Beijing Forest Studio

北京理工大学信息系统及安全对抗实验中心

文本分类硬标签黑盒模型的对抗样本生成方法研究

硕士研究生 郑俊怡

2024年11月17日

问题回溯

- 相关内容
  - 2023.10.22 邵思源《面向NIDS的流量对抗样本检测技术》
  - 2023.05.28 程瑶《单词级文本对抗攻击》

内容提要

- 预期收获
- 题目内涵解析
- 研究背景与意义
- 研究历史与现状
- 知识基础
- 算法原理
  - PAT
  - FastTextDodger
- 特点总结与工作展望
- 参考文献

预期收获

- 预期收获
  - 掌握硬标签黑盒和对抗样本生成的基本概念
  - 理解文本分类硬标签黑盒模型的对抗样本生成方法的基本原理
  - 了解现有方法的缺陷以及未来发展方向

内涵解析与研究目标

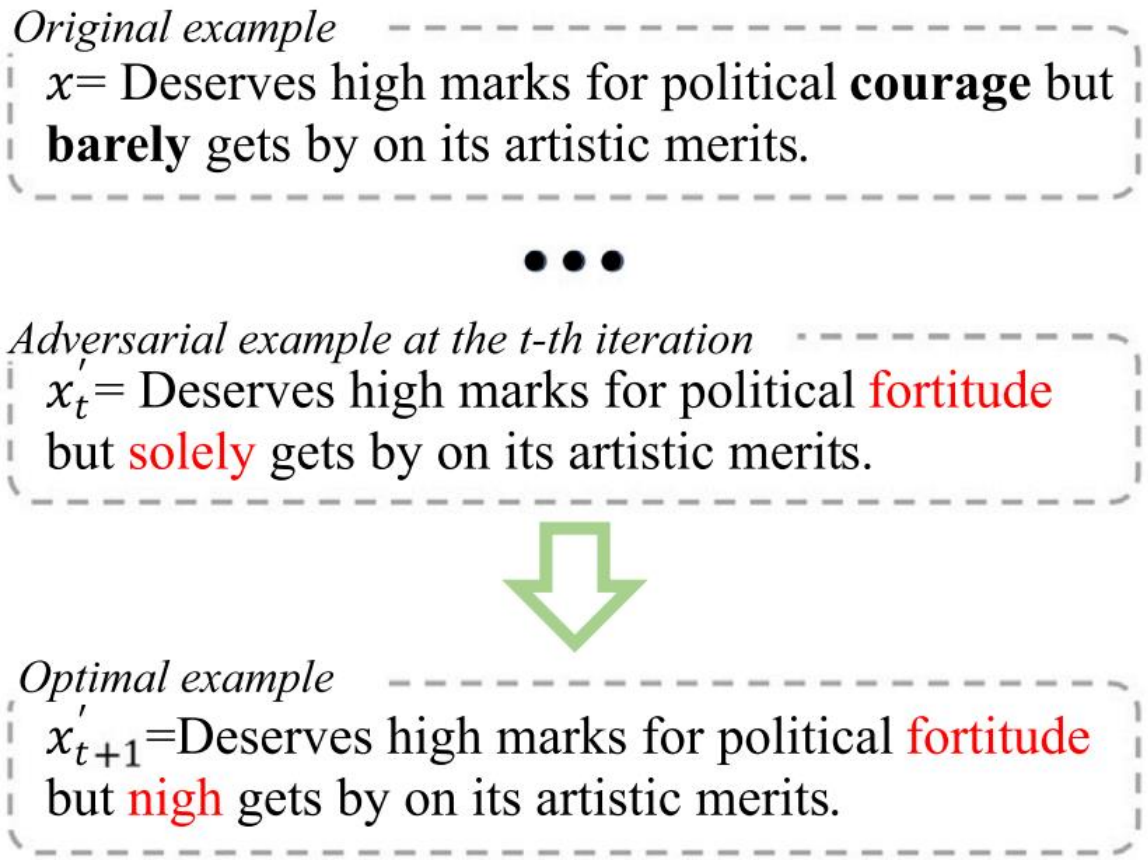
- 内涵解析
  - 对抗样本生成：向原始样本中添加人类难以察觉的微小扰动，以生成对抗样本，使得深度学习模型输出错误结果
  - 硬标签黑盒：黑盒指无法获取模型具体参数和架构，其中硬标签指仅能获取预测标签而无法获取概率分数

