本进行语义偏移的约束.

2 对抗样本生成方案——CWordCheater

2.1 扰动方法

对抗样本生成过程就是扰动的过程, CWordCheater 扰动包括字词替换和标点扰动.

(1) 字词替换使用基于形近字、字形拆分、同音词构建的候选池来实现. 形近字候选池收录的是字形相近的汉字, 通过无监督训练将汉字的位图转换为字形的视觉向量, 进而构建形近字替换候选池; 字形拆分候选池收录的是可以按照左右结构将其拆分的汉字, 拆分后的汉字均为原始汉字的部首. 同音词候选池收录的是拼音相同的词语.

(2) 标点扰动规则是在重要性较高词语的内部加入标点符号, 从而扰乱其分词边界.

表 1 展示了一个对抗样本生成方式的样例.

表 1 对抗样本生成方式样例

处理方法		内容
原始文本		这家水果店的东西很**新xīn鲜(xiān)**
对抗样本生成方式	形近字替换	这家水果店的东西很**新薪(鲜)**
	字形拆分替换	这家水果店的东西很**亲斤(新)鱼羊(鲜)**
	同音词替换	这家水果店的东西很**心(xīn)弦(xián)**
	插入标点符号	这家水果店的东西很**新…鲜**

2.2 系统整体结构

CWordCheater 扰动的过程分为筛选待扰动词语、候选池搜索与扰动、对抗样本评估 (包括扰动效果检测和语义约束) 这 3 个阶段, 不同扰动方式采用的替换候选池不同, 如图 2 所示.

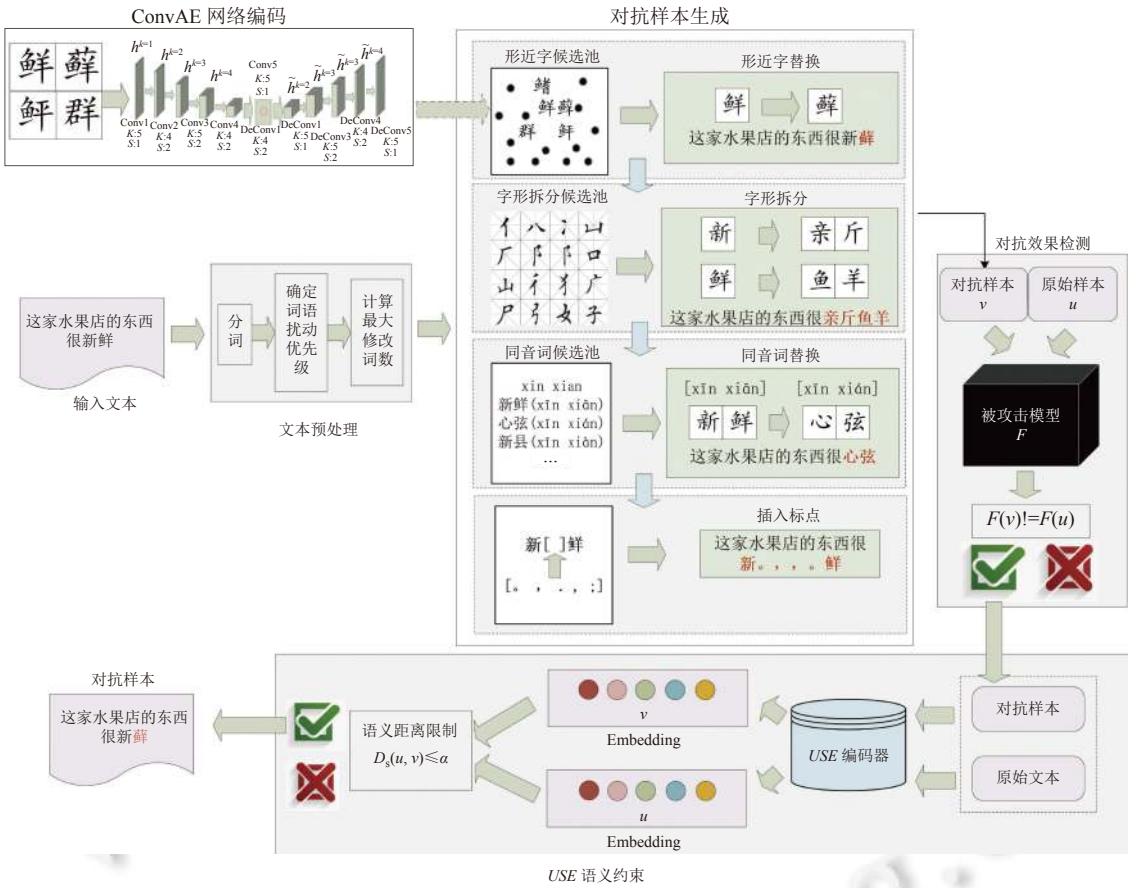


图 2 CWordCheater 系统结构

CWordCheater 生成对抗样本的过程描述如下.

(1) 根据输入文本的长度和句子个数确定最大可修改字数.

(2) 对输入文本进行分词处理, 使用被攻击模型计算各个输入词语的删除分数, 并根据删除分数表示该词语的扰动优先级.

(3) 从 4 种扰动方式中按照形近字替换-字形拆分-同音词替换-插入标点符号的优先级, 依次尝试迭代生成对抗样本.

① 形近字替换. 依次选择当前未被扰动的词语列表中重要性最高的词语, 从形近字替换候选池中按优先级尝试搜索并返回最优替换选项. 不断迭代直至成功改变模型分类结果, 或达到修改字数上限, 或遍历完所有待替换词语.

② 字形拆分. 从输入的各个汉字中, 尝试从字形拆分字典中搜索拆分组合, 不断迭代直至成功改变模型分类结果, 或达到修改字数上限, 或遍历完所有待替换的汉字.

③ 同音词替换. 依次选择当前未被扰动的词语列表中重要性最高的词语, 从同音词替换候选池中按优先级尝试搜索并返回最优替换选项. 不断迭代直至成功改变模型分类结果, 或达到修改字数上限, 或遍历完所有待替换

词语.

④ 标点扰动. 从重要性排名前 β 的词语中依次尝试标点扰动, 在词语内部随机插入标点符号. 若能改变模型分类结果则返回生成的对抗样本.

(4) 若对抗样本与原始文本的 *USE* 编码距离 $\leqslant \alpha$, 则保留生成的对抗样本, 否则丢弃.

2.3 基于词语重要性的扰动对象筛选方法

为了提高扰动效率, 并尽可能保留文本可读性, 需要用最小的搜索时间代价和最少修改内容来生成最优对抗样本, 因此优先攻击重要性较高的词语. 本文使用删除分数 (delete score, *DS*) 来衡量一个词语在文本分类任务中的重要程度, 并以此筛选和排序攻击候选词. 删除分数的定义和计算方式如下.

对输入样本 X_{input} 进行分词后得到序列 $X=[x_1,x_2,x_3,\ldots,x_n]$, 其中 n 表示词语个数和序列长度. 对于序列中的词语 x_i, 其删除分数为移除词语 x_i 前后模型 F 输出结果的差值, 如公式 (1) 所示.

$$DS(X_i)=F(x_1,\ldots,x_{i-1},x_i,x_{i+1},\ldots,x_n)-F(x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n) \tag{1}$$

本文使用删除分数作为序列 X_{input} 中各个词语的扰动优先级.

2.4 基于 *USE* 编码的语义扰动幅度限制

优秀的对抗样本应能在使得模型分类结果发生改变的同时, 保证人类阅读时能理解其原意. 现有对抗样本生成研究中大多忽视了语义偏移问题, 在扰动模型分类结果的同时使语句原意发生偏移, 导致读者难以理解文本原意. 目前已有的语义约束方法, 大多依赖于人工评判打分, 有一定主观性且成本昂贵.

为了实现对扰动前后语义偏差的约束, 本文提出一种基于注意力机制的 *USE* 句编码网络来计算其语义向量, 进而通过限制语义距离来限制扰动前后语义偏移的约束方法. *USE* 句编码模型如图 3 所示.

