

文本中的对抗攻击与防御综述

梁宇航^{1,2}, 林政¹, 王雷¹, 何原野^{1,2}, 王伟平¹

¹中国科学院信息工程研究所 北京 中国 100093

²中国科学院大学网络空间安全学院 北京 中国 100049

摘要 随着人工智能技术与高性能计算设备的发展, 近些年深度神经网络(Deep Neural Networks, DNNs)在计算机视觉、语音识别和自然语言处理等领域取得了非常不错的成就。在图像分类、语音识别和文本分类等任务中, DNNs 的准确率甚至超越了人类, 为人工智能赋予了惊人的能力。然而, 近几年的研究表明, DNN 模型非常容易受到对抗样本的攻击, 只需在正常输入中加入微小不可察觉的扰动, 就能导致 DNN 模型错误的预测, 这一现象引发了人们对模型鲁棒性的深刻思考。在计算机视觉领域中对对抗攻击与防御已经得到了广泛的研究, 研究者们努力开发鲁棒的模型来抵御各种攻击, 但在文本领域中的研究还有些不够, 很多视觉领域的方法并不能直接应用于文本, 尤其是文本离散的特点使得攻击和防御更有挑战性, 也有更多的研究空间。本文全面介绍了文本领域中的对抗攻击与防御以及一些相关工作。具体来说, 本文首先从不同的角度对文本中的对抗攻击与防御进行了分类, 以便更好地理解和研究不同类型的攻击与防御, 接着本文介绍了相应的工作和最新进展, 包括不同攻击与防御方法的性能比较以及实验结果的分析, 最后本文讨论了文本领域对抗攻击与防御存在的挑战, 如何在保持语义准确性的前提下生成质量更高的对抗样本以及如何避免模型只学习浅层特征。这些挑战为研究者提供了宝贵的探索机会, 未来的研究可以集中在这些方向以推动文本领域对抗攻击与防御的发展。

关键词 深度神经网络; 对抗样本; 对抗攻击与防御

中图法分类号 TP309.2 DOI 号 10.19363/J.cnki.cn10-1380/tn.2023.08.11

A Survey on Adversarial Attacks and Defenses in Text Domain

LIANG Yuhang^{1,2}, LIN Zheng¹, WANG Lei¹, HE Yuanye^{1,2}, WANG Weiping¹

¹ Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China

² School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China

Abstract With the development of artificial intelligence and high computational devices, deep neural networks (DNNs), in recent years, have achieved remarkable results in fields such as computer vision, speech recognition, and natural language processing. In tasks such as image classification, speech recognition, and text classification, the accuracy rate of DNNs has even surpassed that of humans, endowing artificial intelligence with amazing capabilities. However, researches in recent years have shown that DNNs are highly vulnerable to adversarial examples which can lead to incorrect predictions by adding small and imperceptible perturbations to the normal inputs. This phenomenon has triggered people's deep thinking about the robustness of DNN models. The adversarial attacks and defenses have been well studied in the field of computer vision, and researchers strive to develop robust models against various attacks, but researches in text domain are still relatively insufficient. Many methods in computer vision domain cannot be directly applied to texts. Especially the input space of texts is discrete which makes attacks and defenses more challenging. So there is still lots of research potentials in this field. This article presents a comprehensive introduction of adversarial attacks and defenses in text domain together with some related work. Specifically, we first classify the adversarial attacks and defenses in texts from different perspectives so that we can better understand and study different types of attacks and defenses. Then we present the corresponding related works and recent advances, including the performance comparison of different attack and defense methods and the analysis of experimental results. Finally we discuss the existing challenges of adversarial attacks and defenses in text domain such as how to generate higher-quality adversarial examples while maintaining semantic similarity and how to avoid the model from only learning shallow features. These challenges provide researchers with valuable exploration opportunities, and future research can focus on these directions to promote the development of adversarial attacks and defenses in text domain.

Key words deep neural networks; adversarial examples; adversarial attacks and defenses

通讯作者: 林政, 博士, 研究员, Email: linzheng@iie.ac.cn。

本课题得到国家自然科学基金项目(No. 61976207, No. 61906187)资助。

收稿日期: 2021-01-29; 修改日期: 2021-04-26; 定稿日期: 2023-08-18

1 引言

随着人工智能与深度学习技术的飞速发展, DNNs 在解决很多任务上显示出了强大的威力, 也因此很多基于 DNN 的系统被广泛地应用在现实世界中, 如人脸识别、语音识别、客服聊天机器人等。然而, Szegedy 等人^[1]的研究却发现一个令人警惕的事情——只需对输入图像样本添加微小不可察觉的扰动就可以很轻易使 DNNs 分类出错, 许多工作相继揭示了这一现象^[2-4]。这些被刻意修改后的输入称为对抗样本, 它们并不会影响人类的判断, 却能使 DNNs 出错。对抗样本的发现给基于 DNN 的系统带来了潜在的安全威胁, 例如在自动驾驶中, 如果系统的决定能够被一个微小的扰动左右是非常可怕的。因此, 近年来对抗攻击成为深度学习和安全领域一个非常重要的问题, 针对攻击的防御技术也相应诞生, 相关的研究越来越多。

在自然语言处理领域, DNNs 被广泛应用于情感分析、机器翻译和阅读理解等基本任务, 但是这些基于 DNN 的系统在面对对抗样本的攻击时也会出现不可思议的错误。Papernot 等人^[5]在图像领域对抗样本的基础上发现, 攻击者可以通过修改输入中的单词生成对抗样本, 并导致 DNNs 分类错误。Jia 等人^[6]在问答系统中发现, 在一篇文章的末尾附加一个句子会使问答系统表现降低, 甚至直接对输入问题进行语义等价的更改也会导致模型返回错误的结果。文本中的对抗样本同样带来了一些关于系统性能的担忧, 这些潜在的安全问题引起了研究者的广泛关注, 大量的科研工作旨在探索什么是对抗样本、对抗样本如何影响 DNN 模型的行为以及如何防御对抗样本。目前在深度学习和计算机视觉领域已经有很多关于对抗攻击与防御的综述性研究^[7-11], 文本领域也相应有一些比较优秀的调研^[12-15], 但是尽管如此, 整体来看文本领域对抗攻击与防御的研究仍然非常有限, 它应该被给予更多的关注。

2 对抗攻击与防御基本理论

在这一部分, 将对对抗攻击与防御基本理论做介绍。具体的, 首先将介绍对抗样本的概念, 接着介绍相应的攻击方法和防御方法分类, 最后介绍对抗攻击中的扰动控制。图 1 给出了文本领域对抗攻击与防御方法的基本分类。

2.1 对抗样本

一个 DNN 分类器可以被描述为 $f: x \rightarrow y$, 其中

f 可以是任意一种分类器, 如循环神经网络^[16-17]和卷积神经网络^[18]等, $x \in X = \{x_1, x_2, \dots, x_N\}$ 是样本集 X 中的一个样本, $y \in C = \{c_1, c_2, \dots, c_K\}$ 是样本 x 所属类别。对于输入空间中的样本 x , 有确定的真实类别 y_{real} 与之对应。如果模型的预测输出 $f(x) = y = y_{real}$, 则称模型对样本分类正确。在对抗攻击场景下, 攻击者的目标就是在输入 x 上添加微小扰动 ϵ , 并且 $\|\epsilon\| \leq \delta$, 但 $f(x + \epsilon) = y' \neq y$, 也就是添加扰动后的样本要使得分类器分类出错, 这样的 $x' = x + \epsilon$ 就是对抗样本, 其中 δ 是扰动幅度限制。值得注意的是, δ 一定要足够小, 使得 x 和 x' 对人类而言几乎不可区分, 这意味着一个好的对抗样本不仅能够使模型出错, 而且对人类来说这个改变也是不可察觉的。而一个鲁棒的分类器则要求对任意添加的微小扰动 ϵ , 只要扰动幅度 $\|\epsilon\| \leq \delta$, 都有 $f(x + \epsilon) = y' = y$, 即对抗攻击并不能使模型预测结果发生改变。

对抗样本通常产生在不同类别决策边界附近的易混淆区域, 训练样本数量有限和分布有限很难覆盖这一区域, 导致该区域的样本容易被错误分类。现在关于对抗样本的存在有两个主要的观点, Goodfellow 等人^[2]认为是神经网络的线性性质, 他们声称现有 DNN 模型是一个巨大的线性模型, 虽然引入了非线性的激活函数, 但这些激活函数接近分段线性, 线性结构分类器生成的决策边界极易产生对抗样本, 尤其是在高维空间。作者还给出了一种基于梯度的攻击方法快速梯度符号法(Fast Gradient Sign Method, FGSM), 其本质就是一种线性攻击。对抗样本的另一个解释是数据的非鲁棒性特征生成的样本, 该观点由 Ilyas 等人^[19]提出, 他们声称对抗样本并不是奇怪的错误, 而是有助于模型泛化的数据分布特征。作者进一步指出, 在分类任务中, 样本特征可以分为鲁棒性和非鲁棒性特征, 它们都是用来预测真实标签的, 但是在非鲁棒特征上加入微小扰动, 可使得模型最终的预测出错, 而鲁棒性特征几乎不受扰动影响。鲁棒性可以作为数据集本身的属性, 当剔除了数据集中的非鲁棒性特征时, 仅仅通过标准训练就能获得高鲁棒性的模型, 而对抗训练等防御方法可以视为通过特征变换来减弱非鲁棒性特征的预测性, 从而引导模型不再依赖非鲁棒性特征。这一观点下, 对抗样本的迁移性也得到了解释, 不经过对抗训练的模型几乎都会被对抗样本攻击成功就是因为模型并不能很好地处理非鲁棒性特征。

2.2 文本对抗攻击方法分类

文本对抗攻击是在文本中加入微小扰动并使 DNN 模型产生错误预测的过程。由于攻击方法的多

样性, 可以从不同角度对攻击方法进行分类, 文本的离散特性也使得文本对抗攻击的分类更加复杂。

图 1 展示了文本对抗攻击与防御的基本分类。

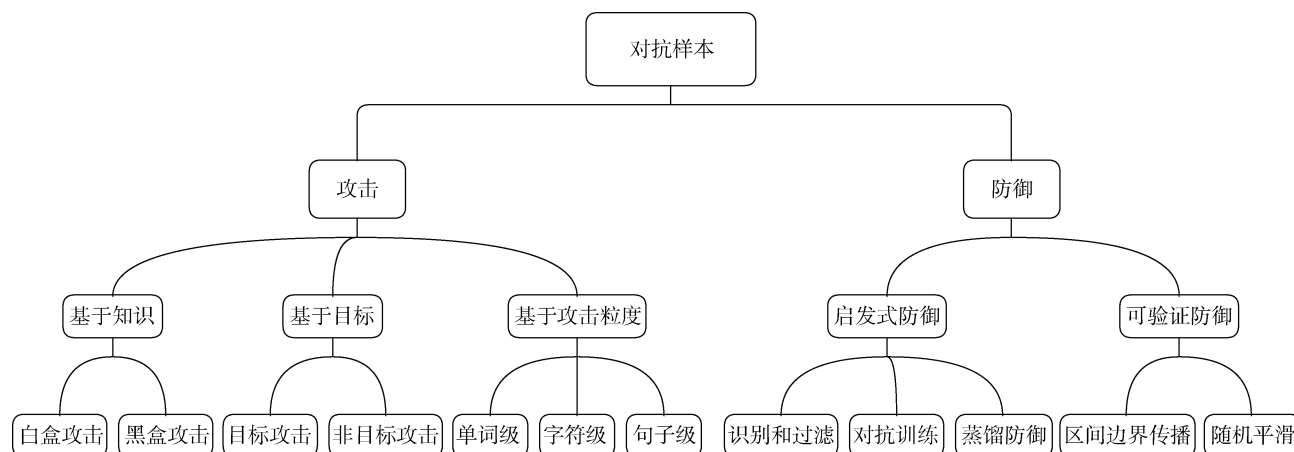


图 1 对抗攻击与防御分类

Figure 1 The classification of adversarial attacks and defenses

2.2.1 基于知识

根据攻击者对目标模型掌握信息的不同可以分为白盒攻击和黑盒攻击。白盒攻击是最理想的攻击情形, 在白盒场景下, 攻击者拥有目标模型的所有信息, 包括模型结构、参数权重和梯度, 因此攻击者往往可以利用模型梯度信息, 使用梯度上升的优化方法来调整扰动生成对抗样本。和白盒攻击相对的是黑盒攻击, 在黑盒场景中, 攻击者对目标模型几乎没有了解, 并不知道模型的结构、参数权重和梯度等信息, 不过却可以查询目标模型来得到输入样本对应的输出结果, 由于对抗样本有迁移性^[20-21], 可以训练替代模型进行攻击。根据模型输出结果的类型, 黑盒攻击可以细分为基于分数和基于决定的攻击, 基于分数可以得知目标模型最终的分类输出概率, 而基于决定只能知道目标模型给出的分类结果, 实际中通常根据模型输出的变化值找到对分类贡献最大的词, 之后针对该词进行攻击。此外, 攻击者甚至可能无法访问目标模型, 这种攻击被称为盲攻击。总体来说, 白盒攻击由于拥有更多关于目标模型的信息, 效果通常比黑盒攻击好。