- 研究目标
  - 面向文本分类模型的安全性测试
  - 硬标签黑盒设置更符合现实攻击环境
  - 提升模型的鲁棒性和安全性

研究背景与意义

- 研究背景
  - 文本分类模型应用广泛
    - 恶意代码检测
    - 垃圾邮件过滤
    - 情感分析
    - 新闻分类
  - 文本分类模型易受对抗样本攻击影响
  - 硬标签黑盒设置更符合现实攻击环境

- 研究意义
  - 分析模型的潜在安全风险
  - 为加强模型鲁棒性提供方向



知识基础　文本分类模型的对抗样本生成

- 根据扰动粒度
  - 句子级
    - 改述、编码后重新解码、添加无关句子
  - 单词级
    - 词替换、词添加、词删除
  - 字符级
    - 字符的添加、删除、替换、交换顺序等
- 根据受害模型可见性
  - 白盒
  - 黑盒
    - 软标签（基于分数）
    - 硬标签（基于决策）

情感分析任务

原本句子：I love Tsinghua University.
模型输出：Positive

句子级别：I love Tsinghua University. This is cool.
模型输出：Negative

单词级别：I admire Tsinghua University.
模型输出：Negative

字符级别：I 1ove Tsinghua University.
模型输出：Negative

x

"panda"
57.7% confidence

x

"panda"

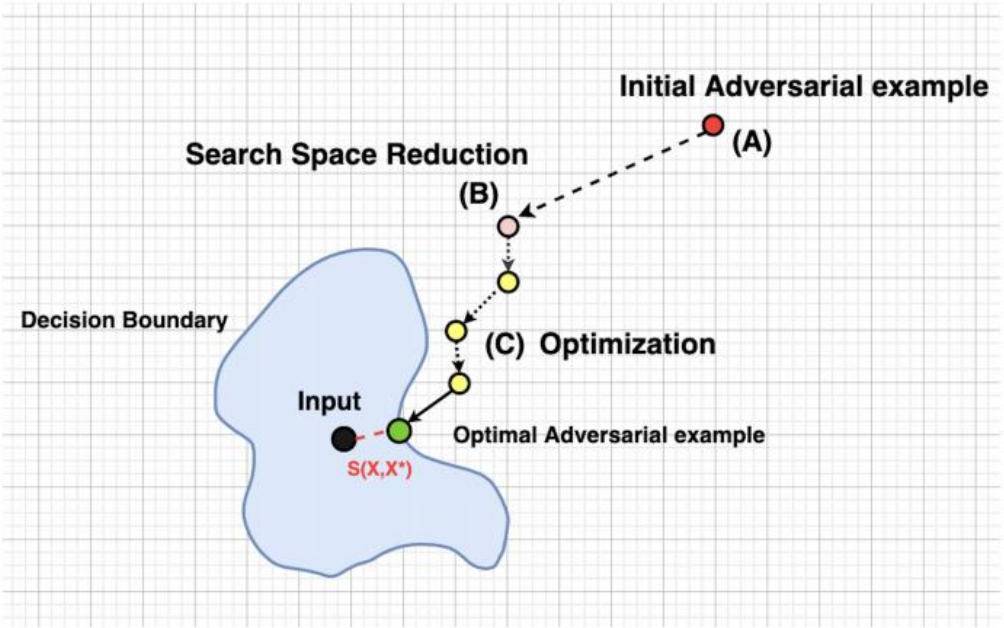
知识基础　同义词集

- 同义词集
  - 同义词替换：在保持文本原意不变的前提下，将文本中的某个或某些单词替换为它们的同义词
  - 同义词：表达意思完全一致，在绝大多数语境中都可以相互替换，同时对上下文也不会产生影响的词语
  - 同义词集：把同义词组织到一个无序的集合形成的词集