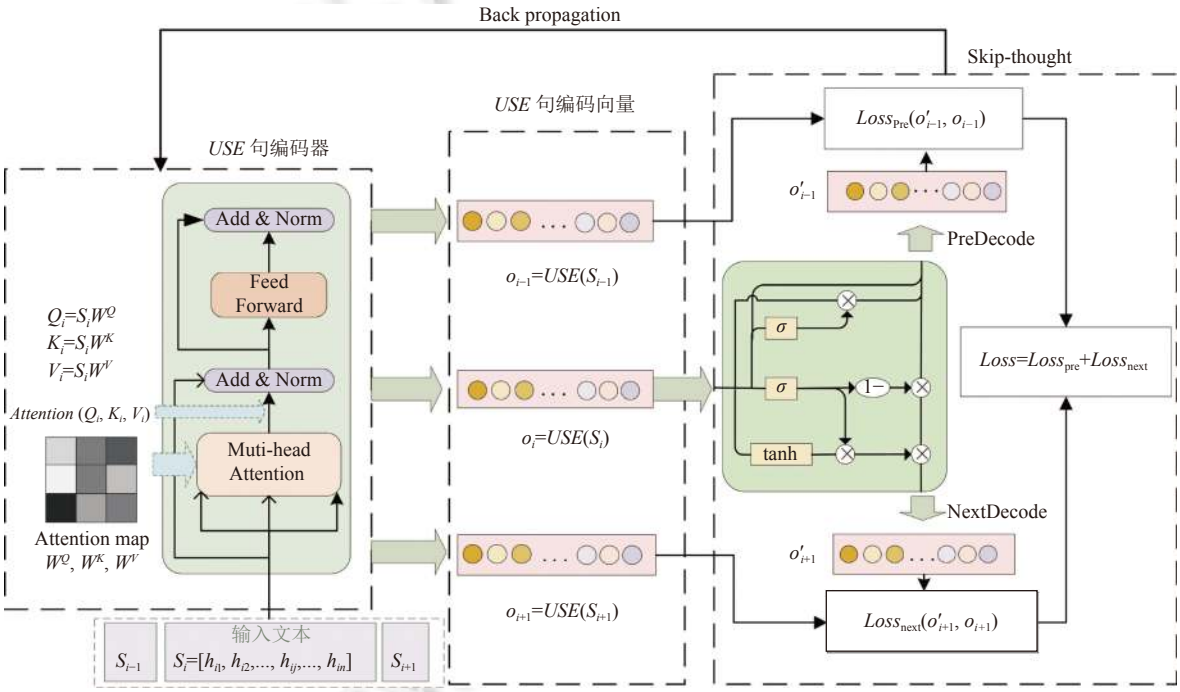


图 3 *USE* 句编码模型

USE 编码器使用 Transformer 网络的 Encoder 部分, 这是一种基于多头自注意力机制的文本编码模型, 即编码时句子中的每个词语对于整句话的语义贡献权重不同. 对于输入语句 $S_i=[h_{i1},h_{i2},\ldots,h_{ij},\ldots,h_{in}]$, h_{ij} 表示输入的各

个词语向量, 它由预训练的单词向量和当前词语在语句S_i中的位置向量共同构成. *USE* 编码器对S_i的编码过程可以用公式 (2) 来表示, 其中o_i表示 *USE* 编码器输出的句编码向量.

$$o_i = USE(S_i) \tag{2}$$

在对整句话编码前需要先对各个词语向量h_{ij}进行注意力编码, 第j个词语向量h_{ij}的注意力编码向量C_{ij}可以由句子中各个位置的词向量线性加权得到, 表示为公式 (3).

$$C_{ij} = \sum_{k=1}^{n} a_{ijk} h_{ik} \tag{3}$$

其中, a_{ijk}表示当前词语h_{ij}相对于句子中其他词语h_{ik}的权重参数. 语句S_i中各个词语向量h_{ij}的权重序列$A_{ij} = [a_{ij1}, a_{ij2}, \ldots, a_{ijk}, \ldots, a_{ijn}]$将通过 3 个权重矩阵$W^Q$、$W^K$、$W^V$得到, 这 3 个权重矩阵的参数将在下游任务中训练学习.

将S_i与W^Q、W^K、W^V相乘, 经过线性变换后得到 3 个注意力矩阵Q_i, K_i, V_i, 分别代表查询 (query)、键值 (key) 和取值 (value), 分别用公式 (4)–公式 (6) 表达.

$$Q_i = S_i W^Q \tag{4}$$

$$K_i = S_i W^K \tag{5}$$

$$V_i = S_i W^V \tag{6}$$

语句S_i的自注意力编码向量$Attention(Q_i, K_i, V_i)$计算为:

$$Attention(Q_i, K_i, V_i) = Softmax\left(\frac{Q_i K_i^{\mathrm{T}}}{\sqrt{d_k}}\right) V_i \tag{7}$$

其中, Q_i和K_i^{T}相乘得到的是句子中各个词语之间的权重矩阵, d_k表示词语向量的维度. 最后经过 *Softmax* 后得到归一化的权重矩阵, 权重矩阵的每一行表示词语向量h_{ij}的权重序列. 引入V_i的目的是增加一个可学习的权重矩阵W^V, 对输入的词向量序列经过参数调整后再进行加权.

在多头自注意力机制中会随机初始化多个W^Q、W^K、W^V, 从而得到多个 *Attention* 向量, 然后将多组 *Attention* 向量拼接后经过线性层变换得到合并后的自注意力编码向量, 随后经过残差连接、求和归一化和前馈网络得到新序列$S' = [h'_{i1}, h'_{i2}, \ldots, h'_{ij}, \ldots, h'_{in}]$, 将$S'_i$中各个词语向量相加后除以句子长度的平方根$\sqrt{n}$, 作为 *USE* 编码器的输出$o_i$, 如公式 (8) 所示.

$$o_i = \frac{\sum_{j=0}^{n} h'_{ij}}{\sqrt{n}} \tag{8}$$

模型使用 Skip-thought 算法训练词语权重参数, 原因包括以下两点.

(1) 本文目标是黑盒场景下的攻击, 希望得到一种与被攻击模型无关的通用型句向量表示方法. 而基于句子共现关系的 Skip-thought 算法无需获悉被攻击模型的参数和结构, 仅通过原始文本就可以调整 *USE* 编码的词语权重.

(2) Skip-thought 对训练样本的利用率较高. 当训练样本较少时, 通过拆分子句的方式能够高效利用少量样本来调整 *USE* 编码权重, 对噪声数据也有一定鲁棒性.

使用训练好的 *USE* 编码网络, 对输入的原始样本和生成的对抗样本进行整句编码, 利用二者的向量距离来衡量其语义相似度. 若原始文本的向量为u, 对抗样本的向量为v, 则二者的语义向量距离$D_s(u, v)$的计算方法为:

$$D_S(u, v) = \frac{\arccos\left(\dfrac{u \cdot v}{\|u\| \|v\|}\right)}{\pi} \tag{9}$$

根据系统需要, 仅采纳满足$D_S(u, v) \leqslant \alpha$的对抗样本.

2.5 候选池的构建

候选池包括形近字候选池、字形拆分候选池和同音词候选池 3 部分.

2.5.1 形近字候选池的构建

与英文字符级别替换的方式不同, 单个汉字不能直接进行偏旁和部首的修改, 由于形近字具有较高的视觉相似性, 在快速阅读时人类可以通过上下文较为准确地还原出形近字对应的原始汉字. 本文利用象形文字的特点, 从视觉特征来构建基于字形的形近字候选池. 使用无监督学习的 ConvAE 网络[23]来训练得到汉字视觉向量的空间嵌入, 进而得到汉字的形近字候选池. 图 4 展示了 ConvAE 网络的训练和通过汉字视觉向量得到形近字替换候选池的过程.