2.2.2 基于目的

根据攻击者的目的, 又可将攻击方法分为目标攻击和非目标攻击。在目标攻击中, 生成的对抗样本 x' 被攻击者有意控制分类为 c' , 这一过程主要依赖于增加模型对错误类别 c' 类的输出概率来实现。而在非目标攻击中, 对手的目的仅仅是使模型出错, 模型给出的分类结果 c' 可以是正确类别 c 以外的任何类别, 因此非目标攻击主要通过降低正确类别 c 的输出概率来

实现。由于目标攻击对攻击目标的约束, 通常生成的对抗样本在迁移性上不如非目标攻击。

2.2.3 基于攻击粒度

在文本领域, 还可以根据扰动粒度的不同对攻击进行分类, 具体可以分为字符级攻击、单词级攻击和句子级攻击。字符级攻击对输入中某一单词的字符进行扰动, 常用的方法包括字符的添加、删除、替换、交换顺序等^[22]。单词级攻击对输入中的词进行扰动, 最主要的方法为词替换, 替换词的选择有很多, 包括基于词向量欧式距离^[23], 词向量余弦相似度^[24]等, 此外也可以添加或删除词, 攻击过程中最重要的是词义和语义相似性的衡量, 因为扰动原则上不能影响文本的语法性和通顺性。句子级攻击将整句原始输入视作扰动的对象, 狭义的句子级攻击会产生一个和原始输入语义相同但却使模型判断出错的句子, 常用的攻击方法包括添加不影响语义的无关句子^[6]、对原句进行语义不变的改述^[25]、对句子编码后重新解码等。更广义的句子级攻击还包括利用模型的潜在脆弱性使模型出错, 如 McCoy 等人^[26]在自然语言推理(Natural Language Inference, NLI)任务中发现, 在某些句法启发式模式下, 模型就会给出错误的推理结果。

2.3 文本对抗防御方法分类

防御的目的是保护基于 DNN 的系统免受对抗样本的攻击^[27], 由于攻击方法的多样性, 对抗防御远比攻击本身要难得多, 目前关于文本防御的工作远少于攻击工作。总体来讲, 现有的文本防御方法可以分为两大类, 启发式防御和可验证防御。

2.3.1 启发式防御

启发式防御方法是受攻击方法启发而来, 它针对攻击过程的某一环节进行破坏, 使得攻击者不能通过该攻击算法生成对抗样本, 因此对一些特定的攻击具有良好的防御性能。基于检测的防御、对抗训练防御和蒸馏防御等都属于启发式防御。其中基于检测的防御不更改训练好的模型, 而是在模型之外增加额外的防御措施, 通常会检测输入, 通过检测来发现恶意样本, 以及检测对模型的异常查询来过滤攻击者对系统的攻击。对抗训练防御方法是迄今为止性能最好的启发式防御算法, 该方法把生成的对抗样本也加入训练中去。在计算机视觉领域, 有很多工作对对抗训练的作用进行了研究^[28-30], 在文本领域, 对抗训练也被广泛使用。蒸馏防御^[31]是由 Hinton 等人^[32]提出的知识蒸馏方法演变而来, 该方法以知识蒸馏为基础, 通过控制蒸馏温度 T 使得模型更加平滑, 进而防御对抗样本。

2.3.2 可验证防御

可验证防御又称可证明防御, 该方法认为启发式防御只对特定攻击生成的对抗样本有效, 于是希望找到一种理论上可证明对所有攻击均鲁棒的防御方法。可验证防御能够计算出在某类攻击算法攻击下模型的鲁棒边界, 只要将鲁棒边界优化的足够好, 就能保证模型对所有微小扰动均鲁棒。关于可验证防御方法又可被分完全可验证方法和非完全可验证方法。在计算机视觉领域, 完全可验证方法如混合整数规划(Mixed Integer Programming, MIP)^[33-35]和可满足性模理论(Satisfiability Modulo Theories, SMT)^[36-37]等, 该类方法通常执行枚举搜索, 以计算精确的鲁棒边界。非完全方法如区间边界传播^[31, 76-77]和随机平滑^[32, 78-79]等, 该类方法仅计算鲁棒边界的上下界。在对抗攻击场景中, x_0 的邻域被描述为以 x_0 为中心的 L_p 球, x_0 附近使模型错误分类的样本到 x_0 的最小距离是改变模型预测的幅度最小攻击, 这也是 x_0 的最大可能鲁棒边界。然而不幸的是, 在具有线性激活函数 RELU^[44]的网络中, 寻找最小扰动的对抗样本已被证明是 NP 完全问题^[36, 45], 这使得完全可验证方法即便对于小型网络也需要非常昂贵的计算代价, 所以完全可验证方法仅具有理论意义而无法运用到现实中去, 通常退而求其次——计算最小扰动的下界^[46-48], 这类方法即非完全方法。区间边界传播和随机平滑是文本领域可验证防御的代表工作, 它们都属于非完全可验证方法。

2.4 扰动控制

在对抗攻击中, 一个基础的假设是样本加入的

扰动对人类是不可见的, 因此需要对扰动幅度进行控制, 使得加入扰动后生成的对抗样本不影响人类的判断。视觉领域中有很多关于扰动的度量方式, 最常用的如的 L_p 范数^[4]。对于一个向量 $\mathbf{z} = (z_1, z_2, \dots, z_n)$, 它的 L_p 范数定义如下式:

$$\|\mathbf{z}\|_p = \left(\sum_{i=1}^n |z_i|^p \right)^{\frac{1}{p}} \quad (1)$$

设原样本为 $\mathbf{x}$, 扰动后样本为 $\mathbf{x}'$, 则扰动的 L_p 范数如下式:

$$\|\Delta\mathbf{x}\|_p = \left(\sum_{i=1}^n |x_i - x'_i|^p \right)^{\frac{1}{p}} \quad (2)$$

其中 $\Delta\mathbf{x}$ 表示加入的扰动向量, x_i 和 x'_i 分别为 n 维样本向量 $\mathbf{x}$ 和扰动后向量 $\mathbf{x}'$ 的第 i 个分量。 p 可取不同值, 代表了不同的范数度量, 最长用的是 $p = 0, 1, 2, \infty$ 。在文本领域, L_2 范数使用的居多, 它可以用来度量原样本和对抗样本的欧式距离, 扰动的 L_2 范数越小说明原样本和对抗样本之间越接近, 通常也意味着相似性越高。不过图像的像素可以连续变化, 但文本是离散的, 对词向量做扰动得到的新向量不一定对应到词, 单是欧式距离并不能揭示文本的特殊性, 这也使得一些基于梯度的对抗攻击方式对很难直接应用于文本^[49-50], 另外, 文本扰动还有一个很强的约束是不能改变句子语义。因此我们需要一些更好的对文本相似性的度量。

余弦距离。词向量 Glove^[51]展示了词嵌入表示之间的一些几何关系, 相似的词之间往往会有相同的方向, 因此一个常用的单词相似性度量方法是计算词向量之间的余弦距离。设 $\mathbf{u} = (u_1, u_2, \dots, u_n)^T$ 和 $\mathbf{v} = (v_1, v_2, \dots, v_n)^T$ 表示两个词向量, 它们之间的余弦距离定义为如下公式:

$$D(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u}^T \cdot \mathbf{v}}{\|\mathbf{u}\| \cdot \|\mathbf{v}\|} \quad (3)$$

相比于欧式距离, 余弦距离更多关注的是词向量的方向。

句编码相似度。句编码旨在为句子学习一个压缩编码表示, 也就是将句子编码为一个向量, 这样句子相似度的衡量就可以在句子的向量表示之间进行。关于句编码的工作有很多, 代表性工作如 Subramanian 等人^[52]提出的多任务联合学习句编码和 Cer 等人^[53]提出的通用句编码(Universal Sentence Encoder, USE)。其中 USE 目前使用的最为广泛, 该方法利用 Transformer^[54]或深度平均网络^[55](DAN,

Deep Averaging Networks)的编码器部分, 将编码器输出作为词的嵌入表示和句的嵌入表示, 词的嵌入表示比较平凡, 而句的嵌入表示即由词嵌入表示求和后取加权平均得到。作者还提到在比较两个句子相似度时候不用原生的余弦距离, 而是转化为弧度之后计算, 如下公式:

$$\text{sim}(\mathbf{u}, \mathbf{v}) = \left(1 - \arccos \left(\frac{\mathbf{u}^T \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|} \right) / \pi \right) \quad (4)$$

Jaccard 相似度. 给定集合 A 和集合 B , 两个集合的 Jaccard 相似度定义为如下公式:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (5)$$

其中 $|A|$ 表示的是集合 A 中的元素个数, 显然 $0 \leq J(A, B) \leq 1$ 。Jaccard 相似度在文本中是一个常用的度量, 将两个文本看作两个集合, 它们之间的 Jaccard 相似度就是公共单词的个数除以两句总单词的个数, 越接近 1 则说明两个文本中的单词成分越相似。不过由于 Jaccard 相似度并没有考虑词的顺序信息和词的含义, 在语义衡量上效果并不是很好。

词移距离. 词移距离 (Word Mover's Distance, WMD) 是推土机距离 (Earth Mover's Distance, EMD) 的变种, EMD^[56] 常被用来衡量两个分布之间的距离, 是一个典型的线性规划问题。WMD^[57] 借鉴了 EMD 的思想, 将语义相似度的问题也转换成了一个线性规划问题。设有一个训练好的词向量矩阵 $W \in R^{n \times m}$, 词表中共 n 个词, $\mathbf{w}_i \in R^m$ 表示第 i 个词对应的 m 维词向量, 第 i 个词和第 j 个词的欧氏距离为 $d(i, j) = \|\mathbf{w}_i - \mathbf{w}_j\|_2$, 一条文本可以用稀疏向量 $s \in R^n$ 作为词袋表示, 如果在文本中词 i 出现了 c_i 次, 则 s 的第 i 个分量就是第 i 个词的词频:

$$s_i = \frac{c_i}{\sum_{j=1}^n c_j} \quad (6)$$

令 s 和 s' 分别表示待计算相似度的两条文本的词袋表示, 从概率分布的角度看, s 中每个单词都可以全部或部分地转移到 s' , 因此, 定义一个稀疏的转移矩阵 $T \in R^{n \times n}$, T_{ij} 表示有多少量从 s 中的词 i 转移到了 s' 中的词 j , $T_{ij} \geq 0$ 。进一步得到从 s 到 s' 转移的代价和如下:

$$\sum_{i,j} T_{ij} d(i, j) \quad (7)$$

最终将最小化转移代价的和转化为线性规划, 得到如下优化问题:

$$\begin{aligned} \min_{T \geq 0} & \sum_{i,j=1}^n T_{ij} d(i, j) \\ \text{s.t.} & \sum_{j=1}^n T_{ij} = s_i \quad \forall i \in 1, \dots, n \\ & \sum_{i=1}^n T_{ij} = s'_j \quad \forall j \in 1, \dots, n \end{aligned} \quad (8)$$

词移距离可以很好地衡量两个句子或文本之间的相似程度, 不过由于词袋模型并不考虑句法以及词之间的顺序信息, 所以在语义相似性衡量上也有一点不足。

编辑距离. 编辑距离也称 Levenshtein 距离^[58], 是一种量化从一个字符串到另一个字符串的最小变化的方法, 可推广到量化一个句子变换到另一个句子的最小代价。具体的, 设 $S_A = (w_A^1, w_A^2, \dots, w_A^{l_A})$, $S_B = (w_B^1, w_B^2, \dots, w_B^{l_B})$ 分别表示长度为 l_A 和 l_B 的两个句子, 其 w_A^i 表示句子 S_A 中第 i 个词, w_B^j 表示句子 S_B 中第 j 个词。在计算编辑距离时, 以 S_A 为基准句子, 计算 S_B 相对 S_A 的编辑距离, 即认为 S_B 是 S_A 经过删除、插入或替换后得到的新的句子。定义 $c_{i,j}$ 为 $(w_A^1, w_A^2, \dots, w_A^{l_A})$ 经过编辑转换为 $(w_B^1, w_B^2, \dots, w_B^{l_B})$ 所需的最小操作代价, 由于要考虑不同的编辑情况, 所以有如下公式:

$$c_{i,j} = \begin{cases} c_{i,j-1} + i(w_B^j) \\ c_{i-1,j} + d(w_A^i) \\ c_{i-1,j-1} + s(w_A^i, w_B^j) \end{cases} \quad (9)$$

其中 $i(w_B^j)$ 表示插入 w_B^j 的代价, $d(w_A^i)$ 表示删除 w_A^i 的代价, $s(w_A^i, w_B^j)$ 则表示将 w_A^i 替换为 w_B^j 的代价。该问题同样可以用动态规划算法解决。结合词向量可以给出更好的用于文本处理的编辑距离^[59], 所谓的结合词向量, 就是对这三个代价函数做了修改, 使之利用词向量所含的语义信息。对插入操作代价定义为如下公式:

$$\begin{aligned} i(w_B^j) &= 1 - \lambda \cdot \max_{w_A^k} (\text{sim}(w_A^k, w_B^j) + \mu) \\ &w_A^k \in S_A, w_A^k \neq w_A^i \end{aligned} \quad (10)$$

其中 sim 用来衡量两个单词的相似度, 对删除操作代价定义为如下公式:

$$\begin{aligned} d(w_A^i) &= 1 - \lambda \cdot \max_{w_B^k} (\text{sim}(w_A^i, w_B^k) + \mu) \\ &w_B^k \in S_B, w_B^k \neq w_B^j \end{aligned} \quad (11)$$