类别	实例	问题
语言知识库	WordNet	忽略了一词多义
	HowNet	许多单词无法找到同义词
词嵌入	反拟合词嵌入空间	获取的很多单词并不是语义同义词
掩码语言模型	BERT	获取的很多单词并不是语义同义词
	RoBERTa	获取的很多单词并不是语义同义词

文本分类硬标签黑盒模型的对抗样本生成

- 对抗样本生成
  - 初始化
    - 随机同义词替换
      - 随机替换位置
      - 随机同义词
    - 分层注意力网络（FastTextDodger）
    - 全局替换（HyGloadAttack）
  - 优化
    - 词嵌入空间
    - 语义相似度
    - 其他



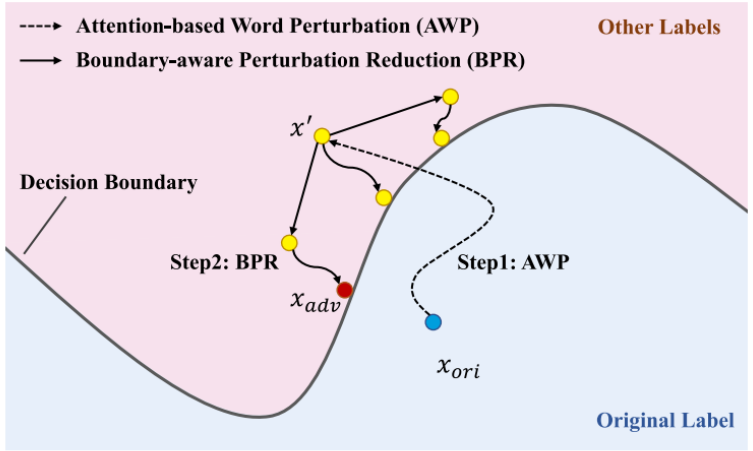


Fig. 2. Insight of FastTextDodger. x_{ori} crosses the decision boundary to obtain x' in step 1, and x' finds the adversarial text x_{adv} that is closer to x_{ori} in step 2.

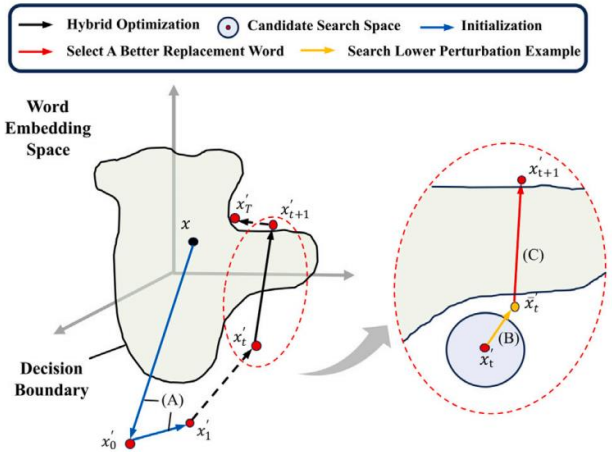


Fig. 2. Overview of HyGloadAttack.

知识基础 评价指标

- 评价指标
  - 攻击成功率
    - Attack Success Rate (ASR)
    - After Attack Accuracy (Acc)
  - 对抗样本质量
    - 语义相似度
    - 单词替换率
    - 语法错误率
  - 攻击效率
    - 查询次数

$$ASR = \frac{AttackSample_{success}}{AttackSample_{total}} \qquad Acc = \frac{Sample_{success}}{Sample_{total}}$$

Surrey poised to sign Harbhajan
Surrey are waiting for approval from the Board of Control for Cricket in India before announcing Harbhajan Singh as an overseas signing for 2005

Surrey 准备签下 Harbhajan
Surrey 正在等待印度板球控制委员会的批准，然后宣布 Harbhajan Singh 作为 2005 年的海外签约

语义相似度：75.7%
单词替换率：6.897%

Surrey poised to sign Harbhajan
Surrey are waiting for approval from the Board of Control for Predatory in India before announcing Harbhajan Singh as an overseas subscribers for 2005

Surrey 准备签下 Harbhajan
Surrey 正在等待印度掠夺控制委员会的批准，然后宣布 Harbhajan Singh 成为 2005 年的海外订户