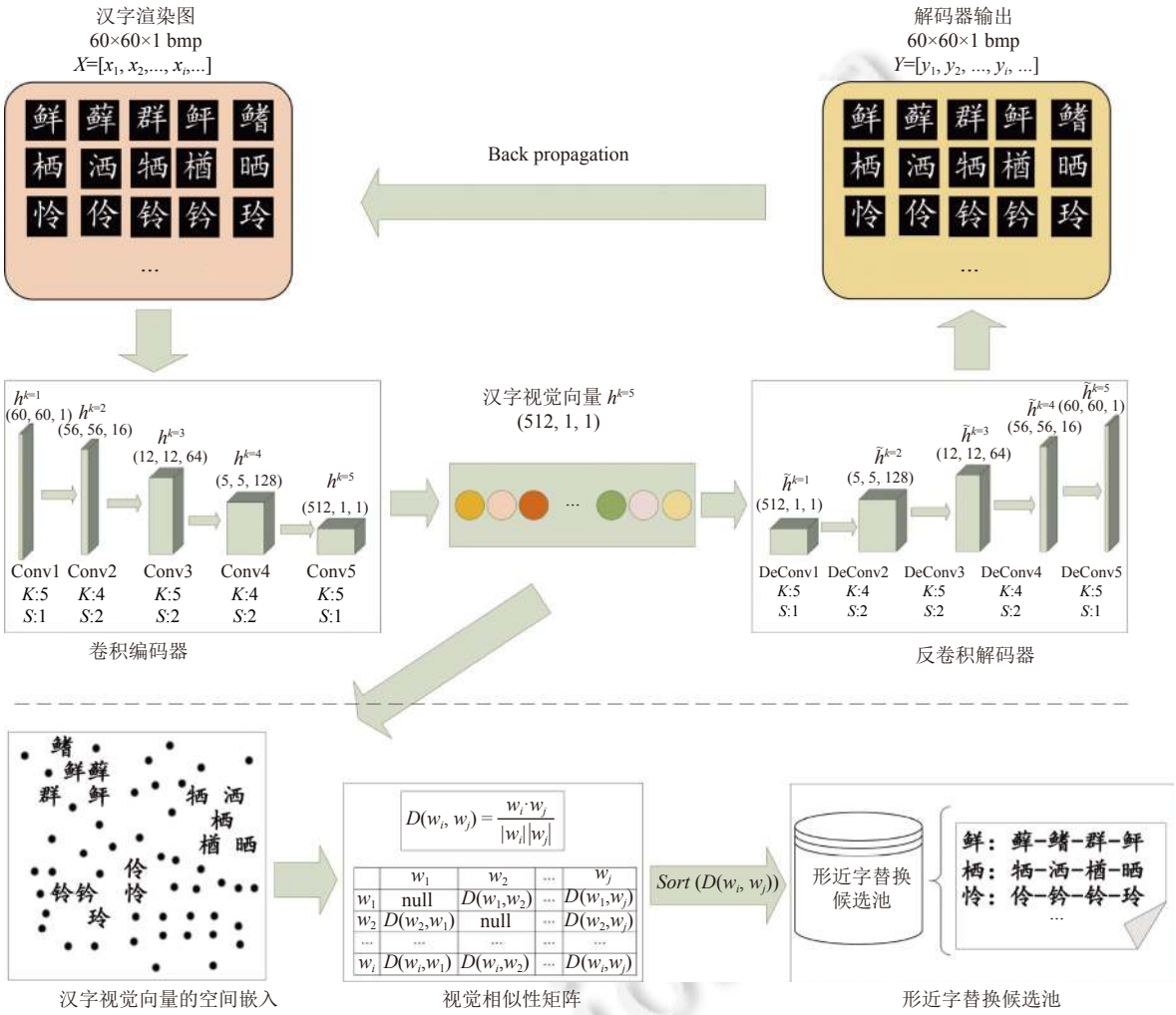


图 4 汉字视觉向量嵌入模型和候选池构建

ConvAE 是一种图像识别领域的自编码器, 用于降低渲染字符位图的尺寸并捕获高级视觉特征. Su 等人首先将其用于提取汉字的视觉信息, 以辅助词语相似度评价任务[24]. ConvAE 由对称的卷积编码器和卷积解码器构成, 包括 5 个不同步长和跳步的卷积核, 卷积层使用 Denoise autoencoder, 池化层使用 Max pooling. 接下来使用卷积编码器对输入的汉字位图 (60×60×8) 进行无监督编码, 将其输出的 512×1×1 尺寸的向量作为汉字的视觉信息. 图 4 中 K 表示卷积核的尺寸, 若 K=5, 则当前卷积核的长宽均为 5; S 表示卷积核的滑动步长 (注: GBK 标准收录了 GB2312 标准中的 6 763 个简体字和 BIG5 编码标准中的 13 053 个繁体字). ConvAE 模型描述如下.

对所有汉字位图 $X=[x_1,x_2,\ldots,x_i,\ldots]$ 依次输入到 ConvAE 网络中, 初始化 k 个卷积核 W 和对应的偏置 b. 以 x

为输入向量, W^k 表示第 k 个卷积参数, b^k 表示偏置, σ 表示激活函数, h^k 表示经过 k 次卷积操作后得到的汉字视觉向量, 则:

$$h^k = \sigma(x \times W^k + b^k) \tag{10}$$

随后对输入的 h^k 进行反卷积解码, y 表示经过 k 次反卷积操作得到的解码向量, 计算方法如公式 (11) 所示, 其中 $\tilde{W}^k$ 表示第 k 个反卷积核的参数, 同时也是公式 (10) 中 W^k 的转置矩阵, c 表示偏置常数.

$$y = \sigma\left(\sum_k h^k \times \tilde{W}^k + c\right) \tag{11}$$

使用随机梯度下降法更新权值, 计算过程用公式 (12) 表示.

$$\frac{\partial E(\theta)}{\partial W^k} = x \times \delta h^k + h^k \times \delta y \tag{12}$$

其中, δh^k 和 δy 表示隐藏状态和重建的增量.

对于每个汉字, 选取 top-N 个与其视觉距离最近的汉字加入该字的形近字替换候选池, 按相似度从高到低排序构建形近字替换的候选池.

与基于形近字字典进行替换的方式不同, GBK 编码可实现包括常用简体字、生僻字和繁体字的替换攻击, 而不依赖于现有的字典语料, 其案例如后文表 2 所示.

2.5.2 字形拆分候选池的构建

本文还提出了一种通过对汉字进行偏旁部首拆分, 将其替换为多个独立汉字的扰动方法. 针对现代汉语横向书写的特点, 从现代汉语词典中选取左右结构的汉字, 对其偏旁部首拆分, 若能将其拆分独立的单字, 则将拆分后的字序列加入拆字字典. 由于人类从左到右的阅读习惯, 这种拆分生成的对抗样本几乎不影响阅读体验, 读者能够快速还原文本并理解其语义. 字形拆分实例如表 3 所示.

表 2 通过形近字替换生成对抗样本的例子

原始文本	形近字对抗样本	形近字类型
黄焖鸡的酱汁咸淡正合**适**	黄焖鸡的酱汁咸淡正合**適(适)**	繁体字
他家奶茶真是**绝绝**子, YYDS!	他家奶茶真是**铯(绝)铯(绝)**子, YYDS!	冷门字

表 3 通过字形拆分生成对抗样本的例子

被拆分汉字	拆分组合
新	亲、斤
鲜	鱼、羊
吃	口、乞
骗	马、扁
…	…
嫩	女、敕

2.5.3 同音词候选池的构建

字音候选池由词典构建, 从海量文本语料中提取出词库, 按照拼音进行分类. 对于同一读音下的词语, 按照词频由高到低进行排序.

2.6 字词替换算法

形近字替换、字形拆分替换、同音词替换算法流程描述如算法 1.

算法 1. 字词替换算法.

输入: 文本序列 $X = [x_1, x_2, x_3, \ldots, x_n]$, 词语重要度向量 $DS = [DS_1, DS_2, DS_3, \ldots, DS_n]$, 各个词语的替换候选池 $\{P_{x_i}\} = [p_{i_1}, p_{i_2}, p_{i_3}, \ldots, p_{i_m}]$, 最大修改字数 e;

输出: 最优替换序列 X'.

1) 初始化 $X' = X$.

2) 根据词重要性序列 DS, 取出当前重要性分数最高的词语 x_i.

3) 从替换候选池 P_{x_i} 中依次取出优先级最高的替换词语 p_{ij}, 尝试进行替换, 并记录替换前后模型输出结果的差值 s_i. 最终得到候选池中每个候选词语替换后的交叉熵序列 $S_{x_i}=[s_{i_1},s_{i_2},s_{i_3},\ldots,s_{i_m}]$.