对替换操作代价定义为如下公式:

$$s(w_A^i, w_B^j) = 2 - 2\text{sim}(w_A^i, w_B^j) \quad (12)$$

即当插入新的词时, 如果该词和句子 A 中的词有相

似的, 那么对语义的影响就不会很大。当删除句子 A 中的词时, 如果句子 B 中还有词和这个删除的词相似, 那么对语义的影响也不会很大。

3 文本中的攻击方法

这一部分将按攻击粒度来介绍文本领域的对抗攻击相关工作, 根据攻击粒度可将攻击分为字符级攻击、单词级攻击和句子级攻击。

3.1 字符级攻击

字符级攻击通过修改单词中的字符来生成对抗样本。常用的方法包括字符的添加、删除、替换、交换顺序等, 具体到字符替换方法, 有字符随机替换^[60]、基于 One-Hot 编码的字符替换^[22]以及基于字形相似的替换^[61]等。

Ebrahimi 等人^[22]提出了一种基于单词中字符的原子翻转操作(替换、插入、删除等)的白盒攻击方法 HotFlip。具体的, 对文本而言, 其字符序列表示如下公式所示:

$$\mathbf{x} = [(x_{11}, \dots, x_{1n}); \dots (x_{m1}, \dots, x_{mn})] \quad (13)$$

其中 n 是一个单词允许的最多字符数量, m 是一个样本中的单词个数, V 是字符表, 也就是语言下所有字符, 每个 $x_{ij} \in \{0, 1\}^{|V|}$ 都是一个 $|V|$ 维 one-hot 向量, 表示第 i 个词的第 j 个字符。则第 i 个单词的第 j 个字符从 a 变为 b 的翻转操作可以定义为如下公式:

$$\vec{v}_{ijb} = (\vec{0}, \dots; (\vec{0}, \dots (0, \dots -1, 0, \dots, 1, 0)_j, \dots \vec{0}_i; \vec{0}, \dots) \quad (14)$$

其中 a 和 b 是字符表的第 a 和第 b 个字符, -1 和 1 分别是 a 和 b 对应的位置, 论文将对于样本的替换操作视为一个个向量, 利用损失函数对于向量的导数, 就能知道哪种替换操作能够让损失最大化, 如下公式:

$$\begin{aligned} & \max \nabla_x J(\mathbf{x}, \mathbf{y})^T \cdot \vec{v}_{ijb} \\ &= \max_{ijb} \frac{\partial J^{(b)}}{\partial x_{ij}} - \frac{\partial J^{(a)}}{\partial x_{ij}} \end{aligned} \quad (15)$$

插入操作可以视为替换操作的一种特殊情形, 只要将插入位置的后一位字母替换为待插入字母, 再将插入位置后字母依次后移即可, 相应导数计算如下公式:

$$\begin{aligned} & \max \nabla_x J(\mathbf{x}, \mathbf{y})^T \cdot \vec{v}_{ijb} = \\ & \max_{ijb} \frac{\partial J^{(b)}}{\partial x_{ij}} - \frac{\partial J^{(a)}}{\partial x_{ij}} \\ & + \sum_{j'=j+1}^n \left(\frac{\partial J^{(b)}}{\partial x_{ij'}} - \frac{\partial J^{(a')}}{\partial x_{ij'}} \right) \end{aligned} \quad (16)$$

删除操作则是将删除位置后的字符依次向前移动, 同样可以写成类似的形式。得到哪种操作能够让损失最大化后就可以执行相应的操作进行攻击。

HotFlip 方法为后来的很多字符级攻击方法提供了基础。Gil 等人^[62]提出并不需要每次都执行具体的攻击算法, 可以训练替代模型模仿白盒攻击的行为, 为此作者利用 HotFlip 攻击算法生成的对抗样本为训练集来挖掘对抗样本的生成模式, 并将练出的神经网络用于黑盒攻击, 成功模拟了 HotFlip 的行为, 由于此方法生成对抗样本的耗时与 HotFlip 方法的执行无关, 这大幅度降低了生成对抗样本所需要的时间, 而且能达到几乎和 HotFlip 同样的效果。Ebrahimi 等人^[63]在机器翻译中发现, 使用 HotFlip 等针对性的攻击能帮助发现模型的潜在漏洞, 再借助对抗训练可有助于提升字符级机器翻译模型的鲁棒性。

Gao 等人^[64]提出了一种黑盒场景生成对抗样本的方法 DeepWordBug。该方法分为两个阶段, 第一个阶段计算单词的重要性, 这个过程会选择对分类贡献最重要的词。第一阶段对单词重要性的计算如下公式:

$$CS(x_i) = THS(x_i) + \lambda TTS(x_i) \quad (17)$$

它是由两个重要性打分函数的结果综合而来, 第一个部分 THS(x_i) 的计算如下公式:

$$THS(x_i) = F(x_1, \dots, x_i) - F(x_1, \dots, x_{i-1}) \quad (18)$$

其中 F 是神经网络分类器, THS 衡量了单词 x_i 对其之前文本的重要性。第二个部分 TTS(x_i) 计算如下公式:

$$TTS(x_i) = F(x_i, \dots, x_n) - F(x_{i+1}, \dots, x_n) \quad (19)$$

TTS 衡量了单词 x_i 对其之后文本的重要性。最终单词的重要性 $CS(x_i)$ 是考虑语境的综合结果, λ 是一个平衡前后语境的超参数。第二个阶段则是攻击的具体实现, 该过程将对候选词的字符进行扰动, 具体扰动操作同 HotFlip^[22], 并使用编辑距离来进行扰动幅度控制。

由于字符级操作大概率会生成未登录词, 破坏句子的语法性并造成对抗样本不可读, 所以字符级攻击生成的对抗样本质量通常并不是很高, 一些基于检测和纠错的方法能轻松过滤或纠正这类对抗样本。

3.2 单词级攻击

单词级攻击是文本对抗攻击中最重要也是最常用的攻击方法, 在文本分类^[5]、机器翻译^[65]等任务中都可以见到单词级攻击的身影。单词级攻击最主要实现方式为词替换, 替换词的选择包括基于词向量相似度^[23,66]、基于同义词^[67-68]和基于义原^[69]等。

Papernot 等人^[2]最先在循环神经网络上研究了词级对抗样本的问题。具体的, 作者通过计算梯度得到每一个时间步的输入对分类输出的贡献值, 贡献值越大说明对分类结果越重要, 对这些词进行扰动就能大概率使模型出错。在扰动实现上, 作者将视觉领域的 JSMA 算法^[70]扩展到了文本领域, 并利用 FGSM 计算扰动, 由于词向量不能取任意实数值, 作者将扰动后得到的嵌入向量投影到词向量空间中最近的向量上, 这样就能得到有效的词嵌入向量, 从而对应到真实的单词。不过尽管该攻击方法使模型出错了, 但句子中被替换的单词有可能和原单词是无关的, 因此出现病句的可能性非常高, 不够符合对抗样本对人类而言不可察觉这一要求。

为了保证生成对抗样本的质量, 基于同义词替换的方法应运而生。Kuleshov 等人^[71]提出了一种贪婪搜索算法(Greedy Search Algorithm, GSA), 旨在用同义词替换来保持文本语义和语法的正确性。GSA 首先为输入文本 $x = \{w_1, w_2, \dots, w_i, \dots, w_n\}$ 构造一个同义词集合 W_s , 其中:

$$W_s = \{\text{Syn}(w_i, \sigma, k) \mid 1 \leq i \leq n\} \quad (20)$$

$\text{Syn}(w_i, \sigma, k)$ 表示单词 w_i 的前 k 个在距离 σ 之内的近义词, 初始时令对抗样本 $x_{adv} = x$, 之后在迭代过程中, 对于 $x_{adv} = \{w'_1, w'_2, \dots, w'_n\}$, GSA 找到一个单词 w'_i 并将它替换为可使模型输出置信度最小的同义词 $\hat{w}'_i$, 更新 $x_{adv} = \{w'_1, w'_2, \dots, \hat{w}'_i, \dots, w'_n\}$, 该过程迭代进行直至 x_{adv} 成为对抗样本。

Alzantot 等人^[23]提出了一种基于遗传算法(Genetic Algorithm, GA)生成对抗样本的方法, 该算法包含三个基本操作, $\text{Mutate}(x)$, $\text{Sample}(P)$ 和 $\text{Crossover}(a, b)$ 。其中 $\text{Mutate}(x)$ 操作是在文本 x 中选择一个尚未更新的单词 w_i , 并使用同义词 $\text{Syn}(w_i, \sigma, k)$ 之一的 $\hat{w}_i$ 进行替换, 要求替换后的文本能使模型输出的置信度降低; $\text{Sample}(P)$ 操作是从 x 的变异亲代 $P = \{x_0, x_1, \dots, x_i, \dots, x_m\}$ 中以一定概率随机抽取 x_i ; $\text{Crossover}(a, b)$ 操作是构建一个新的文本 $c = \{w_1^c, \dots, w_i^c, \dots, w_n^c\}$, 其中 w_i^c 从 $\{w_i^a, w_i^b\}$ 中随机选择得到。对于文本 x , GA 首先生成一个初始大小为 m 的亲代 P^0 :

$$P^0 = \{\text{Mutate}(x), \dots, \text{Mutate}(x)\} \quad (21)$$

接着在每轮迭代中, GA 通过 Mutate 和 Crossover 操作生成更好的下一代:

$$\begin{aligned} x_{adv}^{i+1} &= \arg \min_{\tilde{x} \in \mathcal{P}^i} S_m(\tilde{x}, y_{\text{true}}) \\ c_k^{i+1} &= \text{Crossover}(\text{Sample}(P^i), \\ &\quad \text{Sample}(P^i)) \\ P^{i+1} &= \{x_{adv}^{i+1}, \text{Mutate}(c_1^{i+1}), \\ &\quad \dots, \text{Mutate}(c_{m-1}^{i+1})\} \end{aligned} \quad (22)$$

其中 $S_m(x, y)$ 表示类别 y 的置信度, 当 GA 找到对抗样本或达到最大迭代次数时则终止迭代。Wang 等人^[72]认为 GA 在迭代时每次只替换同一位置单词一次, 这很可能会让算法陷入局部极小值, 于是在 GA 的基础上提出了改进遗传算法(Improved Genetic Algorithm, IGA), 解决了原始 GA 可能陷入局部极小值的问题。

Goodfellow 等人^[2]提出的方法可以基于梯度在特征空间做扰动生成对抗样本, 但这种方式生却不能直接应用于文本, 因为文本是离散的, 对词向量空间的扰动生成的新向量并不一定对应到真实的单词。Gong 等人^[49]通过在词向量空间中搜索对抗样本来解决这个问题, 虽然扰动后的词向量并不一定对应到真实单词, 但是可以通过最近邻居搜索到的词向量对应的真实单词来代替, 为了保证生成对抗样本的质量, 作者采用 WMD 来衡量对抗样本和原样本的相似度。Sato 等人^[73]提出了 iAdv-Text 的攻击方法, 通过对扰动的方向加入约束, 使得对词向量加入扰动后, 能够得到另一个单词的词向量, 从而能还原为真实单词。该方法的核心过程可以看作一个优化问题, 如下公式:

$$\begin{aligned} \mathcal{J}_{\text{iAdvT}}(D, W) = \\ \frac{1}{|D|} \sum_{(\tilde{X}, \tilde{Y}) \in D} L(\tilde{X} + r(\alpha_{\text{iAdvT}}), \tilde{Y}, W) \end{aligned} \quad (23)$$

其中 D 是训练集, $\tilde{X}$ 和 $\tilde{Y}$ 分别为输入和输入对应的标签, L 是负对数似然损失, W 是网络参数, $r(\alpha_{\text{iAdvT}})$ 是在输入 $\tilde{X}$ 上产生的扰动, α_{iAdvT} 是使损失最大的扰动向量方向, 定义如下:

$$\alpha_{\text{iAdvT}} = \arg \max_{\alpha, \|\alpha\| \leq \delta} \left\{ L\left(\tilde{w} + \sum_{k=1}^{|V|} a_k d_k, \tilde{Y}, W\right) \right\} \quad (24)$$

其中 d_k 是词向量空间中词 w 到词汇表中的第 k 个单词的方向向量, 并且 $\|d_k\|_2 = 1$, a_k 是 $|V|$ 维向量 α 的第 k 个分量, 表示词 w 到词汇表中的第 k 个单词的方向权重, 于是 $\sum_{k=1}^{|V|} a_k d_k$ 就是在输入词向量 $\tilde{w}$ 上产生的约束方向的扰动, 因为公式(24)的计算需要枚举搜索整个词向量空间, 会消耗大量时间, 论文直接使用

梯度方向来近似:

$$\alpha_{i\text{AdvT}} = \frac{\epsilon g}{\|g\|_2} \quad (25)$$

$$g = \nabla_{\alpha} \ell \left(\hat{w} + \sum_{k=1}^{|V|} a_k d_k, \hat{y}, W \right)$$