Label：体育

Label：商业

研究历史

Ye M提出TextHoaxer，在词嵌入空间中使用语义相似度、成对扰动约束、稀疏性约束构建损失函数，使用基于梯度的策略降低损失函数，优化对抗样本

Yu Z提出TextHacker，构建权重表，使用单词替换所引起的预测更新权重表，使用权重表指导局部搜索策略

Ye M提出PAT，使用原型估计方法估计决策边界，将单词和决策边界的距离作为基准，选择保持样本对抗性的同时离决策边界最近的同义词为最佳同义词

Liu Z等人提出HyGloadAttack，使用全局替换进行初始化，利用词嵌入空间中的扰动矩阵在全局初始化后找到附近的对抗性示例，并选择在保持对抗性属性的同时最大化相似性的同义词

2022 2022 2023 2024

2021 2022 2023 2024

Maheshwary提出HLGA，提出初始化生成文本对抗样本后再优化的思想，先通过随机替换以初始化生成对抗样本，再通过减小搜索空间及使用遗传算法优化同义词以优化对抗样本

Ye M提出LeapAttack，在词嵌入空间中使用蒙特卡罗估计梯度方向，选择与梯度方向余弦相似度最大的词为最佳同义词

Liu H提出SSPAttack，将样本与原始样本的语义相似度作为基准，选择保持样本对抗性的同时使得语义相似度最高的同义词为最佳同义词

Hu X等人提出FastTextDodger，引入义原优化同义词集，使用分层注意力网络在初始化阶段对单词进行排序，使用对抗解扩展的方法快速优化对抗样本

文本分类硬标签黑盒模型的对抗样本生成方法

- 贪心策略
- 元启发算法

【SIGKDD】

PAT: Geometry-Aware Hard-Label Black-Box Adversarial Attacks on Text

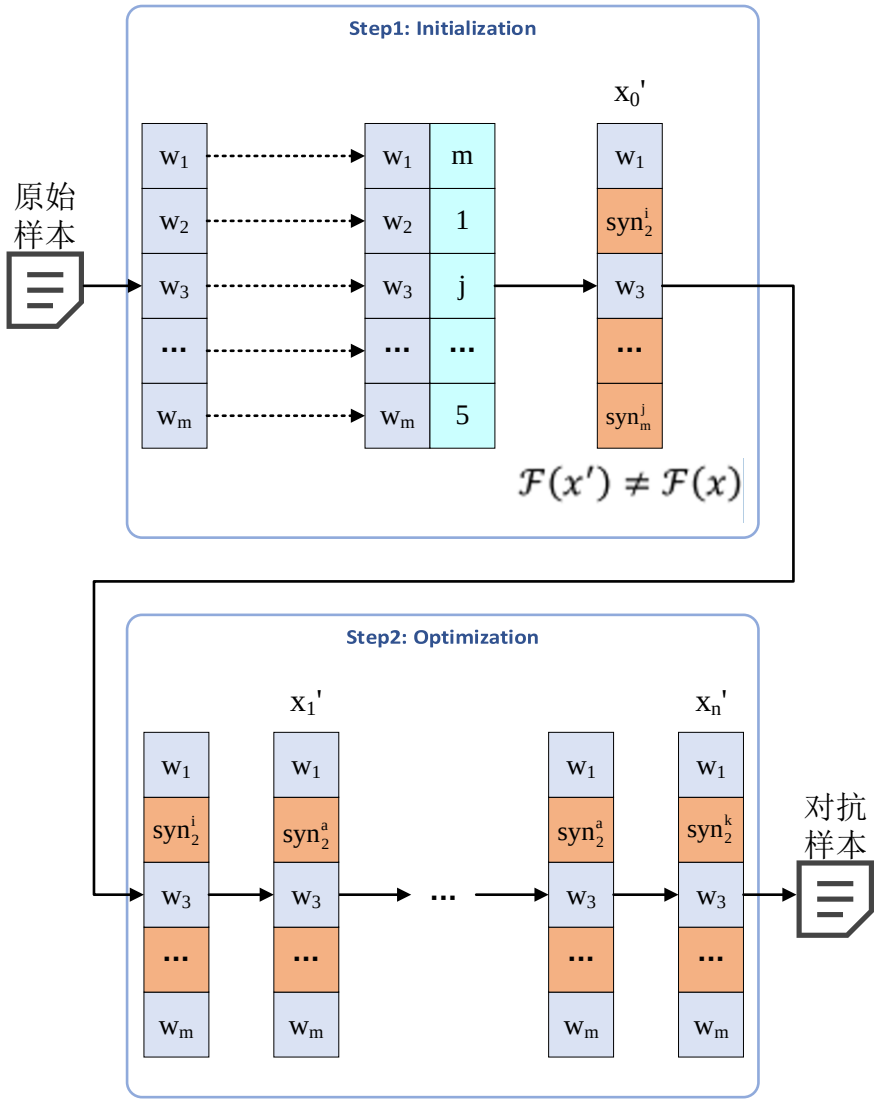
PAT TIPO

T	目标	针对文本分类硬标签黑盒模型，生成对抗样本
I	输入	原始样本、同义词集
P	处理	1. 初始化：随机同义词替换生成初始对抗样本 2. 原词替换：将对模型预测影响微弱的同义词替换回原词 3. 同义词替换：使用原型估计方法估计决策边界，选出最佳同义词进行替换 4. 迭代：对优化后的对抗样本进行原词替换和同义词替换，多轮迭代
O	输出	对抗样本

P	问题	基于梯度的方法将连续的扰动映射到离散的单词时存在近似误差
C	条件	目标模型硬标签黑盒设置
D	难点	1. 如何在硬标签黑盒设置下选择最佳同义词 2. 如何避免遍历同义词集，减少目标模型查询次数
L	水平	SIGKDD 2023 CCF A