4) 根据序列 S_{x_i} 的得分, 选取使得模型分类结果差异 S_{x_i} 最大的 p_{i_s} 作为最优替换选项并返回.

5) 用 p_{i_s} 替换 x_i, 更新序列 X'.

6) 更新序列 DS, 将 DS_i 置为 0.

7) 若此时分类结果发生改变, 则返回序列 X', 对抗样本生成成功, 算法结束.

8) 若此时模型达到最大修改字数且分类结果未发生改变, 则对抗样本生成失败, 算法结束.

9) 若此时分类结果未发生改变, 且未达到最大修改字数, 则返回 2).

2.7 插入标点符号扰动

对于中文文本而言, 分词边界的准确性能够影响部分分类模型的准确率. 因此按照前文筛选出的词语优先级, 在较为重要的词语中插入标点符号, 使其分词边界发生变化, 可有效干扰模型分类结果. 其算法描述如算法 2.

算法 2. 标点符号扰动算法.

输入: 文本序列 $X=[x_1,x_2,x_3,\ldots,x_n]$, 词语重要度向量 $DS=[DS_1,DS_2,DS_3,\ldots,DS_n]$;
输出: 替换后的序列 X'.

1) 确定扰动标点符号的个数. 从 1 到 1/3 句子长度的区间内, 随机选择一个整数 γ, 作为扰动标点符号的个数.

2) 选择插入标点符号的位置. 对当前优先级最高的词语, 随机选择词语中相邻两个汉字之间作为插入位置.

3) 随机插入 γ 个标点符号, 包括句号、顿号、叹号、问号、反斜杠, 生成序列 X'.

4) 若成功改变模型分类结果, 则算法结束, 返回 X'. 否则算法结束, 生成失败.

2.8 组合攻击的优先级

脑认知和语言学的研究表明[25], 人类在阅读英文时会激活大脑中负责朗读的扇区, 主要通过字形-字音-字义的路径来理解文字语义. 而在阅读汉语时会激活大脑中负责书写的扇区, 主要通过字形-字义的路径来理解文字的语义. 因此可以认为基于字形的扰动在中文场景下具有较低的扰动代价, 更有利于人类阅读时还原其原始语义, 因此, 本文组合攻击优先级顺序为形近字替换——字形拆分——同音字替换——插入标点. 形近字替换和字形拆分都是基于字形的扰动方法, 它们生成的对抗样本具有较高的视觉相似性, 因此给予较高的优先级. 由于中文阅读对读音的依赖较轻, 但目前网络文本中仍然有使用同音词和谐音词来代替原始词语的现象, 所以给予同音词替换的优先级仍然高于插入标点的方法; 插入标点符号的方法虽然对于基础模型的攻击效果较为直接, 但这种方法规律型较强, 容易被正则表达式等方法过滤, 因此给予这种方法较低的优先级.

3 实验与结果分析

3.1 实验设置

本文分别对 TextCNN、LSTM、Transformer、BERT 这 4 种中文情感分类模型进行攻击, 来测试对抗样本生成方案的有效性.

数据集使用酒店点评、外卖评论、电商购物评价数据集, 每条数据有“好评”和“差评”两种标签, 对应分类时正样本和负样本. 3 个数据集的样本数量和分布如表 4 所示.

表 4 数据集中正负样本的分布

数据集名称	样本总数	正样本数量	负样本数量
酒店点评	10000	7500	2500
外卖评论	18000	12000	6000
电商购物评价	60000	30000	30000

3.2 评估指标

对于生成的对抗样本, 分别设计了损失代价指标和攻击效果指标来评估其质量. 其中攻击效果指标衡量的是对模型扰动的程度和效果, 包括准确率变化、语言模型困惑度; 损失代价指标衡量的是对原始文本的修改幅度, 包括字符修改比例、语义距离、视觉距离.

3.2.1 攻击效果指标

现有的工作大多使用准确率变化的单一指标来计算对抗样本攻击的效果. 这种方法的缺点是: 有时生成的对抗样本会使得模型输出较为均为的概率分布, 即模型认为这是一条语义偏中性或模棱两可的评论. 而仅使用准确率变化只能反映出模型分类结果的变化, 不能反映出对模型来说当前语句的语义混乱程度. 因此本文引入了基于语言模型的语句困惑度来完善对对抗样本攻击效果的评估.

困惑度常被用于衡量一个文本序列概率分布的混乱程度, 对于给定的句子 $S=[w_1,w_2,\ldots,w_n]$, 其困惑度 $PP(S)$ 表示为:

$$PP(S)=2^{-\frac{1}{n}\sum\log P(w_i|S)} \tag{13}$$

对于句子中的每个词语 w_i, 使用语言模型预测其概率分布, 经上述公式计算后用于表达语句的混乱程度. 可以认为, 语言模型困惑度越大, 则语句语义越混乱, 分类模型越会输出较为均匀的概率分布, 对抗样本的扰动效果越好. 因此本文使用基于语言模型的困惑度和分类准确率变化作为攻击效果指标.

3.2.2 攻击代价指标

现有的工作大多使用字符修改比例来衡量扰动幅度, 同时使用人工打分的方式来评估生成对抗样本在语法、语义层面的质量. 人工评估的方式较为主观, 且成本昂贵. 因此除字符修改比例外, 本文增加了语义相似度和视觉相似度这两种指标, 以更全面地衡量攻击代价. 攻击代价指标包括以下 3 种.

(1) 字符修改比例. 将插入、删除、替换的字符数量作为修改字符总数 m, 原始文本长度为 n, 则字符修改比例 δ 表示为:

$$\delta=\frac{m}{n} \tag{14}$$

(2) 语义相似度. 用于衡量生成的对抗样本和原始样本在语义上的差异. 采用 *USE* 编码相似度作为衡量指标, 若原始文本的 *USE* 句向量为 u, 生成对抗样本的句向量为 v, 二者的语义相似度 $sim_S(u,v)$ 可以用公式 (15) 来表示.

$$sim_S(u,v)=1-\frac{\arccos\left(\dfrac{u\cdot v}{\|u\|\|v\|}\right)}{\pi} \tag{15}$$

若语义相似度差异过大, 则生成的对抗样本可能已经发生了语义偏移, 读者在阅读时也无法理解语句原意.

(3) 视觉相似度. 对抗样本生成过程中采用的形近替换等方法虽然会改变模型的分类结果, 但人在阅读时仍然可以凭借视觉相似性还原其原始语义, 视觉相似性越高, 攻击代价越小. 本文使用 shape context 算法 [26]来衡量原始文本和对抗样本的视觉相似度, 该算法对变形字符识别具有较好的鲁棒性, 计算方式如下.

对于原始文本的图像 P 和 Q, 首先使用边缘提取和均匀采样的方法得到其形状的点集合 $I_P=\{p_1,p_2,\ldots,p_n\}$ 和 $I_q=\{q_1,q_2,\ldots,q_n\}$. 每个点包含的形状信息由所有其他点与之形成的相对向量构成. 使用形状直方图计算 P 和 Q 中任意两点 i 和 j 的匹配代价 $O_{i,j}$, 表示为:

$$O_{i,j}=M(p_i,q_j)=0.5\times\sum_{k=1}^{K}\frac{[g(k)-h(k)]^2}{g(k)+h(k)} \tag{16}$$

其中, K 表示直方图中颜色区间 (bin) 的个数, $g(k)$ 表示形状 P 的点 p_i 的形状直方图在第 k 个 bin 时的取值, $h(k)$ 表示形状 Q 的点 q_j 的形状直方图在第 k 个 bin 处的取值. 由此得到 P 和 Q 的代价矩阵 M. 随后基于代价矩阵, 对 P 和 Q 中的各个点进行匹配操作, 目标是使公式 (16) 取得最小值, 计算过程用公式 (17) 表示.

$$H(\pi)=\sum_i M(p_i,q_\pi(i)) \tag{17}$$

根据匹配操作, 对各个点之间的代价加权取平均, 得到二者形状距离 D_{sc}, 表示为:

$$D_{sc}(P,Q)=\frac{1}{n}\sum_i M_{\min}(p_i,q_\pi(i)) \tag{18}$$

进而计算原始文本和对抗样本的视觉相似度 $sim_v(P,Q)$, 表示为:

$$sim_v(P,Q)=1-D_{sc}(P,Q) \tag{19}$$

3.3 实验结果

3.3.1 模型有效性分析

为了检验本文提出的 CWordCheater 的有效性, 本文使用 WordHanding 方法[16]和 CWordAttacker[18]方法作为对比.