同时作者还利用余弦相似度来挑选最优的扰动词。

Li 等人^[74]提出的 Textbugger 方法在黑盒和白盒情况下均适用, 在白盒设定下, 该方法使用雅克比矩阵 J 来衡量文本中单词的重要性, 公式如下:

$$C_{x_i} = J_{F(i,y)} = \frac{\partial F_y(x)}{\partial x_i} \quad (26)$$

其中 $F_y(\cdot)$ 表示输入句 x 属于类别 y 的置信度, C_{x_i} 表示文本 x 中第 i 个单词 x_i 的重要性分数。作者根据 x_i 的重要性来决定待攻击词的选择。在待攻击词选择之后, Textbugger 有 5 种攻击词生成方法:

- 1) 插入一个空格到单词中;
- 2) 删除除第一个和最后一个字符外的任意字符;
- 3) 在单词中随机交换两个相邻的字母; 但不改变第一个或最后一个字母;
- 4) 用视觉上相似的字符或键盘上相邻的字符代替原字符;
- 5) 在词向量空间中, 用它的最近邻词来替换。

在黑盒场景下, 作者将文本分割为序列, 然后查询目标模型, 从原始文档中选择出对预测结果贡献最大的句子, 接着在该句子中通过删除其中的词来计算类别置信分数得到最重要的词, 公式如下:

$$C_{x_i} = F_y(x_1, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_n) - F_y(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) \quad (27)$$

接着执行和白盒场景相同的攻击词选择方法来生成对抗样本。这类基于单词重要性的攻击思路几乎成为一种范式, 很多攻击方法都遵循了这个思路^[66-68,74-76]。

Ren 等人^[67]提出了一种新的基于词重要性的同义词替换攻击(Probability Weighted Word Saliency, PWWS), 该攻击方法综合考虑了词的重要性和分类置信度。具体的, PWWS 首先计算了文本 x 中每个单词的重要性 $S(x, w_i)$:

$$S(x, w_i) = S_m(x, y_{\text{true}}) - S_m(\bar{x}_i, y_{\text{true}}) \quad (28)$$

其中 $\bar{x}_i = \{w_1, \dots, w_{i-1}, \text{unk}, w_{i+1}, \dots, w_n\}$, unk 表示从文本中移除了该单词。接着 PWWS 计算用同义词替换 w_i 所引起的最大可能分类置信度变化:

$$\Delta S_{m_i}^*(x) = \max_{\hat{w}_i \in \text{Syn}(w_i, \sigma, k)} [S_m(x, y_{\text{true}}) - S_m(\hat{x}_i, y_{\text{true}})] \quad (29)$$

其中 $\hat{x}_i = \{w_1, \dots, w_{i-1}, \hat{w}_i, w_{i+1}, \dots, w_n\}$ 。之后 PWWS 按 $\phi(S(x, w_i))_i \Delta S_{m_i}^*(x)$ 的值降序检查单词, 其中 $\phi(z)_i$ 是 Softmax 作用后的第 i 个分量, 将 w_i 替换为它的最优近义词 w_i^* :

$$w_i^* = \arg \max_{\hat{w}_i \in \text{Syn}(w_i, \sigma, k)} [S_m(x, y_{\text{true}}) - S_m(\hat{x}_i, y_{\text{true}})] \quad (30)$$

当找到对抗样本 x_{adv} 或已经替换完文本 x 中所有单词时, PWWS 终止。

Jia 等人^[66]提出 TextFooler 算法试图在前人基础上生成质量更高的对抗样本, 与 Samanta, Li 等人^[50,54]的方法类似, 该方法也是基于单词重要性来进行攻击, TextFooler 成功攻击了基于 Bert^[77]的预训练语言模型。

随着词攻击方法的演变, 研究者们逐渐意识到虽然现有攻击方法能使得模型性能大幅度下降, 但生成的对抗样本在流畅性和自然性上仍旧有所欠缺。Garg 等人^[78]认为 TextFooler 仅仅通过词向量的余弦距离来衡量单词的相似性, 这样替换词后很容易生成一些人类读起来不自然的句子, 为此作者提出了基于预训练语言模型的攻击方法(Bert-based Adversarial Examples, BAE), 该方法利用 Bert 的遮蔽语言模型特性, 利用语境生成对抗样本。具体的, 作者首先通过分类器的输出来找到对分类输出贡献最大的词, 这个过程和 TextFooler 等基于单词重要性的攻击方法相同, 找到这些词后对它们进行排序作为待攻击词。在攻击的实现上有替换和插入两种, 替换和插入的词是将待攻击词遮蔽后, 语言模型根据语境预测出来的词, 选取预测词中的前 k 个为候选, 为了保证句子的语义和通顺性, 作者还使用 USE 来衡量替换和插入词后的语义相似度来对这 k 个候选进行过滤, 然后在过滤后的词中进行搜索, 如果使得模型出错则找到对抗样本。

Li 等人^[76]同样基于 Bert 提出了一种高效的对抗样本生成方法 BERT-Attack, 该方法首先在给定的输入序列中找到易受攻击的词, 这个过程和 BAE 方法等类似, 单词 w_i 的重要性计算如下公式:

$$I_{w_i} = o_y(S) - o_y(S_{\setminus w_i}) \quad (31)$$

其中 $S_{\setminus w_i} = [w_0, w_1, \dots, [\text{MASK}], \dots]$, 表示将单词 w_i 遮蔽后模型的输出 $O_y(S)$ 会改变多少, 接着可以根据重要性对单词排序并选出一部分作为待攻击词。然后以语义保持的方式应用 Bert 生成待攻击词的替代候选。利用 BERT 的能力, 在考虑语境的情况下生成对抗样本。

单词级攻击通常能生成质量非常好的对抗样本,

因为对一句话中某些词进行同义替换或删除某些词并不会改变这句话的含义, 所以单词级攻击是使用最广泛的一种攻击。

3.3 句子级攻击

句子级攻击以整个句子为关注对象, 目标是生成一个不改变原句语义的句子, 并使模型预测出错。常用的句子级攻击方法包括添加与输出无关的句子^[6]、句子改述^[25,79]、编码后重新解码^[80-81]、启发式攻击^[26]等。

Jia 和 Wang 等人^[6,82-83]首先在问答系统中研究了文本领域的对抗样本, 只需在文章段落中添加句子或对问题进行更改就能使模型回答出错。在实现上, 他们提取了问题中的名词和形容词, 并用反义词进行替代, 同时命名实体和数字都用词向量空间中最接近的词代替, 修改后的问句还被转换为陈述句加入到原段落的末尾。这些简单的操作可大幅降低问答系统的效果。

Iyyer 等人^[79]提出了一种通过句法控制对原句进行重述的编码-解码模型架构(Syntactically Controlled Paraphrase Networks, SCPN), 该方法首先通过反向翻译获得重述句子对 (s_1, s_2) , 接着对句子进行句法分析得到对应的句法成分对 (p_1, p_2) , 然后利用标记信息训练一个编码-解码模型, 其中模型的输入是 s_1 和 p_2 , 生成 s_2 , 为了避免模型学习偏差, 作者同时训练了 (s_2, s_1) , 即输入是 s_2 和 p_1 , 生成 s_1 。该方法生成的对抗样本会表现出更少的词汇改变, 而更倾向于纯粹的句法修改。Gan 等人^[81]设计了一个能根据原句子和句子改述建议词来生成改述句的网络, 该网络同样是一个编码-解码模型架构, 因为对原句子的改述是一个生成过程, 生成的新句子可能用到原句子中的单词, 所以作者利用 See 等人^[84]在机器翻译任务中提出的复制机制扩展了模型的解码器。

Ribeiro 等人^[25]提出了一种根据语义等价来生成对抗样本的方法语义等价对抗(Semantically Equivalent Adversarials, SEAs), 具体的, 考虑一个黑盒模型 f , 它对输入 x 做出预测 $f(x)$, 通过生成 x 的转述 x' , 并观察新的 x' 从 f 中得到预测, 如果 x' 能改变原始预测, 则得到对抗样本:

$$\begin{aligned} \text{SEA}(x, x') &= 1 \\ \text{if SemEq}(x, x') \wedge f(x) &\neq f(x') \end{aligned} \quad (32)$$

其中 $\text{SemEq}(x, x')$ 是语义等价指示函数, 如果 x 和 x' 等价则返回 1 否则返回 0, $\text{SEA}(x, x')$ 表示语义等价对抗样本 x' 是否获得, 作者将得到的这些对抗样本

概括为语义等价的对抗规则, 这些简单的替换规则可以在许多文本上诱导模型出错。

Minervin 等人^[85]在自然语言推理中研究了对抗样本, 他们将对对抗样本的生成视为一个组合优化问题, 并根据指定的一阶逻辑约束来生成对抗样本。具体的, 设 **body** 和 **head** 分别为推理前提和推理结论, $X_1, X_2, \dots, X_n$ 表示一阶逻辑规则中的变量, $S = \{X_1 \rightarrow s_1, X_2 \rightarrow s_2, \dots, X_n \rightarrow s_n\}$ 表示候选替换集合, 其中 s_i 是句子, $p(S; \text{body})$ 和 $p(S; \text{head})$ 表示当逻辑变量 X_i 被替换为句子 s_i 时, 相应一阶逻辑规则成立的概率。论文通过优化非一致性损失 J_I 来生成对抗样本, 如公式:

$$\begin{aligned} \max_S J_I(S) &= [p(S; \text{body}) - p(S; \text{head})]_+ \\ \text{s.t. } \log p_L(S) &\leq \tau \end{aligned} \quad (33)$$

其中 $[x]_+ = \max(0, x)$, $p_L(S)$ 表示句子 S 的复杂度, 为了防止生成的对抗样本没有意义, 通过语言模型^[86]将其复杂度限制在 τ 幅度下。该方法生成的对抗样本也被作者用来评估自然语言推理模型的鲁棒性。

McCoy 等人^[26]认为 DNN 这类统计模型的效果都是源自启发式, 即 DNN 学习到的仅是训练集中的统计规律, 并把规律应用到与训练数据同分布的测试数据。因此 DNN 模型只是服从于训练数据所表现出来的统计特性, 而不是捕捉更根本的特征。如果数据集集中这样的启发式模式经常对应正确的输出, 那么损失函数就不能激励模型学习真正重要的知识, 即便模型产生了正确的输出也是因为启发式导致的, 并不是模型真的理解了句子。作者提到了三种启发式: 词汇重叠启发式、子序列启发式和成分启发式, 三者是由低到高的层次关系。围绕这几种启发式作者重新设计了 HANS 数据集, 结果所有 DNN 模型在 HANS 数据集上都出错了, 很多当下流行的模型在 HANS 上的表现大幅降低, 甚至是 0 的准确率。这说明 DNN 学习到的只是浅层特征, 并没有捕捉到根本特征。

Moosavi 等人^[64]在视觉领域发现了可用于全局的攻击, 该方法能生成对任何图像实现攻击的通用扰动, 使得加入该扰动的样本都分类错误。与之类似, Wallace 和 Behjati 等人^[88-89]在文本领域提出了一种序列级别的通用扰动, 只需在文本中添加与输入无关的单词序列, 即可触发模型生成特定的预测, 具体做法是优化下式:

$$\underset{x_{adv}}{\operatorname{argmin}} \mathbb{E}_{x \sim \mathcal{T}} [\mathcal{L}(\tilde{y}, f(x_{adv}; x))] \quad (34)$$

其中 $\mathcal{T}$ 是数据输入序列分布, $\mathcal{L}$ 是任务的损失函数, $\hat{y}$ 是目标类别, $\mathbf{x}$ 是初始单词序列, $\mathbf{x}_{adv}$ 是待搜索的攻击触发序列。将 $\mathbf{x}_{adv}$ 加入句子开头或结尾就能触发模型生成特定预测。Song 和 Zhao 等人^[80,90]认为 Wallace 和 Behjati 等人的方法生成的攻击序列通常是无意义的单词组合, 为了让生成的触发序列更加自然, 作者使用了一个预先训练好的对抗正则化自编码器 (Adversarially Regularized AutoEncoders, ARAE)^[91], ARAE 由编码器结构^[92]和生成对抗网络^[93]组成, 其输入是标准高斯噪声 $\mathbf{t}$, 由生成器映射为中间向量 $\mathbf{z}$ 后再由解码器解码生成一个单词序列 $\mathbf{x}_{adv}$, 整个过程如下公式:

$$\begin{aligned} \mathbf{z} &= \text{GENERATOR}(\mathbf{t}), \\ \mathbf{x}_{adv} &= \text{DECODER}(\mathbf{z}), \\ \mathbf{x}' &= [\mathbf{x}; \mathbf{x}_{adv}] \end{aligned} \quad (35)$$

因为句子级攻击从句子语义角度生成对抗样本, 在扰动控制上更加困难, 因此句子级攻击的难度比字符级和单词级要大的多, 目前尚未有较好的体系化攻击策略, 通常来说句子级攻击最能检验 DNN 模型的鲁棒性。

3.4 文本对抗攻击方法总结

对比以上三种不同粒度的攻击, 字符级攻击所产生的对抗样本质量往往一般, 通常情况下会生成未登录词并破坏句子可读性, 目前有一些基于检测和纠错的防御方法可以很好地应对字符级的攻击。相比字符级攻击, 单词级攻击在生成对抗样本质量