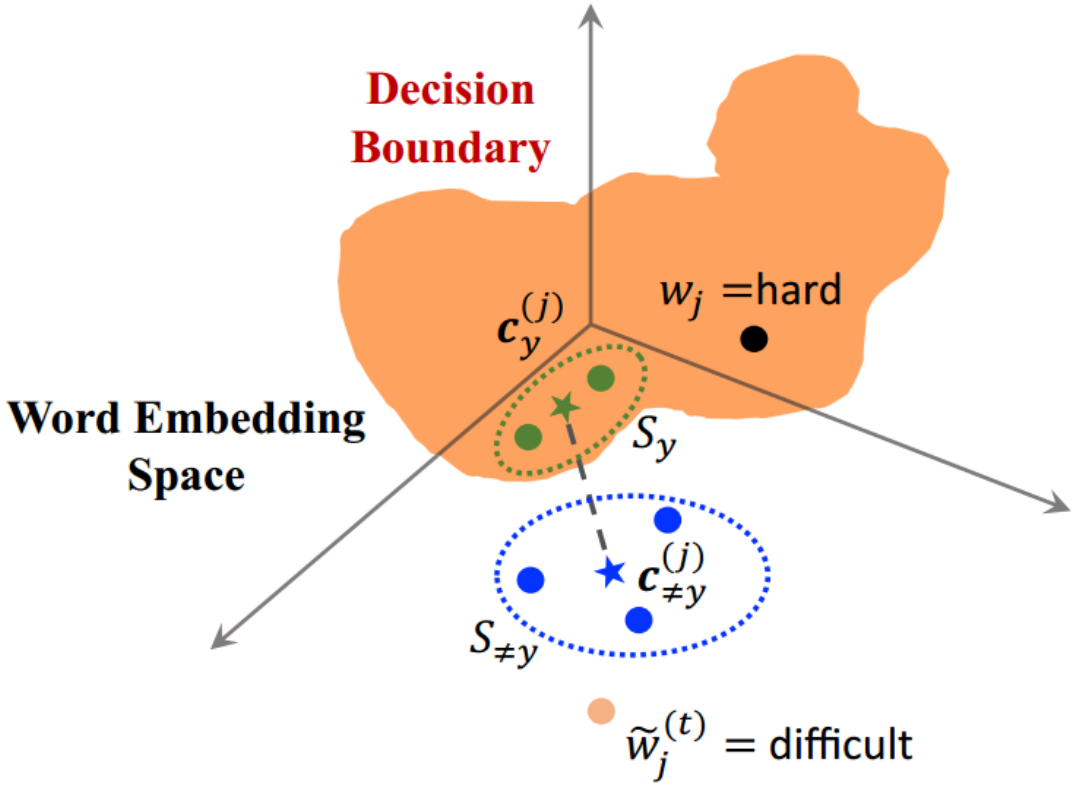
PAT 算法原理

- 算法原理图
  - 初始化
    - 随机替换位置，随机同义词
  - 优化
    - 原词替换
      - 选出影响微弱的同义词
      - 决定替换次序
      - 原词替换
    - 同义词替换
      - 决定替换次序
      - 估计决策边界（原型估计）
      - 选出最佳同义词



PAT 原型估计

- 原型估计
  - 同义词抽样
  - 抽样单词对抗性判别
    - 对抗性同义词
    - 非对抗性同义词
  - 原型估计
    - 对抗性原型
    - 非对抗性原型
  - 决策边界估计
    - 两原型连线的中垂面



(c) Prototype Estimation

PAT 最佳同义词选择