● WordHanding 方法使用词语的 TF-IDF 得分和删除分数计算其重要性, 并用同音词替换来生成对抗样本, 在黑盒场景下进行攻击.

● CWordAttacker 方法使用词语的删除分数计算其重要性, 并联合使用繁体字替换、拼音改写、词组拆解、词序扰动的攻击策略生成与原句语义一致的对抗样本.

分别对 3 个数据集和 4 种情感分类模型进行攻击. 其中 CWordCheater 的 *USE* 语义距离约束阈值 α 分别取 0.1、0.15 和 0.2. 表 5、表 6 和表 7 分别展示了 WordHanding、CWordAttacker 和 CWordCheater 在外卖评论、酒店评论和电商购物评论数据集上的攻击表现.

表 5 对外卖评论数据集的攻击效果和攻击代价

被攻击模型	攻击方法	攻击效果		攻击代价 (%)		
		模型分类准确率 (%)	模型困惑度	语义相似度	视觉相似度	修改比例
TextCNN	攻击前	90.1	23.5	Null	Null	Null
	WordHanding	70.8	32.1	85.6	89.6	13.5
	CWordAttacker	58.1	**46.7**	73.8	89	17.6
	CWordCheater α=0.1	60.6	34.5	**91.2**	91.7	**13.3**
	CWordCheater α=0.15	59.6	35.9	89.3	**91.9**	14.7
	CWordCheater α=0.2	**57.8**	37.3	86	91.2	14.5
LSTM	攻击前	91.1	23.5	Null	Null	Null
	WordHanding	72	30.7	86.1	87.9	**13.7**
	CWordAttacker	62.7	37.9	85.2	86.4	17.4
	CWordCheater α=0.1	62.9	35	**90.5**	**91.1**	14.6
	CWordCheater α=0.15	62.3	36.6	88.5	90.3	16.6
	CWordCheater α=0.2	**61.2**	**38.2**	86.2	91	16.4
Transformer	攻击前	95.2	23.5	Null	Null	Null
	WordHanding	74.1	31.2	85.8	87.5	**14.1**
	CWordAttacker	67	36.5	74.5	85.2	17.2
	CWordCheater α=0.1	65.4	37.8	**91.9**	**90.9**	15.8
	CWordCheater α=0.15	64.1	38.9	86.7	90	16.2
	CWordCheater α=0.2	**63.6**	**39.5**	86.4	89.7	16.5
BERT	攻击前	95.8	23.5	Null	Null	Null
	WordHanding	78.6	33.8	87.4	87.2	**14.6**
	CWordAttacker	71.5	38.2	77.3	84.4	17.5
	CWordCheater α=0.1	68.6	36.1	**91.5**	**90.5**	16
	CWordCheater α=0.15	67.4	37.6	88.2	89.8	16.3
	CWordCheater α=0.2	**66.2**	**38.4**	87.6	89.3	16.6

表 6 对酒店评论数据集的攻击效果和攻击代价

被攻击模型	攻击方法	攻击效果		扰动代价(%)		
		模型分类准确率 (%)	模型困惑度	语义相似度	视觉相似度	修改比例
TextCNN	攻击前	91.2	21.6	Null	Null	Null
	WordHanding	73.6	31.9	83.2	86.3	15.2
	CWordAttacker	60.6	40.8	76.1	89.7	16.9
	CWordCheater α=0.1	64.7	39.8	**91.9**	**92.1**	14.5
	CWordCheater α=0.15	62.1	40.3	90.4	91	14.9
	CWordCheater α=0.2	**58.7**	**44.2**	84.5	92.8	**13.8**
LSTM	攻击前	93.7	21.6	Null	Null	Null
	WordHanding	75.8	30.4	84.7	84.2	**14.6**
	CWordAttacker	63.7	46.9	80.7	85	16.2
	CWordCheater α=0.1	63.9	45.6	**91.5**	90.8	15.3
	CWordCheater α=0.15	63.4	47.8	86.2	**90.9**	16.3
	CWordCheater α=0.2	**61**	**50.3**	83.4	90.8	15.1
Transformer	攻击前	95.3	21.6	Null	Null	Null
	WordHanding	78.2	29.8	84.5	88.1	16.4
	CWordAttacker	71.3	41.6	72.2	85.2	17.2
	CWordCheater α=0.1	69.9	42.7	**90.6**	90.1	**16**
	CWordCheater α=0.15	65.3	49.8	86.5	**90.6**	16.7
	CWordCheater α=0.2	**64.8**	**53.1**	85.3	89.8	16.2
BERT	攻击前	95.5	21.6	Null	Null	Null
	WordHanding	78.4	31.2	85.9	88.6	**15.1**
	CWordAttacker	68.2	48.5	73.9	85.9	16.9
	CWordCheater α=0.1	71.3	45.8	**91.3**	**91**	15.8
	CWordCheater α=0.15	70.1	47.3	87.8	87.5	16.2
	CWordCheater α=0.2	**67.4**	**50.9**	86.2	86.7	16.5

由表 5、表 6 和表 7 有以下结论.

(1) 和攻击前的模型对比, CWordCheater 的攻击方法是有效的. 以 D_S 阈值 α 取 0.1 为例, 此时 4 个分类模型在外卖数据集上的分类平均准确率从 93.1% 下降到了 64.4%, 下降了 30.8%.

(2) CWordCheater 的攻击效果随着 D_S 阈值 α 的提升而愈发明显. 以电商数据集在 TextCNN 模型上的表现为例, D_S 阈值放宽 5%, 模型分类准确率下降 4%–5%.

(3) 当语义距离阈值 D_S 取 0.2 时, 在 3 个数据集和 4 种分类模型上, CWordCheater 相较于其他方法能取得更佳性能, 它能使 3 个模型分类准确率至少下降 27.9%. 以 TextCNN 模型在酒店数据集上的表现为例, WordHanding 只采用同音词替换攻击, 效果较差, 准确率为 73.6%; CWordAttacker 联合使用了多种扰动方法, 攻击后准确率降低为 60.6%, 但其扰动代价最大, 修改比例达 16.9%, 并且语义相似度和视觉相似度较低. CWordCheater 以 13.8% 的最小修改比例使得模型分类准确率下降为 58.6%, 下降了 35.6%, 并且语义相似度和视觉相似度分别高达 84.5% 和 92.8%.

(4) CWordCheater 对现在被广泛使用的 LSTM、Transformer 和 BERT 模型的攻击效果更佳. 对 Transformer 模型, 当 D_S 阈值取 0.1 时, 在 3 个数据集上均可以以最小扰动代价达到最佳扰动效果. 以 Transformer 模型在酒店数据集上的表现为例, CWordCheater 以 16% 的修改比例使得模型分类准确率下降为 69.9%, 下降 26.7%, 并且语义相似度和视觉相似度保持在 90% 以上, 高于 WordHanding 和 CWordAttacker. 对于 LSTM 模型, CWordCheater 在 D_S 阈值取 0.15 时, 其扰动效果和效率也高于 WordHanding 和 CWordAttacker 方法. 对于 BERT 模型, CWordCheater 在 D_S 阈值取 0.2 时, 其扰动效果和效率高于 WordHanding 和 CWordAttacker.