上往往表现的更好, 一般对一句话中某些词进行同义替换或添加并不会改变句子的语义, 再利用语言模型进行控制, 所生成的对抗样本的通顺性、语法性也往往能够得到保证, 因此攻击成功率也较高。在攻击的实现上, 现有攻击方法通常可分为两个阶段, 第一个阶段评估单词对分类结果的贡献度并进行排序, 第二个阶段针对分类贡献大的这些词进行攻击。Zang 等人^[69]进一步指出现有字符级和单词级攻击本质上是一个组合优化问题, 因为对于词级文本对抗攻击, 在词表有限的情况下, 词级扰动也是有限的, 再加上语义的限制, 词级扰动的选择实际也非常有限。而攻击的目标则是找出一个能够成功攻击模型的对抗样本, 即优化问题的最优解。

句级的攻击往往使得对抗样本和原始输入之间有较大差别, 通常来说很难控制对抗样本句的质量, 不过句子级攻击是从语义角度出发的, 对句子级攻击的防御最能体现模型的鲁棒性, 因此句子级攻击的研究也非常重要。表 1 给出了不同攻击粒度的工作汇总。表 2 给出了一些典型攻击方法的实例。

4 文本对抗攻击的防御

这一部分将介绍针对文本对抗攻击的防御方法。防御的目的是使得模型免受恶意对抗样本的攻击, 由于生成对抗样本的多样性, 对攻击的防御比攻击本身要难得多, 也因此目前关于防御的工作很有限。

表 1 不同攻击粒度的方法对比

Table 1 Work comparison of different attack granularity

攻击方法	粒度	黑盒/白盒	目标/非目标	扰动控制	基于梯度
Eger 等人 ^[61]	字符级	黑盒	非目标	余弦相似度	否
HotFlip ^[22]	字符级	白盒	目标攻击	余弦相似度	是
Gil 等人 ^[62]	字符级	黑盒	非目标攻击	-	否
DeepWordBug ^[64]	字符级	黑盒	非目标攻击	编辑距离	否
Ebrahimi 等人 ^[63]	字符级	黑盒和白盒	非目标攻击	-	是
Belinkov 等人 ^[60]	字符级	黑盒	非目标攻击	-	否
Papernot 等人 ^[70]	单词级	白盒	非目标攻击	-	是
Samanta 等人 ^[68]	单词级	白盒	非目标攻击	-	是
GSA ^[71]	单词级	黑盒	非目标攻击	欧氏距离	否
iAdv-Text ^[73]	单词级	白盒	非目标攻击	余弦相似度	是
PWWS ^[67]	单词级	白盒	非目标攻击	欧式距离	是
GA ^[23]	单词级	黑盒	目标攻击	欧氏距离	否
Cheng 等人 ^[97]	单词级	白盒	非目标攻击	余弦相似度	是
Yang 等人 ^[75]	单词级	黑盒	非目标攻击	-	否
Liang 等人 ^[50]	单词级	黑盒和白盒	目标攻击	-	是
Cheng 等人 ^[65]	单词级	白盒	目标攻击	-	是

续表					
攻击方法	粒度	黑盒/白盒	目标/非目标	扰动控制	基于梯度
Song 等人 ^[90]	单词级	白盒	非目标攻击	-	是
TextFooler ^[66]	单词级	白盒	目标攻击	综合	是
Gong 等人 ^[49]	单词级	白盒	目标攻击	词移距离	是
Zhang 等人 ^[24]	单词级	黑盒和白盒	目标攻击	余弦相似度	是
IGA ^[72]	单词级	黑盒	非目标攻击	p-范数	否
Zang 等人 ^[69]	单词级	黑盒	非目标攻击	-	否
BAE ^[78]	单词级	黑盒	非目标	句编码相似度	否
BERT-ATTACK ^[76]	单词级	黑盒	非目标	句编码相似度	否
TextBugger ^[74]	混合	黑盒和白盒	非目标攻击	综合	否
Jia 等人 ^[6]	句子级	黑盒	非目标攻击	-	否
Wang 等人 ^[82]	句子级	黑盒	非目标攻击	-	否
SCPN ^[79]	句子级	黑盒	非目标攻击	-	否
Gan 等人 ^[81]	句子级	黑盒	非目标攻击	-	否
Zhao 等人 ^[80]	句子级	白盒	非目标攻击	-	是
Wallace 等人 ^[88]	句子级	黑盒	目标攻击	-	否
Minervini 等人 ^[85]	句子级	白盒	非目标攻击	-	是
SEAs ^[25]	句子级	黑盒	非目标攻击	-	否
Mudrakarta 等人 ^[83]	句子级	白盒	非目标攻击	-	是
McCoy 等人 ^[26]	句子级	黑盒	非目标攻击	-	否

(注: “-” 表示使用了其它的扰动控制方法。)

表 2 一些攻击方法的实例

Table 2 Instances of some attack methods

攻击方法	主要思想	扰动粒度	对抗样本例子
DeepWordBug ^[64]	基于分数的贪心字符扰动	字符级	原句: it's the cinematic equivalent of a good page-turner, and even if it's nonsense , its claws dig surprisingly deep . (65.44% positive) 扰动句: it's the cinematic equivalent of a good page-turner, and even if it's nonsenes , its claws dig surprisingly depe . (78.32% negative)
HotFlip ^[22]	梯度优化的字符扰动	字符级	原句: despite some gulps the film is a fuzzy huggy. (99.72% positive) 扰动句: despte some gulps the film is a fuzzx huggy. 99.00% negative
PWWS ^[67]	基于分数的贪心词替换	单词级	原句: it's one of the saddest films i have ever seen that still manages to be uplifting but not overly sentimental. (100% negative) 扰动句: It ' s one of the saddest films i have ever seen that still manages to be uplifting but non overly sentimental. (54.13% positive)
TextBugger ^[74]	基于分数的贪心词、字符替换	字符级、单词级	原句: here's a british flick gleefully unconcerned with plausibility, yet just as determined to entertain you. (71.27% negative) 扰动句: here's a british flcik gleefully unconcerned with plausibility, yet just as determined to entertain you. (53.00% positive)
GA ^[23]	基于遗传算法的词替换	单词级	原句: it is ridiculous, of course, but it is also refreshing, disarming, and just outright enjoyable despite its ridiculousness. (77.74% positive) 扰动句: it is ridiculous, of course, but it is also refreshing , disarming, and just outright enjoyable notwithstanding its ridiculousness. (53.24% negative)
TextFooler ^[66]	基于分数的贪心词替换	单词级	原句: The characters, cast in impossibly contrived situations , are totally estranged from reality. (72.47% negative) 扰动句: The characters, cast in impossibly engineered circumstances , are fully estranged from reality. (84.25% positive)
SEAs ^[25]	基于规则的句子重述	句子级	原句: this is a movie I really don't like, it's terrible. (100% negative) 扰动句: that is a movie I really don't like, it is terrible. (57.84% positive)
SCPN ^[79]	基于句法的句子重述	句子级	原句: there is no pleasure in watch this movie . (90% negative) 扰动句: in watch this movie, there is no pleasure . (57.94% positive)

4.1 基于检测的防御

对抗样本是有着特殊目的一些数据, 因此可以考虑检测出对抗样本。Metzen 等人^[94]提出可以用神经网络探测器来识别对抗样本, 即把对抗样本的识别直接转化为端到端方式的二分类问题, 进而过滤对抗样本。Zhou 等人^[95]提出一个架构来学习和区分对抗扰动, 该方法不需要改动原始网络, 而是在原始网络之外引入扰动鉴别器和词嵌入估计器, 扰动鉴别器将计算并衡量文本中的单词序列受到扰动的可能性, 并提供一组可能加入的扰动, 对于每个可能的扰动, 词嵌入估计器会根据上下文学习恢复原始单词的嵌入向量。Pruthi 等人^[80]在下游分类器前使用了一个单词拼写识别模型来纠正拼写错误, 进而防御拼写错误类对抗样本。尽管检测和纠错的确能防御一些对抗样本, 但正常样本也有被鉴别为对抗样本的可能性, 基于检测的防御会使模型在正常样本上的准确性降低, 因此基于检测的防御效果并不是很好。

4.2 基于对抗训练的防御

对抗训练方法被广泛地运用在图像领域防御对抗样本的攻击, 该方法在特征空间做扰动生成对抗样本并将其加入训练, 其本质是一种数据增强。对抗训练可以被理论化为 Min-Max 优化, 公式如下:

$$\min_{\theta} \rho(\theta) \quad (36)$$

$$\rho(\theta) = \mathbb{E}_{(x,y) \sim \mathcal{D}} \left[\max_{\delta \in \mathcal{S}} L(\theta, x + \delta, y) \right]$$

Max 过程是可以看作生成对抗样本的过程, 这些对抗样本可以使模型准确率降低, 而 Min 过程就是将对抗样本加入训练集的训练过程。但是在文本领域, 对特征空间做扰动生成的新向量并不一定对应到真实单词, 因此文本领域的对抗训练比图像领域棘手得多, 必须首先生成高质量的对抗样本, 这依赖于优秀的攻击算法。Ebrahimi, Alzantot, Ren 等^[22-23,67]都提到对抗训练方法可以提升 NLP 模型的分类鲁棒性。Cheng 等人^[97]则发现对抗训练可以提升机器翻译模型的鲁棒性。

值得一提的是, 虽然在词向量空间做扰动并不一定对应到真实的单词, 但也能直接用到对抗训练过程。Miyato 等人^[98]直接对词向量空间做扰动并进行对抗训练, 在文本任务上却取得了更好的结果, Zhu 等人^[99]也使用词向量空间扰动的对抗训练方法提升了 Bert 模型在众多任务上的表现。这种对抗训练更类似于一种正则化的手段, 含义不同而语言结构角色类似的词能够通过特征空间扰动的对抗训练分离开, 从而使得词向量的质量更好。

4.3 基于蒸馏的防御

Papernot 等人^[31]提出可以用蒸馏来防御对抗样本。该方法首先根据训练特征 X 和标签 Y 训练一个初始的深度神经网络模型 $f(X)$ 。然后用样本特征 X 和模型输出 $f(X)$ 作为新的标签训练一个架构相同、蒸馏温度为 T 的蒸馏网络 $F(X)$, 再利用网络 $F(X)$ 来进行分类或预测, 就可以有效的防御对抗样本的攻击。蒸馏温度为 T 下的输出概率如公式:

$$q_i = \frac{\exp(z_i / T)}{\sum_k \exp(z_k / T)} \quad (37)$$

作者解释蒸馏方法之所以能防御对抗样本是因为当蒸馏温度 T 比较大时, 会使每个类别产生的概率尽可能接近(当 T 无限大时, 就是均匀分布), 这会使得模型梯度减小, 从而更加平滑, 也因此降低了模型对扰动的敏感性。理论上 T 越大对扰动的抵抗能力就越强。

尽管蒸馏防御能成功防御一些基于梯度生成的小扰动对抗样本, 但该方法毕竟是一种静态防御, 无法保证面对未来更强攻击时的鲁棒性, 如 Carlini 等人^[4]提出的 CW 攻击方法就能成功攻击蒸馏防御网络。

4.4 基于可验证的防御

虽然启发式防御方法可以很好地防御一些对抗样本, 但这些防御都是与攻击有关的, 有效性也仅在有限的实验上得到验证, 而没有在理论上得到证明, 如果无法从理论上保证模型的防御性能, 启发式防御很可能被未来更强大的新攻击所打破^[100], 于是研究者们试图寻找一种理论上可证明的方法来防御对抗样本。

Gowal 等人^[38]在视觉领域提出了区间边界传播的方法, 该方法首先计算扰动传播所决定的损失函数上下界, 接着优化在最坏情况下损失函数的上界, 当最坏情况下损失上界很小时, 可保证模型对所有微小扰动都是鲁棒的。如公式:

$$\min_{\theta} \sum_{z \in D} (1 - \kappa) f(z, \theta) + \kappa u^{\text{final}}(z, \theta) \quad (38)$$

其中 z 是输入样本, θ 是模型参数, $u^{\text{final}}(z, \theta)$ 是损失函数在样本 z 处的损失上界, $0 \leq \kappa \leq 1$ 是一个平衡普通损失和损失上界的超参数。Jia 等人^[39]将区间边界传播的方法运用到了文本领域, 由于文本是离散的, 作者将界定义为包含所有词向量的最小轴对齐框。Ko 等人^[101]用类似的方法给出了用于循环神经网络鲁棒边界上下界的计算方法, Shi 等人^[102]则用界传播方法得到了 Transformer 网络结构下模型的鲁棒边界。

Cohen, Salman 等人^[41-42]在视觉领域提出了一种随机平滑的方法, 该方法可以对任意类型分类器 f 构造一个平滑分类器, 并计算可信的鲁棒半径, 从而保证模型对任意在鲁棒半径内的扰动都具有鲁棒性。具体的, 考虑一个 $x \in \mathbb{R}^d, f: x \rightarrow y$ 到的分类问题, 基础分类器为 f (如 f 可以是深度神经网络), 可以由 f 构造一个平滑的分类器 g , 平滑分类器 g 会返回基础分类器 f 在样本加入高斯噪声后最有可能的分类类别, 如公式:

$$g(x) = \arg \max_{c \in \mathcal{Y}} \mathbb{P}(f(x + \epsilon) = c) \quad (39)$$

其中 $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$ 。当最大的两个分类概率满足条件:

$$\begin{aligned} \mathbb{P}(f(x + \epsilon) = c_A) &\geq \underline{p_A} \geq \overline{p_B} \\ &\geq \max_{c \neq c_A} \mathbb{P}(f(x + \epsilon) = c) \end{aligned} \quad (40)$$

则 $g(x + \delta) = c_A$ 对任意的 $\|\delta\|_2 \leq R$ 成立, R 即为鲁棒半径, 定义如公式:

$$R = \frac{\sigma}{2} (\Phi^{-1}(\underline{p_A}) - \Phi^{-1}(\overline{p_B})) \quad (41)$$

其中 $\Phi(x)$ 是标准正态分布函数。Ye 等人^[79]将随机平滑思想扩展到文本任务, 不同于连续特征空间中加入高斯噪声进行平滑, 该方法为输入句子的所有可能词替换创建一个随机扰动集合, 通过该集合构造平滑分类器。基于随机平滑的可证明防御方法的理论基础是假设检验, 因此性能受随机扰动集合的影响极大, 如何构造好的随机集合是值得研究的问题, 另外该方法通常需要非常大量的采样才能得到置信度高的鲁棒边界。

4.5 其他防御方法

Jones 等人^[103]认为启发式防御不能保证模型面对更强大攻击的鲁棒性, 可验证防御也难以扩展到 Bert 等大型网络, 于是提出了鲁棒编码(Robust Encodings, RobEn)框架。RobEn 的核心是编码函数, 它可以将不同的单词映射到相同的编码, 如可将单词及其拼写错误映射到同一编码, 以这些编码作为输入的模型仅仅通过标准训练就能获得鲁棒性。具体的, 设 $B(x_i)$ 是单词 x_i 所有可能的扰动集合, 由于模型的鲁棒性主要来源于编码函数, 于是问题等价于寻找一个好的编码函数, 这个过程可以被描述为聚类——哪些单词和扰动词应共享编码表示。为此定义聚类簇的稳定度:

$$\text{Stab}(C) = -\sum_{i=1}^N \rho(w_i) |B_\pi(w_i)| \quad (42)$$

它衡量了扰动词被映射到同一个编码的可能性, 其

中 $\rho(w_i)$ 是单词 w_i 在具体数据分布中的频率, $B_\pi(w_i)$ 是单词的拼写错误被编码后的集合。同时定义保真度:

$$\text{Fid}(C) = -\sum_{i=1}^N \rho(w_i) \|\bar{v}_i - \bar{\mu}_{c(i)}\|^2 \quad (43)$$

它衡量了单词独立成簇的可能性, 其中 $\bar{v}_i$ 是词表中单词 w_i 对应的词向量, $\bar{\mu}_{c(i)}$ 是第 i 个簇的编码表示对应的向量。聚类过程是在稳定性和保真度之间取折中:

$$\Phi(C) = \gamma \text{Fid}(C) + (1 - \gamma) \text{Stab}(C) \quad (44)$$

有了编码函数, 模型将做出预测 $f_a(x) = f(\alpha(x))$, 所以只需要在编码后的数据 $\{(\alpha(x), y)\}$ 上进行标准训练即可。RobEn 主要针对拼写错误攻击, 同一时期 Wang 等人^[72]则针对近义词替换攻击提出了类似的近义词编码, 通过在词嵌入空间对同义词进行聚类, 并为每个聚类分配一个唯一的编码表示, 来抵御同义词替换攻击。

4.6 文本防御方法总结

纵观现有的文本防御方法, 基于检测的防御由于存在过滤或更改正常样本的可能性, 所以会使模型在正常样本上分类的准确率降低, 另一方面, 即便模型能识别对抗样本, 但过滤它们是不公平的, 因为它们对于人类来说和正常样本一样, 问题的核心还应放在模型上, 基于检测的防御治标不治本。同样的问题还存在于基于蒸馏的防御, 蒸馏防御虽然使得模型更加平滑, 但也只能防御一些基于梯度生成的小扰动对抗样本, 而无法保证面对其他攻击的有效性。基于对抗训练的防御是使用最为广泛的防御方法, 也是目前为止最有效的启发式防御方法, 但对抗训练需要大量的正常样本和对抗样本, 而生成对抗样本本身就需要大量的资源和时间, 因此该防御方法并不适用于大型网络和大规模数据集, 另外, 对抗训练本质上仍然属于数据增强, 基于对抗训练的防御通常只对加入训练的同分布对抗样本有效, 而无法保证对其他攻击生成的对抗样本的鲁棒性。

最后, 基于可验证的防御虽然理论上对所有可能的攻击均具有鲁棒性, 然而该类方法同样面临很多问题, 如边界传播类的可验证方法和网络结构有关, 可扩展性比较差; 随机平滑方法虽然与网络结构无关, 但性能依赖于扰动集合的构造。总体来说可验证防御效果目前还远不能满足实际要求, 不过可验证防御为未来文本防御的发展提供了一个不错的思路, 是未来有潜力的研究方向。表 3 给出了文本防御工作效果的对比。

表 3 文本防御方法对比
Table 3 Comparison of text defense methods

防御工作	启发式/可验证	模型	数据集	攻击方式	防御效果(准确率%)
Pruthi 等人 ^[96]	启发式	Bert	IMDB	拼写错误	73.0
Jones 等人 ^[103]	启发式	Bert	SST-2	拼写错误	80.7
Ren 等人 ^[67]	启发式	CNN&BiLSTM	IMDB	词替换	87.9
Zhu 等人 ^[99]	启发式	Bert	SST-2	嵌入扰动	93.5
Miyato 等人 ^[98]	启发式	CNN&BiLSTM	IMDB	嵌入扰动	94.1
Huang 等人 ^[40]	可验证	CNN	IMDB	词替换	78.7
Jia 等人 ^[39]	可验证	CNN	IMDB	词替换	79.7
Ye 等人 ^[43]	可验证	Bert	IMDB	词替换	81.2

5 数据集和相关工具包

为了读到这篇综述的研究者们实验的方便,这一部分整理了对抗攻击与防御领域常用的公开数据集和相关工具包。表 4 给出了这些数据集在不同工作和任务中的应用。

表 4 一些公共数据集的应用
Table 4 Application of some public datasets

数据集	工作中的应用	任务
IMDB	[23,73-74]	情感分析
SNLI	[23,88]	自然语言推理
HANS	[26]	自然语言推理
SST	[22,69]	情感分析
Amazon Review	[43]	文本分类
SQuAD	[6,81-82]	问答系统
AG's News	[66,76]	文本分类
GLUE	[99,103]	综合

(注:表中仅列出了部分代表工作。)

5.1 数据集

IMDB^①: 该数据集是由 Maas 等人^[104]从互联网影评数据中收集整理,共包括 50000 条关于电影的评论,正面评论和负面评论的个数基本相当,每条评论的平均长度接近 200 个单词,同时 IMDB 还包含额外的未标记数据可供使用。

SNLI^②: 斯坦福自然语言推理(Stanford Natural Language Inference, SNLI)数据集由斯坦福大学 Bowman 等人^[105]收集得到,其中包含 570000 个人工

手写英文句子对,针对推理前提与推理假设之间是否存在逻辑关系,人工标注了蕴含、中立和矛盾三种标签。

HANS^③: 该数据集由 McCoy 等人^[26]根据句子的启发式而设计,包含对于句子的三种启发式方式,词汇重叠启发式、子序列启发式和成分启发式,用于 NLI 任务,该数据集可以很好地评价 NLI 系统的鲁棒性。

SST^④: 斯坦福情感树库(Stanford Sentiment Treebank, SST)是斯坦福大学发布的一个情感分析数据集^[106],主要针对电影评论来做情感分类,其中 SST-2 是二分类, SST-5 是五分类, SST-5 的情感极性区分得更细致。

Amazon Review^⑤: 该数据集包含了从 1996 年开始亚马逊电商平台上用户对商品的数亿条评论数据,且每年都在更新。目前使用最多的是 2013 年的版本^[107],其中每个情感极性集均包含 180 万个训练样本和 20 万个测试样本。

SQuAD^⑥: 该数据集是斯坦福大学于推出的一个阅读理解数据集,文章选自维基百科, SQuAD1.0 版本^[108]包括 107785 个问题,以及相应的 536 篇文章, 2.0 版本^[109]在 1.0 版本的基础上新增了 50000 个人类撰写的问题,而且问题不一定有对应答案,不仅要求机器能从对应段落中找到问题的答案,还测试机器在没有对应答案时可以拒识,而不是瞎猜。

AG's News^⑦: 该数据集是超过 100 万篇的新闻数据,由一个学术新闻搜索引擎从 2000 多个新闻来源收集得到,其中的主题分类数据集由 Zhang 等

① <http://ai.stanford.edu/amaas/data/sentiment/>
② <https://nlp.stanford.edu/projects/snli/>
③ <https://github.com/tommccoyl/hans>
④ <https://nlp.stanford.edu/sentiment/index.html>
⑤ <http://snap.stanford.edu/data/web-Amazon-links.html>
⑥ <https://rajpurkar.github.io/SQuAD-explorer/>
⑦ <https://github.com/mhjabreel/CharCNN/tree/master/data/>

人^[110]从以上数据集中构建而来,它来自原始语料库中4个最大的类,每个类包含30000训练样本和1900测试样本。

GLUE⁸:通用自然语言理解平台(General Language Understanding Evaluation, GLUE)是由美国纽约大学和华盛顿大学等机构创建的多任务自然语言理解基准和分析平台^[111],GLUE 包含九项自然语言处理任务,每个任务都对应一个数据集,分别是 CoLA、SST-2、MRPC、STS-B、QQP、MNLI、QNLI、RTE 和 WNLI,涉及语义相似、自然语言推理、文本蕴含、情感分析等多个任务,是关于语言理解非常全面的数据集。

5.2 工具包

目前有 TextAttack⁹ 和 OpenAttack¹⁰ 这两个文本对抗工具包,二者均包含了一些现有的文本对抗攻击方法。其中 TextAttack 构建时间较早,由 Morris 等人^[112]开发,TextAttack 的模块化设计使研究人员可以轻松地通过现有组件的组合来构建攻击,该工具包提供了来自文献的16种对抗攻击的实现,并支持各种模型和数据集,而且支持包括 Bert 和其他基于 Transformer 的模型以及所有 GLUE 任务,TextAttack 还包括数据增强和对抗训练模块,可以使用对抗训练的组件来提高模型的准确性和鲁棒性。OpenAttack 构建时间较晚,由清华大学 Zeng 等人^[113]开发,OpenAttack 构建了涵盖所有攻击类型的12种典型攻击模型,其高度包容的模块化设计不仅支持快速利用现有攻击模型,而且具有极大的灵活性和可扩展性,研究者可以使用它比较和评估攻击模型,评价被攻击模型的鲁棒性,协助开发新的攻击模型等。

这些工具包使得学习对抗攻击与防御变的方便快捷,任何人都可以仅用几行代码就尝试对模型进行攻击,并进行数据增强和对抗训练实验。

6 总结与展望

尽管 DNN 在各种各样的自然语言处理任务上都具有很高的性能,但它们很容易受到对抗样本的攻击。本文对文本领域的对抗攻击和防御工作做了调研,在第2章介绍了文本对抗攻击与防御的基本分类,第3章介绍了关于攻击的相关工作,第4章介绍了关于防御的相关工作,第5章介绍了数据集和相关工具包。

6.1 现有方法回顾

对于现有攻击方法,从攻击粒度来看,单词级

攻击在对抗样本质量和攻击成功率方面有更好的表现,也因此单词级攻击使用的最为广泛。在攻击的实现上,可分为两个阶段,第一个阶段计算单词对分类结果的贡献度并进行排序,第二个阶段针对分类贡献度大的词进行攻击;从攻击者掌握模型知识的角度来看,白盒类攻击通常需要模型的梯度信息,付出的代价更大但往往能生成更好的对抗样本,黑盒类攻击总体效果不如白盒但通过模型反馈也能生成质量不错的对抗样本;从生成对抗样本的迁移性来看,目标攻击生成的对抗样本在迁移性上不如非目标攻击。

现有防御方法使用最为广泛的是对抗训练,但对抗训练需要大量的正常样本和对抗样本,这极大地增加了训练时间和所需资源,另外,对抗训练本质上属于数据增强,基于对抗训练的防御通常也只能防御同类型对抗样本。可验证防御试图在理论上保证模型对所有微小扰动的鲁棒性,其中区间边界传播方法优化在最坏情况下的损失上界,但该方法与模型有关,很难扩展到其它网络上去。随机平滑方法构造的平滑分类器利用了神经网络的判别能力,使样本在噪声下也能很好的分类,该方法与结构无关且能扩展到任意网络,但随机平滑方法基于假设检验,通常需要非常大量的采样才能得到高置信度的结果。总体而言,实现鲁棒的基于 DNN 的自然语言处理系统还任重道远。

6.2 研究展望

攻击和防御是相辅相成的,攻击旨在发现模型的弱点,而防御旨在使模型抵御攻击。对攻击而言,现有攻击方法虽然能成功攻击模型,而且句子语义也在一定程度上得到了保留,但生成的新句子对人类而言还是会有些不自然,因此,对于对抗样本流畅性和自然性的评估是未来应该关注的点。另外,现有攻击方法在生成对抗样本时普遍需要进行大量搜索,未来可以考虑从对抗样本的生成效率上进一步提升对抗样本的质量,同时,现有方法的候选搜索词表往往还需要人类参与构建,未来这个过程应尽量减少人工干预,实现自动化。