表 7 对电商购物评论数据集的攻击效果和攻击代价

被攻击模型	攻击方法	攻击效果		攻击代价 (%)		
		模型分类准确率 (%)	平均困惑度	语义相似度	视觉相似度	修改比例
TextCNN	攻击前	90.5	25.5	Null	Null	Null
	WordHanding	76.8	31.4	85.2	88.5	14.7
	CWordAttacker	65.4	54.2	79.8	87.8	15.1
	CWordCheater α=0.1	68.7	51.3	**91.7**	**91.3**	**12**
	CWordCheater α=0.15	64.7	55.8	89.4	90.3	14
	CWordCheater α=0.2	**59.4**	**57.8**	83.6	90	14.3
LSTM	攻击前	92.1	25.5	Null	Null	Null
	WordHanding	78.1	31.6	84.7	88.2	14.6
	CWordAttacker	68.3	56.3	78.2	87.1	14.9
	CWordCheater α=0.1	72.7	54.9	**91.4**	**90.8**	**13.2**
	CWordCheater α=0.15	67	58.2	88.3	89.5	15.6
	CWordCheater α=0.2	**64.5**	**59.6**	83.4	90.2	14.8
Transformer	攻击前	94.4	25.5	Null	Null	Null
	WordHanding	79.5	32.3	86.4	87.5	**14.3**
	CWordAttacker	72	51.9	74.2	86.6	16.2
	CWordCheater α=0.1	71.8	54.2	**91.2**	**90.3**	14.4
	CWordCheater α=0.15	69.1	55.3	87.7	89.2	16
	CWordCheater α=0.2	**67.3**	**56.1**	83.2	89.8	15.9
BERT	攻击前	94.8	25.5	Null	Null	Null
	WordHanding	79.7	33.9	86.8	88	**14.1**
	CWordAttacker	71.6	52.5	74.7	85.2	17.5
	CWordCheater α=0.1	72.4	53.8	**91**	**90.7**	14.6
	CWordCheater α=0.15	70.9	54.7	88.4	90.1	15.8
	CWordCheater α=0.2	**68.3**	**55.6**	84.3	89.6	16.3

3.3.2 语义距离约束对攻击效果的影响

一个好的攻击方法, 不仅能使模型分类准确率大幅下降、模型困惑度明显上升, 还应保证语义不会发生大幅度偏移. 下面这组实验将研究在不同 *USE* 距离阈值下攻击效果的变化. 由于 WordHanding 和 CWordAttacker 的方法没有考虑语义约束, 本文对其添加语义约束筛选, 分别取不同语义距离阈值 α, 即当生成的对抗样本和原始文本的语义距离 D_S 大于 α 时, 舍弃当前的对抗样本. 图 5 展示了对于 3 种数据集和 4 个被攻击模型, 分类准确率随 *USE* 语义距离阈值 α 的变化情况. 其中 α 的取值区间为 [0.1, 1], 当 $\alpha = 1$ 时表示没有语义距离限制.

由图 5 可见:

(1) 随着 *USE* 距离阈值 α 的增大, 语义约束条件被放宽, 3 种方法的攻击对模型分类准确率的扰动效果均有提升.

(2) 在相同 α 取值下, 3 种方法中 CWordCheater 的攻击成功率最高, 使得模型准确率下降幅度最大. 以外卖评论数据集上攻击 TextCNN 模型的实验为例: 在相同 α 取值下, 同 CWordAttacker 相比, CWordCheater 能使模型分类准确率至少多下降 5%; 同 WordHanding 相比, CWordCheater 能使模型分类准确率至少多下降 14%.

(3) 被攻击模型的分类准确率下降程度相同时, CWordCheater 生成的对抗样本和原始文本语义相似度更高. 以在酒店评论数据集上攻击 BERT 模型的实验为例, 当 $\alpha = 0.1$ 时, CWordCheater 攻击后模型分类准确率下降到 71.3%, 与 CWordAttacker 在 $\alpha = 0.5$ 时的攻击效果接近, WordHanding 攻击后的模型准确率始终高于二者. CWordAttacker 生成的对抗样本在语义上更接近于原始文本. 存在更少的语义偏移问题.

图 6 展示了对于 3 种数据集和 4 个被攻击模型, 模型困惑度随 *USE* 语义距离阈值 α 的变化情况. 其中 α 的取值区间为 [0.1, 1], 当 $\alpha = 1$ 时表示没有语义距离限制.

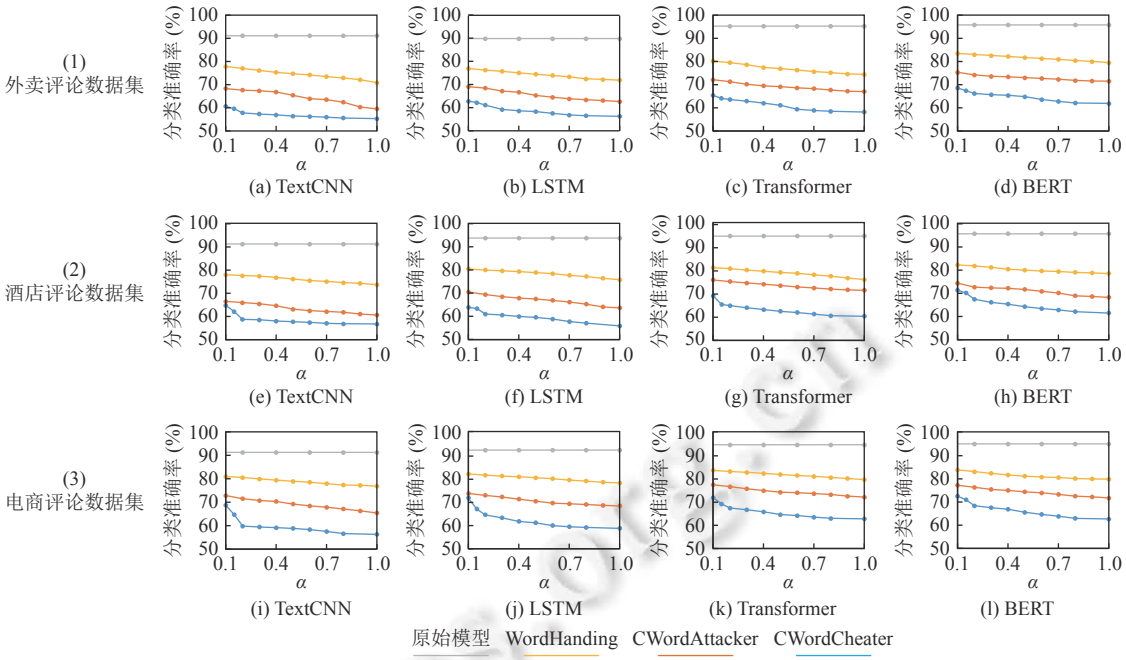


图 5 模型分类准确率随 *USE* 语义距离阈值 α 的变化趋势

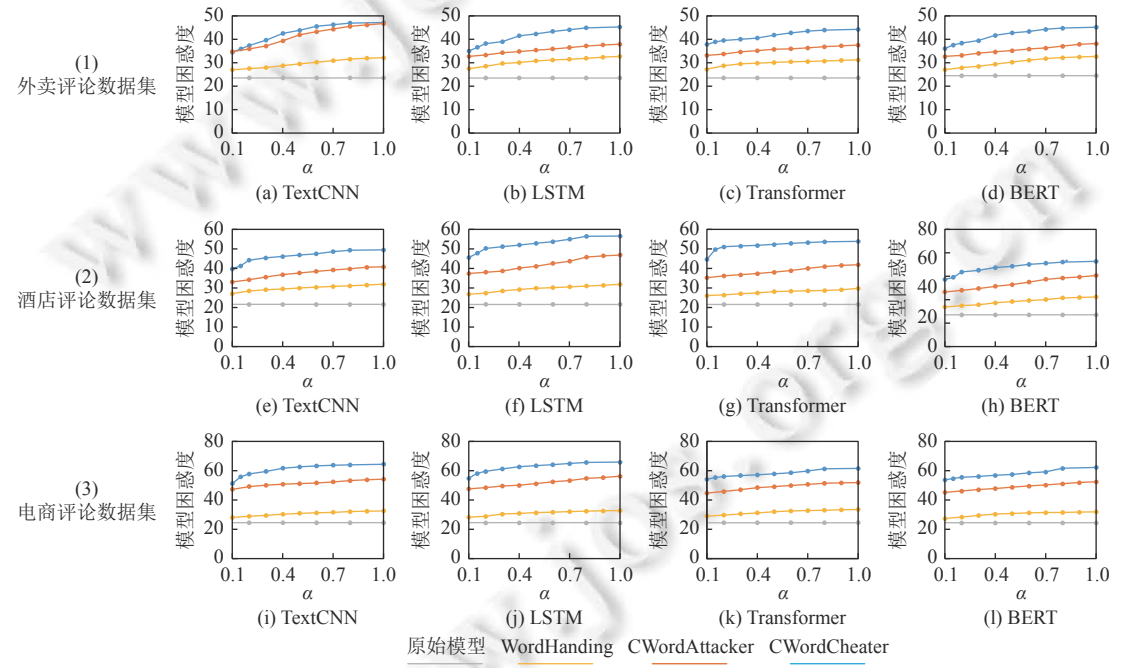


图 6 模型困惑度随 *USE* 语义距离阈值 α 的变化趋势

由图 6 可见:

(1) 随着 *USE* 距离阈值 α 的增长, 语义约束条件被放宽, 3 种方法攻击后模型困惑度均上升.

(2) 在相同 α 取值下, CWordCheater 攻击后模型困惑度最高, 被攻击模型越难以理解对抗样本的语义. 以在电商购物评论数据集上攻击 Transformer 模型的实验为例: 同 CWordAttacker 相比, CWordCheater 攻击后的模型困

惑度高出至少 8; 同 WordHanding 相比, CWordCheater 攻击后模型困惑度至少高出 30.

(3) 模型困惑度相同时, CWordCheater 在攻击时的 α 值更大. 以在酒店数据集中攻击 BERT 模型的实验为例: CWordCheater 在 $\alpha=0.1$ 时生成对抗样本的模型困惑度为 45.8, 与 CWordAttacker 在 $\alpha=0.5$ 时的表现接近, 而 WordHanding 在任何 α 取值下模型困惑度均低于二者. 此时 CWordCheater 生成的对抗样本与原始文本语义更接近, 存在更少的语义偏移问题.

3.3.3 各种攻击方式的效果和代价分析

本文统计了 4 种方法单独进行攻击时, 模型分类准确率的变化, 此时 *USE* 语义距离阈值 α 取 0.1. 以外卖数据集为例, 不同方法的攻击效果如图 7 所示. 其他数据集的表现与外卖数据集类似. 不同方法攻击的成功率如表 8 所示. 不同方法攻击时的视觉相似度代价如表 9 所示. 此外, 我们还统计了组合攻击时 4 种扰动方法在生成的对抗样本中所占的比重, 如表 10 所示.

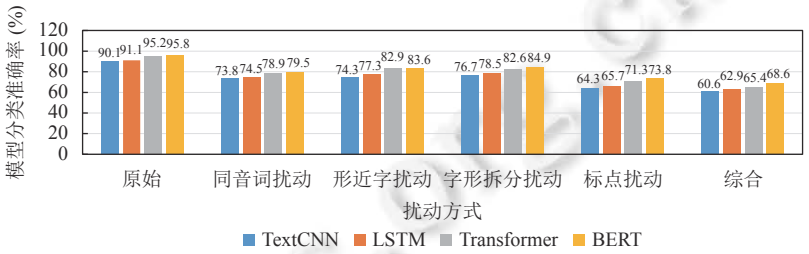


图 7 外卖数据集中不同扰动方法攻击效果的对比

表 8 不同攻击方法的成功率 (%)

数据集	模型	形近字扰动	字形拆分扰动	同音词扰动	标点扰动	综合
外卖	TextCNN	17.54	14.87	18.09	28.63	32.74
	LSTM	15.15	13.83	18.22	27.88	30.95
	Transformer	12.92	14.29	17.12	25.11	31.30
	BERT	12.73	11.38	17.01	22.96	28.39
酒店	TextCNN	15.57	12.72	17.65	26.64	29.06
	LSTM	16.22	14.09	18.57	30.52	31.80
	Transformer	15.22	14.59	16.89	24.03	26.65
	BERT	14.03	14.87	16.54	20.83	25.34
电商	TextCNN	13.48	11.82	17.79	21.99	24.09
	LSTM	13.79	12.92	17.81	19.54	22.15
	Transformer	15.04	12.61	16.74	20.97	23.94
	BERT	13.92	12.55	16.35	20.67	23.63

表 9 不同攻击方法的视觉相似度 (%)

数据集	模型	形近字扰动	字形拆分扰动	同音词扰动	标点扰动	综合
外卖	TextCNN	96.3	92.4	89.5	85.9	91.7
	LSTM	95.2	92.8	88.7	85.6	91.1
	Transformer	94.9	92.3	90.2	85.5	90.9
	BERT	94.4	91.8	89.6	85.2	90.5
酒店	TextCNN	95.6	93.1	90.4	86.1	92.1
	LSTM	95.4	92	88.6	85.2	90.8
	Transformer	94.8	91.9	89.2	84.3	90.1
	BERT	94.7	92.3	89.1	84.6	91
电商	TextCNN	95.1	92.6	89.7	85.8	91.3
	LSTM	95.3	92.2	87.9	84.3	90.8
	Transformer	95	92.1	88.5	84.9	90.3
	BERT	94.9	92.5	88.2	84.5	89.6

表 10　4 种扰动方法在生成的对抗样本中所占比重 (%)

数据集	模型	形近字扰动	字形拆分扰动	同音词扰动	标点扰动
外卖	TextCNN	59.5	8.3	12.5	19.7
	LSTM	53.7	12.7	13.8	19.8
	Transformer	43.4	18.5	19.4	18.7
	BERT	46.8	16.3	19.9	17.0
酒店	TextCNN	58.8	11.4	12.3	17.5
	LSTM	54.4	14.0	14.7	16.9
	Transformer	59.9	7.8	16.2	16.1
	BERT	58.0	9.6	16.7	15.7
电商	TextCNN	61.8	9.1	13.9	15.2
	LSTM	71.1	6.4	12.5	10.0
	Transformer	66.5	8.2	13.2	12.1
	BERT	56.0	11.6	18.1	14.3

由图 7、表 8、表 9 和表 10 的实验结果有如下结论.

(1) 从攻击成功率和模型分类准确率可以看出, 4 种方式的攻击效果排序为插入标点>同音词扰动>形近字扰动>字形拆分. 且 4 种方法单独攻击时均拥有 12% 以上的成功率, 标点符号单独攻击的成功率甚至超过 19%. 由于在攻击时我们同时考虑了字形、字音和标点符号 3 种语言特征, 综合攻击策略生成的对抗样本覆盖了更多类型, 所以它拥有更优的组合攻击效果.

(2) 从对抗样本的视觉相似度可以看出, 4 种方式的攻击代价排序为插入标点>同音词扰动>字形拆分>形近字扰动.

(3) 标点扰动的成功率最高, 视觉相似度最低, 扰动代价最高. 以在外卖数据集中攻击 TextCNN 模型的实验为例, 标点扰动生成的对抗样本与原始文本的视觉相似度仅有 85.9%. 同时该方法规律性较强, 容易被规则过滤. 应给予最低的优先级, 保证扰动效果并降低扰动代价.

(4) 同音词扰动的成功率仅次于标点扰动, 其扰动代价仅低于标点扰动. 汉语中有大量异意的同音词, 所以同音词候选池中拥有大量优质替换选项, 导致扰动成功率较高. 而同音词之间字形差异通常较大, 在阅读时需借助读音来还原语义, 不如基于字形的扰动方式直观, 应给予较低的优先级, 以降低扰动代价.

(5) 基于字形的两种扰动方式中, 形近字替换的成功率与同音词扰动的成功率接近, 但扰动代价显著低于后者, 证明形近字替换候选池是有效的. 例如在酒店数据集上攻击 Transformer 模型的实验中, 形近字扰动的成功率为 15.22%, 与同音词扰动 16.89% 的成功率接近. 但形近字扰动的视觉相似度为 94.8%, 明显优于字音扰动 89.2% 的视觉相似度. 为了优化扰动代价, 应当给予形近字扰动最高的优先级. 而字形拆分的扰动成功率较低, 这是因为汉字中能够按照左右结构进行拆分的样本数量较少, 可以将字形拆分作为形近字替换的补充, 给予次高的优先级.