对防御而言,现有防御方法仍然很难保证 DNN 的鲁棒性,这主要是因为,攻击算法只要能够成功使 DNN 模型出错即可,而一个有效的防御算法则需要去抵御所有可能的攻击。基于对抗训练的防御目前使用的最为广泛,是迄今为止最有效的启发式防

⑧ <https://gluebenchmark.com/tasks>

⑨ <https://github.com/QData/TextAttack>

⑩ <https://github.com/thunlp/OpenAttack>

御方法,但是其计算代价比较昂贵,能防御的对抗样本也比较局限。可验证防御虽然理论上能防御各种攻击生成的对抗样本,但其准确性和有效性目前还远远不能满足实际要求,而且也不能扩展到非常深的神经网络和大型数据集。不过可验证防御在理论上为防御工作指明了一条道路,未来的防御工作应该向着可证明防御发展,鲁棒的模型需要做到防御各种对抗样本的攻击。此外,随着目前攻击方法的演变,一些在自然语言推理上的工作发现模型的脆弱性和启发式有关,模型更倾向于服从训练数据所表现出来的统计特性,而不是学习更根本的特征,也即模型实际上只学到了浅层特征,这也许是模型脆弱性的根本原因,如何使 DNN 这类统计模型避免只学习浅层特征是一个悬而未决的问题,也是未来防御工作应重点考虑的点。

参考文献

- [1] Szegedy C, Zaremba W, Sutskever I, et al. Intriguing Properties of Neural Networks[EB/OL]. 2013: arXiv: 1312.6199. <https://arxiv.org/abs/1312.6199>.
- [2] Goodfellow I J, Shlens J, Szegedy C. Explaining and Harnessing Adversarial Examples[EB/OL]. 2014: arXiv: 1412.6572. <https://arxiv.org/abs/1412.6572>
- [3] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks[EB/OL]. 2016. ArXiv Preprint ArXiv:1501.04599.
- [4] Carlini N, Wagner D. Towards Evaluating the Robustness of Neural Networks[EB/OL]. 2016: arXiv: 1608.04644. <https://arxiv.org/abs/1608.04644>
- [5] Papernot N, McDaniel P, Swami A, et al. Crafting Adversarial Input Sequences for Recurrent Neural Networks[C]. *MILCOM 2016 - 2016 IEEE Military Communications Conference*, 2016: 49-54.
- [6] Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems[EB/OL]. 2017: ArXiv Preprint ArXiv:1707.07328
- [7] Akhtar N, Mian A. Threat of Adversarial Attacks on Deep Learning in Computer Vision: A Survey[EB/OL]. 2018: arXiv: 1801.00553. <https://arxiv.org/abs/1801.00553>
- [8] Gilmer J, Adams R P, Goodfellow I, et al. Motivating the Rules of the Game for Adversarial Example Research[EB/OL]. 2018: arXiv: 1807.06732. <https://arxiv.org/abs/1807.06732>
- [9] Liu Q, Li P, Zhao W T, et al. A Survey on Security Threats and Defensive Techniques of Machine Learning: A Data Driven View[J]. *IEEE Access*, 2018, 6: 12103-12117.
- [10] Yuan X Y, He P, Zhu Q L, et al. Adversarial Examples: Attacks and Defenses for Deep Learning[EB/OL]. 2017: arXiv: 1712.07107. <https://arxiv.org/abs/1712.07107>
- [11] Zhang J L, Li C. Adversarial Examples: Opportunities and Challenges[J]. *IEEE Transactions on Neural Networks and Learning Systems*, 2020, 31(7): 2578-2593.
- [12] Wang W Q, Wang R, Wang L N, et al. Towards a Robust Deep Neural Network in Texts: A Survey[EB/OL]. 2019: arXiv: 1902.07285. <https://arxiv.org/abs/1902.07285>
- [13] Xu H, Ma Y, Liu H C, et al. Adversarial Attacks and Defenses in Images, Graphs and Text: A Review[EB/OL]. 2019: arXiv: 1909.08072. <https://arxiv.org/abs/1909.08072>
- [14] Belinkov Y, Glass J. Analysis Methods in Neural Language Processing: A Survey[EB/OL]. 2018: arXiv: 1812.08951. <https://arxiv.org/abs/1812.08951>
- [15] Zhang W E, Sheng Q Z, Alhazmi A, et al. Adversarial Attacks on Deep Learning Models in Natural Language Processing: A Survey[EB/OL]. 2019: arXiv: 1901.06796. <https://arxiv.org/abs/1901.06796>
- [16] Hochreiter S, Schmidhuber J. Long Short-Term Memory[J]. *Neural Computation*, 1997, 9(8): 1735-1780.
- [17] Cho K, van Merriënboer B, Gulcehre C, et al. Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation[EB/OL]. 2014: arXiv: 1406.1078. <https://arxiv.org/abs/1406.1078>
- [18] Kim Y. Convolutional Neural Networks for Sentence Classification[EB/OL]. 2014: arXiv: 1408.5882. <https://arxiv.org/abs/1408.5882>
- [19] Ilyas A, Santurkar S, Tsipras D, et al. Adversarial Examples are not Bugs, they are Features[EB/OL]. 2019: arXiv: 1905.02175. <https://arxiv.org/abs/1905.02175>
- [20] Liu Y P, Chen X Y, Liu C, et al. Delving into Transferable Adversarial Examples and Black-Box Attacks[EB/OL]. 2016: arXiv: 1611.02770. <https://arxiv.org/abs/1611.02770>
- [21] Naseer M, Khan S H, Rahman S, et al. Task-Generalizable Adversarial Attack Based on Perceptual Metric[EB/OL]. 2018: arXiv: 1811.09020. <https://arxiv.org/abs/1811.09020>
- [22] Ebrahimi J, Rao A Y, Lowd D, et al. HotFlip: White-Box Adversarial Examples for Text Classification[EB/OL]. 2017: arXiv: 1712.06751. <https://arxiv.org/abs/1712.06751>
- [23] Alzantot M, Sharma Y, Elgohary A, et al. Generating Natural Language Adversarial Examples[EB/OL]. 2018: arXiv: 1804.07998. <https://arxiv.org/abs/1804.07998>
- [24] Zhang H Z, Zhou H, Miao N, et al. Generating Fluent Adversarial Examples for Natural Languages[EB/OL]. 2020: arXiv: 2007.06174. <https://arxiv.org/abs/2007.06174>
- [25] Ribeiro M T, Singh S, Guestrin C. Semantically Equivalent Adversarial Rules for Debugging NLP Models[C]. *The 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018: 856-865.
- [26] McCoy T, Pavlick E, Linzen T. Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference[C]. *The 57th Annual Meeting of the Association for Computational Linguistics*, 2019: 3428-3448.
- [27] Carlini N, Athalye A, Papernot N, et al. On Evaluating Adversarial Robustness[EB/OL]. 2019: arXiv: 1902.06705. <https://arxiv.org/abs/1902.06705>
- [28] Madry A, Makelov A, Schmidt L, et al. Towards Deep Learning Models Resistant to Adversarial Attacks[EB/OL]. 2017: arXiv: 1706.06083. <https://arxiv.org/abs/1706.06083>
- [29] Tsipras D, Santurkar S, Engstrom L, et al. Robustness may be at Odds with Accuracy[EB/OL]. 2018: arXiv: 1805.12152. <https://arxiv.org/abs/1805.12152>

- /arxiv.org/abs/1805.12152
- [30] Xie C H, Tan M X, Gong B Q, et al. Adversarial Examples Improve Image Recognition[EB/OL]. 2019: arXiv: 1911.09665. <https://arxiv.org/abs/1911.09665>
 - [31] Papernot N, McDaniel P, Wu X, et al. Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks[EB/OL]. 2015: arXiv: 1511.04508. <https://arxiv.org/abs/1511.04508>
 - [32] Hinton G, Vinyals O, Dean J. Distilling the Knowledge in a Neural Network[EB/OL]. 2015: arXiv: 1503.02531. <https://arxiv.org/abs/1503.02531>
 - [33] Tjeng V, Xiao K, Tedrake R. Evaluating Robustness of Neural Networks with Mixed Integer Programming[EB/OL]. 2017: arXiv: 1711.07356. <https://arxiv.org/abs/1711.07356>
 - [34] Cheng C H, Nührenberg G, Ruess H. Maximum Resilience of Artificial Neural Networks[EB/OL]. 2017: arXiv: 1705.01040. <https://arxiv.org/abs/1705.01040>
 - [35] Bunel R, Turkaslan I, Torr P H S, et al. A Unified View of Piecewise Linear Neural Network Verification[EB/OL]. 2017: arXiv: 1711.00455. <https://arxiv.org/abs/1711.00455>
 - [36] Katz G, Barrett C, Dill D, et al. Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks[EB/OL]. 2017: arXiv: 1702.01135. <https://arxiv.org/abs/1702.01135>
 - [37] Nicholas Carlini, Guy Katz, Clark Barrett, et al. Ground-truth adversarial examples[EB/OL]. 2017: ArXiv Preprint ArXiv:1709.10207.
 - [38] Goyal S, Dvijotham K, Stanforth R, et al. On the Effectiveness of Interval Bound Propagation for Training Verifiably Robust Models[EB/OL]. 2018: arXiv: 1810.12715. <https://arxiv.org/abs/1810.12715>
 - [39] Jia R, Raghunathan A, Göksel K, et al. Certified Robustness to Adversarial Word Substitutions[EB/OL]. 2019: arXiv: 1909.00986. <https://arxiv.org/abs/1909.00986>
 - [40] Huang P S, Stanforth R, Welbl J, et al. Achieving Verified Robustness to Symbol Substitutions via Interval Bound Propagation[EB/OL]. 2019: arXiv: 1909.01492. <https://arxiv.org/abs/1909.01492>
 - [41] Salman H, Yang G, Li J, et al. Provably Robust Deep Learning via Adversarially Trained Smoothed Classifiers[EB/OL]. 2019: arXiv: 1906.04584. <https://arxiv.org/abs/1906.04584>
 - [42] Cohen J M, Rosenfeld E, Kolter J Z. Certified Adversarial Robustness via Randomized Smoothing[EB/OL]. 2019: arXiv: 1902.02918. <https://arxiv.org/abs/1902.02918>
 - [43] Ye M, Gong C Y, Liu Q. SAFER: A Structure-Free Approach for Certified Robustness to Adversarial Word Substitutions[EB/OL]. 2020: arXiv: 2005.14424. <https://arxiv.org/abs/2005.14424>
 - [44] Glorot X, Bordes A, Bengio Y. Deep sparse rectifier neural networks[C]. *The fourteenth international conference on artificial intelligence and statistics*, 2011: 315-323.
 - [45] Sinha A, Namkoong H, Volpi R, et al. Certifying some Distributional Robustness with Principled Adversarial Training[EB/OL]. 2017: arXiv: 1710.10571. <https://arxiv.org/abs/1710.10571>
 - [46] Peck J, Roels J, Goossens B, et al. Lower Bounds on the Robustness to Adversarial Perturbations[C]. *The 31st International Conference on Neural Information Processing Systems*, 2017: 804-813.
 - [47] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Certified defenses against adversarial examples[EB/OL]. 2019: ArXiv Preprint ArXiv:1810.09344.
 - [48] Weng T W, Zhang H, Chen H G, et al. Towards Fast Computation of Certified Robustness for ReLU Networks[EB/OL]. 2018: arXiv: 1804.09699. <https://arxiv.org/abs/1804.09699>
 - [49] Gong Z T, Wang W L, Li B, et al. Adversarial Texts with Gradient Methods[EB/OL]. 2018: arXiv: 1801.07175. <https://arxiv.org/abs/1801.07175>
 - [50] Liang B, Li H C, Su M Q, et al. Deep Text Classification can be Fooled[C]. *The Twenty-Seventh International Joint Conference on Artificial Intelligence*, 2018: 4209-4215.
 - [51] Mikolov T, Sutskever I, Chen K, et al. Distributed Representations of Words and Phrases and Their Compositionality[EB/OL]. 2013: arXiv: 1310.4546. <https://arxiv.org/abs/1310.4546>
 - [52] Subramanian S, Trischler A, Bengio Y, et al. Learning General Purpose Distributed Sentence Representations via Large Scale Multi-Task Learning[EB/OL]. 2018: arXiv: 1804.00079. <https://arxiv.org/abs/1804.00079>
 - [53] Cer D, Yang Y F, Kong S Y, et al. Universal Sentence Encoder[EB/OL]. 2018: arXiv: 1803.11175. <https://arxiv.org/abs/1803.11175>
 - [54] Vaswani A, Shazeer N, Parmar N, et al. Attention is all You Need[EB/OL]. 2017: arXiv: 1706.03762. <https://arxiv.org/abs/1706.03762>
 - [55] Iyyer M, Manjunatha V, Boyd-Graber J, et al. Deep Unordered Composition Rivals Syntactic Methods for Text Classification[C]. *The 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2015: 1681-1691.
 - [56] Rubner Y, Tomasi C, Guibas L J. A Metric for Distributions with Applications to Image Databases[C]. *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, 2002: 59-66.
 - [57] Kusner M J, Sun Y, Kolkin N I, et al. From Word Embeddings to Document Distances[J]. *32nd International Conference on Machine Learning, ICML 2015*, 2015, 2: 957-966.
 - [58] Levenshtein V I. Binary codes capable of correcting deletions, insertions, and reversals[C]. *Soviet physics doklady*. 1966, 10(8): 707-710.
 - [59] Niu Y L, Qiao C, Li H, et al. Word Embedding Based Edit Distance[EB/OL]. 2018: arXiv: 1810.10752. <https://arxiv.org/abs/1810.10752>
 - [60] Belinkov Y, Bisk Y. Synthetic and Natural Noise both Break Neural Machine Translation[EB/OL]. 2017: arXiv: 1711.02173. <https://arxiv.org/abs/1711.02173>
 - [61] Eger S, Şahin G G, Rücklé A, et al. Text Processing Like Humans Do: Visually Attacking and Shielding NLP Systems[EB/OL]. 2019: arXiv: 1903.11508. <https://arxiv.org/abs/1903.11508>
 - [62] Gil Y, Chai Y, Gorodissky O, et al. White-to-Black: Efficient Distillation of Black-Box Adversarial Attacks[EB/OL]. 2019: arXiv: 1904.02405. <https://arxiv.org/abs/1904.02405>
 - [63] Ebrahimi J, Lowd D, Dou D J. On Adversarial Examples for Character-Level Neural Machine Translation[EB/OL]. 2018: arXiv: 1806.09030. <https://arxiv.org/abs/1806.09030>