(6) 从 4 种扰动方法在生成的对抗样本中所占比重可以看出, 4 种方法中形近字替换的比重最高, 在所有模型和数据集上占比均高于 40%; 同音词扰动和标点扰动次之; 字形拆分的比重最低, 在所有模型和数据集上占比均低于 20%. 经过优先级调整后的攻击策略, 在攻击时会优先尝试扰动代价较低的生成方式. 以在电商评论数据集上攻击 BERT 模型为例, 单独使用插入标点符号攻击时的视觉相似度仅有 84.5%, 扰动代价较高, 而它在组合攻击策略中占比仅有 14.3%; 而在两种基于字形的攻击方式单独攻击时, 形近字扰动和字形拆分的视觉相似度分别高达 94.9% 和 92.5%, 并且两种方法在组合攻击时合计占比高达 67.6%; 4 种方法组合后视觉相似度高达 89.6%, 高于 WordHanding 的 88% 和 CwordAttacker 的 85.2%. 这说明我们的攻击策略设计的优先级是合理的.

综上, 由实验结果可得, 这 4 种攻击方式都是有效的, 同时 CWordCheater 对攻击优先级策略的设计也是合理的.

4 总　结

本文面向中文情感分类任务, 针对中文的意音、象形等语言特征, 提出了一套综合攻击策略, 以更小的扰动代

价获得了更优的扰动效果. 提出了一种与被攻击模型无关的基于 *USE* 的句编码语义约束方法, 有效避免了对抗样本的语义偏移问题. 提出了包含语言模型困惑度和模型准确率的攻击效果指标和包含语义相似度、视觉相似度、修改比例的攻击代价指标, 综合评估了生成的对抗样本质量.

CWordCheater 目前仅适用于纯中文场景, 还没有深入挖掘中英文混杂或拼音缩写、简写、表情符号等攻击方法. 在未来的工作中我们将针对上述场景, 进一步扩展对抗样本的生成策略, 同时研究将生成的对抗样本合理加入训练数据中, 通过对抗训练提升分类模型的鲁棒性.

References:

[1] Ilyas A, Santurkar S, Tsipras D, Engstrom L, Tran B, Madry A. Adversarial examples are not bugs, they are features. In: Proc. of the 33rd Int'l Conf. on Neural Information Processing Systems. Vancouver: Curran Associates Inc., 2019. 12.

[2] Ruan WJ, Yi XP, Huang XW. Adversarial robustness of deep learning: Theory, algorithms, and applications. In: Proc. of the 30th ACM Int'l Conf. on Information & Knowledge Management. ACM, 2021. 4866–4869. [doi: 10.1145/3459637.3482029]

[3] Xu H, Ma Y, Liu HC, Deb D, Liu H, Tang JL, Jain AK. Adversarial attacks and defenses in images, graphs and text: A review. Int'l Journal of Automation and Computing, 2020, 17(2): 151–178. [doi: 10.1007/s11633-019-1211-x]

[4] Kim Y. Convolutional neural networks for sentence classification. In: Proc. of the 2014 Conf. on Empirical Methods in Natural Language Processing (EMNLP). Doha: Association for Computational Linguistics, 2014. 1746–1751. [doi: 10.3115/v1/D14-1181]

[5] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Computation, 1997, 9(8): 1735–1780. [doi: 10.1162/neco.1997.9.8.1735]

[6] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I. Attention is all you need. In: Proc. of the 31st Int'l Conf. on Neural Information Processing Systems. Long Beach: Curran Associates Inc., 2017. 6000–6010.

[7] 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. Minneapolis: Association for Computational Linguistics, 2019. 4171–4186. [doi: 10.18653/v1/N19-1423]

[8] Xing XY, Jin ZJ, Jin D, Wang BN, Zhang Q, Huang XJ. Tasty burgers, soggy fries: Probing aspect robustness in aspect-based sentiment analysis. In: Proc. of the 2020 Conf. on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, 2020. 3594–3605. [doi: 10.18653/v1/2020.emnlp-main.292]

[9] Chaubey A, Agrawal N, Barnwal K, Guliani KK, Mehta P. Universal adversarial perturbations: A survey. arXiv: 2005.08087, 2020.

[10] Yu YR, Gao XT, Xu CZ. LAFEAT: Piercing through adversarial defenses with latent features. In: Proc. of the 2021 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Nashville: IEEE, 2021. 5731–5741. [doi: 10.1109/CVPR46437.2021.00568]

[11] Wang WQ, Wang R, Wang LN, Wang ZB, Ye AS. Towards a robust deep neural network in texts: A survey. arXiv:1902.07285, 2019.

[12] Jia R, Liang P. Adversarial examples for evaluating reading comprehension systems. In: Proc. of the 2017 Conf. on Empirical Methods in Natural Language Processing. Copenhagen: Association for Computational Linguistics, 2017. 2021–2031.

[13] Papernot N, McDaniel P, Jha S, Fredrikson M, Celik ZB, Swami A. The limitations of deep learning in adversarial settings. In: Proc. of the 2016 IEEE European Symp. on Security and Privacy (EuroS&P). Saarbruecken: IEEE, 2016. 372–387. [doi: 10.1109/EuroSP.2016.36]

[14] 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]

[15] 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.

[16] 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]

[17] Wang BX, Pan BY, Li X, Li B. Towards evaluating the robustness of Chinese BERT classifiers. arXiv:2004.03742, 2020.

[18] 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]

[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. In: Proc. of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th Int'l Joint Conf. on Natural Language Processing: System Demonstrations. Association for Computational Linguistics, 2021. 363–371. [doi: 10.18653/v1/2021.acl-demo.43]

[20] Michel P, Li X, Neubig G, Pino J. On evaluation of adversarial perturbations for sequence-to-sequence models. In: Proc. of the 2019

Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Vol. 1. Minneapolis: Association for Computational Linguistics, 2019. 3103–3114. [doi: 10.18653/v1/N19-1314]

[21] Jin D, Jin ZJ, Zhou JT, Szolovits P. Is BERT really robust? A strong baseline for natural language attack on text classification and entailment. Proc. of the AAAI Conf. on Artificial Intelligence, 2020, 34(5): 8018–8025. [doi: 10.1609/aaai.v34i05.6311]

[22] Cer D, Yang YF, Kong SY, Hua N, Limtiaco N, John RS, Constant N, Guajardo-Cespedes M, Yuan S, Tar C, Strope B, Kurzweil R. Universal sentence encoder for English. In: Proc. of the 2018 Conf. on Empirical Methods in Natural Language Processing: System Demonstrations. Brussels: Association for Computational Linguistics, 2018. 169–174. [doi: 10.18653/v1/D18-2029]

[23] Masci J, Meier U, Cireşan D, Schmidhuber J. Stacked convolutional auto-encoders for hierarchical feature extraction. In: Proc. of the 21st Int'l Conf. on Artificial Neural Networks. Espoo: Springer, 2011. 52–59. [doi: 10.1007/978-3-642-21735-7_7]

[24] Su TR, Lee HY. Learning Chinese word representations from glyphs of characters. In: Proc. of the 2017 Conf. on Empirical Methods in Natural Language Processing. Copenhagen: Association for Computational Linguistics, 2017. 264–273. [doi: 10.18653/v1/D17-1025]

[25] Tan LH, Spinks JA, Eden GF, Perfetti CA, Siok WT. Reading depends on writing, in Chinese. Proc. of the National Academy of Sciences of the United States of America, 2005, 102(24): 8781–8785. [doi: 10.1073/pnas.0503523102]

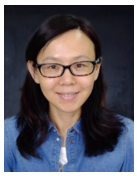
[26] Mori G, Belongie S, Malik J. Efficient shape matching using shape contexts. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2005, 27(11): 1832–1837. [doi: 10.1109/TPAMI.2005.220]

附中文参考文献:

[16] 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]

[18] 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]

李相葛(1995－), 男, 博士生, 主要研究领域为自然语言处理.



孙岩(1970－), 女, 博士, 教授, 博士生导师, CCF 高级会员, 主要研究领域为物联网, 区块链, 大数据分析与挖掘.

罗红(1968－), 女, 博士, 教授, 博士生导师, CCF 高级会员, 主要研究领域为物联网大数据智能分析, 自然语言处理.