- [64] Gao J, Lanchantin J, Soffa M L, et al. Black-Box Generation of Adversarial Text Sequences to Evade Deep Learning Classifiers[EB/OL]. 2018: arXiv: 1801.04354. <https://arxiv.org/abs/1801.04354>
- [65] Cheng M H, Yi J F, Chen P Y, et al. Seq2Sick: Evaluating the Robustness of Sequence-to-Sequence Models with Adversarial Examples[EB/OL]. 2018: arXiv: 1803.01128. <https://arxiv.org/abs/1803.01128>
- [66] 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[EB/OL]. 2019: arXiv: 1907.11932. <https://arxiv.org/abs/1907.11932>
- [67] Ren S H, Deng Y H, He K, et al. Generating Natural Language Adversarial Examples through Probability Weighted Word Saliency[C]. *The 57th Annual Meeting of the Association for Computational Linguistics*, 2019: 1085-1097.
- [68] Samanta S, Mehta S. Towards Crafting Text Adversarial Samples[EB/OL]. 2017: arXiv: 1707.02812. <https://arxiv.org/abs/1707.02812>
- [69] Zang Y A, Qi F C, Yang C H, et al. Word-Level Textual Adversarial Attacking as Combinatorial Optimization[C]. *The 58th Annual Meeting of the Association for Computational Linguistics*, 2020: 6066-6080.
- [70] Papernot N, McDaniel P, Jha S, et al. The Limitations of Deep Learning in Adversarial Settings[EB/OL]. 2015: arXiv: 1511.07528. <https://arxiv.org/abs/1511.07528>
- [71] Kuleshov Volodymyr, Thakoor Shantanu, Lau Tingfung, et al. Adversarial examples for natural language classification problems. <https://openreview.net/forum?id=r1QZ3zbAZ>. Feb. 2018.
- [72] Xiaosen Wang, Hao Jin, and Kun He. Natural language adversarial attacks and defenses in word level[EB/OL]. 2020: ArXiv Preprint ArXiv:1909.06723.
- [73] Sato M, Suzuki J, Shindo H, et al. Interpretable Adversarial Perturbation in Input Embedding Space for Text[EB/OL]. 2018: arXiv: 1805.02917. <https://arxiv.org/abs/1805.02917>
- [74] Li J F, Ji S L, Du T Y, et al. TextBugger: Generating Adversarial Text Against Real-World Applications[EB/OL]. 2018: arXiv: 1812.05271. <https://arxiv.org/abs/1812.05271>
- [75] Yang P, Chen J B, Hsieh C J, et al. Greedy Attack and Gumbel Attack: Generating Adversarial Examples for Discrete Data[EB/OL]. 2018: arXiv: 1805.12316. <https://arxiv.org/abs/1805.12316>
- [76] Li L Y, Ma R T, Guo Q P, et al. BERT-ATTACK: Adversarial Attack Against BERT Using BERT[EB/OL]. 2020: arXiv: 2004.09984. <https://arxiv.org/abs/2004.09984>
- [77] Devlin J, Chang M W, Lee K, et al. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding[EB/OL]. 2018: arXiv: 1810.04805. <https://arxiv.org/abs/1810.04805>
- [78] Garg S, Ramakrishnan G. BAE: BERT-Based Adversarial Examples for Text Classification[EB/OL]. 2020: arXiv: 2004.01970. <https://arxiv.org/abs/2004.01970>
- [79] Iyyer M, Wieting J, Gimpel K, et al. Adversarial Example Generation with Syntactically Controlled Paraphrase Networks[EB/OL]. 2018: arXiv: 1804.06059. <https://arxiv.org/abs/1804.06059>
- [80] Zhao Z L, Dua D, Singh S. Generating Natural Adversarial Examples[EB/OL]. 2017: arXiv: 1710.11342. <https://arxiv.org/abs/1710.11342>
- [81] Gan W C, Ng H T. Improving the Robustness of Question Answering Systems to Question Paraphrasing[C]. *The 57th Annual Meeting of the Association for Computational Linguistics*, 2019: 6065-6075.
- [82] Wang Y C, Bansal M. Robust Machine Comprehension Models via Adversarial Training[EB/OL]. 2018: arXiv: 1804.06473. <https://arxiv.org/abs/1804.06473>
- [83] Mudrakarta P K, Taly A, Sundararajan M, et al. Did the Model Understand the Question? [EB/OL]. 2018: arXiv: 1805.05492. <https://arxiv.org/abs/1805.05492>
- [84] See A, Liu P J, Manning C D. Get to the Point: Summarization with Pointer-Generator Networks[EB/OL]. 2017: arXiv: 1704.04368. <https://arxiv.org/abs/1704.04368>
- [85] Minervini P, Riedel S. Adversarially Regularising Neural NLI Models to Integrate Logical Background Knowledge[EB/OL]. 2018: arXiv: 1808.08609. <https://arxiv.org/abs/1808.08609>
- [86] Bengio Y, Schwenk H, Senécal J S, et al. Neural Probabilistic Language Models[M]. *Innovations in Machine Learning*. Berlin/Heidelberg: Springer-Verlag, 2006: 137-186.
- [87] Moosavi-Dezfooli S M, Fawzi A, Fawzi O, et al. Universal Adversarial Perturbations[EB/OL]. 2016: arXiv: 1610.08401. <https://arxiv.org/abs/1610.08401>
- [88] Wallace E, Feng S, Kandpal N, et al. Universal Adversarial Triggers for Attacking and Analyzing NLP[EB/OL]. 2019: arXiv: 1908.07125. <https://arxiv.org/abs/1908.07125>
- [89] Behjati M, Moosavi-Dezfooli S M, Baghshah M S, et al. Universal Adversarial Attacks on Text Classifiers[C]. *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019: 7345-7349.
- [90] Song L W, Yu X W, Peng H T, et al. Universal Adversarial Attacks with Natural Triggers for Text Classification[EB/OL]. 2020: arXiv: 2005.00174. <https://arxiv.org/abs/2005.00174>
- [91] Zhao J, Kim Y, Zhang K, et al. Adversarially Regularized Autoencoders[EB/OL]. 2017: arXiv: 1706.04223. <https://arxiv.org/abs/1706.04223>
- [92] Kingma D P, Welling M. Auto-Encoding Variational Bayes[EB/OL]. 2013: arXiv: 1312.6114. <https://arxiv.org/abs/1312.6114>
- [93] Goodfellow I J, Pouget-Abadie J, Mirza M, et al. Generative Adversarial Networks[EB/OL]. 2014: arXiv: 1406.2661. <https://arxiv.org/abs/1406.2661>
- [94] Metzen J H, Genewein T, Fischer V, et al. On Detecting Adversarial Perturbations[EB/OL]. 2017: arXiv: 1702.04267. <https://arxiv.org/abs/1702.04267>
- [95] Zhou Y C, Jiang J Y, Chang K W, et al. Learning to Discriminate Perturbations for Blocking Adversarial Attacks in Text Classification[EB/OL]. 2019: arXiv: 1909.03084. <https://arxiv.org/abs/1909.03084>
- [96] Pruthi D, Dhingra B, Lipton Z C. Combating Adversarial Misspellings with Robust Word Recognition[EB/OL]. 2019: arXiv: 1905.11268. <https://arxiv.org/abs/1905.11268>
- [97] Cheng Y, Jiang L, Macherey W. Robust Neural Machine Translation with Doubly Adversarial Inputs[EB/OL]. 2019: arXiv: 1906.

02443. <https://arxiv.org/abs/1906.02443>
- [98] Miyato T, Dai A M, Goodfellow I. Adversarial Training Methods for Semi-Supervised Text Classification[EB/OL]. 2016: arXiv: 1605.07725. <https://arxiv.org/abs/1605.07725>
- [99] Zhu C, Cheng Y, Gan Z, et al. FreeLB: Enhanced Adversarial Training for Natural Language Understanding[EB/OL]. 2019: arXiv: 1909.11764. <https://arxiv.org/abs/1909.11764>
- [100] Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods[EB/OL]. 2017: ArXiv Preprint ArXiv:1608.04644.
- [101] Ko C Y, Lyu Z Y, Weng T W, et al. POPQORN: Quantifying Robustness of Recurrent Neural Networks[EB/OL]. 2019: arXiv: 1905.07387. <https://arxiv.org/abs/1905.07387>
- [102] Zhouxing Shi, Huan Zhang, Kai-Wei Chang, et al. Robustness verification for transformers[EB/OL]. 2019: ArXiv Preprint ArXiv:1909.01492.
- [103] Jones E, Jia R, Raghunathan A, et al. Robust Encodings: A Framework for Combating Adversarial Typos[EB/OL]. 2020: arXiv: 2005.01229. <https://arxiv.org/abs/2005.01229>
- [104] Maas A L, Daly R E, Pham P T, et al. Learning Word Vectors for Sentiment Analysis[C]. *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, 2011: 142-150.
- [105] Bowman S R, Angeli G, Potts C, et al. A Large Annotated Corpus for Learning Natural Language Inference[EB/OL]. 2015: arXiv: 1508.05326. <https://arxiv.org/abs/1508.05326>
- [106] Socher R, Perelygin A, Wu J Y, et al. Recursive Deep Models for Semantic Compositionality over a Sentiment Treebank[J]. *EMNLP 2013 - 2013 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 2013: 1631-1642.
- [107] McAuley J, Leskovec J. Hidden Factors and Hidden Topics: Understanding Rating Dimensions with Review Text[C]. *The 7th ACM conference on Recommender systems*, 2013: 165-172.
- [108] Rajpurkar P, Zhang J, Lopyrev K, et al. SQuAD: 100, 000+ Questions for Machine Comprehension of Text[EB/OL]. 2016: arXiv: 1606.05250. <https://arxiv.org/abs/1606.05250>
- [109] Rajpurkar P, Jia R, Liang P. Know what You Don't Know: Unanswerable Questions for SQuAD[EB/OL]. 2018: arXiv: 1806.03822. <https://arxiv.org/abs/1806.03822>
- [110] Zhang X, Zhao J B, LeCun Y. Character-Level Convolutional Networks for Text Classification[EB/OL]. 2015: arXiv: 1509.01626. <https://arxiv.org/abs/1509.01626>
- [111] Wang A, Singh A, Michael J, et al. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding[C]. *The 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, 2018: 353-355.
- [112] Morris J X, Lifland E, Yoo J Y, et al. TextAttack: A Framework for Adversarial Attacks, Data Augmentation, and Adversarial Training in NLP[EB/OL]. 2020: arXiv: 2005.05909. <https://arxiv.org/abs/2005.05909>
- [113] Zeng G Y, Qi F C, Zhou Q R, et al. OpenAttack: An Open-Source Textual Adversarial Attack Toolkit[EB/OL]. 2020: arXiv: 2009.09191. <https://arxiv.org/abs/2009.09191>



梁宇航 于 2019 年在北京信息科技大学信息与计算科学专业获得理学学士学位, 现在中国科学院信息工程研究所攻读硕士学位。研究兴趣包括: 文本对抗攻击与防御。Email: liangyuhang@iie.ac.cn



林政 于 2014 年在中国科学院计算技术研究所获博士学位。现任中国科学院信息工程研究所第三研究室研究员。研究方向包括: 情感分析、文本生成、模型压缩等。Email: linzheng@iie.ac.cn



王雷 于 2012 年在中国科学院获博士学位, 现任中国科学院信息工程研究所第三研究室高级工程师。研究领域包括: 密码工程与应用、身份管理与网络信任技术、大数据智能处理等。Email: wanglei@iie.ac.cn



何原野 于 2017 年在云南大学和元智大学获得硕士学位, 现任中国科学院信息工程研究所第三研究室工程师。研究兴趣包括: 情感分析等。Email: heyuanye@iie.ac.cn



王伟平 于 2007 年在哈尔滨工业大学获得博士学位, 现任中国科学院信息工程研究所第三研究室研究员、博士生导师。研究领域包括: 大数据, 人工智能, 数据安全等。Email: wangweiping@iie.ac.cn