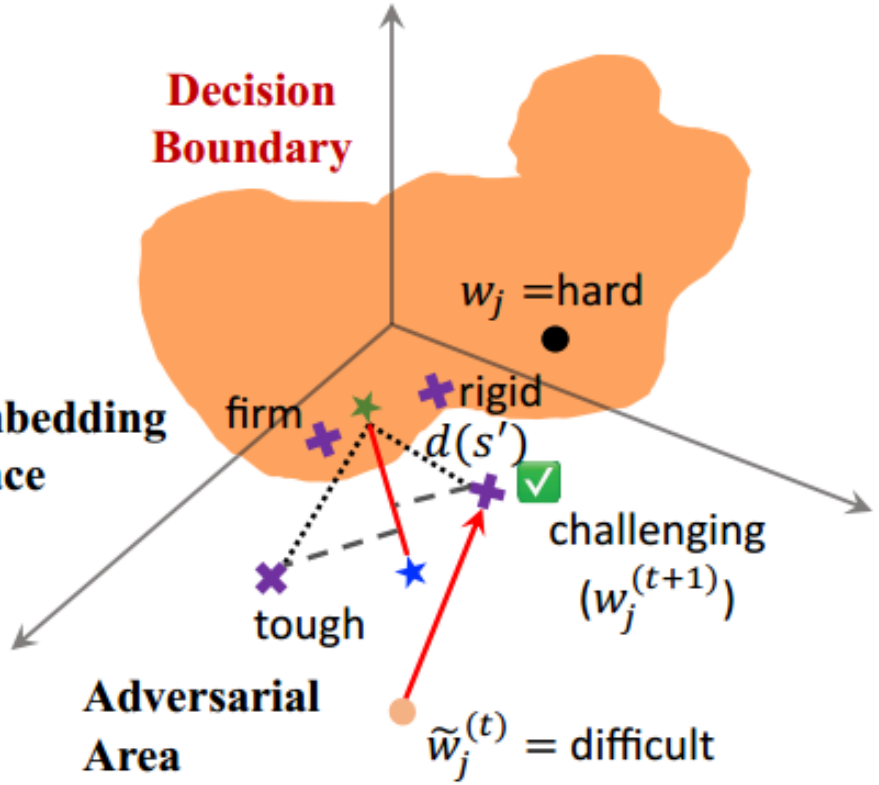
- 最佳同义词
  - 同义词对抗性判断
    - 对抗性同义词
    - 非对抗性同义词

$$||\mathcal{H}(s) - \mathbf{c}_{\neq y}^{(j)}|| < ||\mathcal{H}(s) - \mathbf{c}_{y}^{(j)}||,$$

  - 边界距离计算

$$d(s') = ||\mathcal{H}(s') - \mathbf{c}_{y}^{(j)}|| \cdot |\cos(\mathcal{H}(s') - c_{y}^{(j)}, \mathbf{c}_{\neq y}^{(j)} - \mathbf{c}_{y}^{(j)})|,$$

  - 同义词排序
  - 最佳同义词选择



(d) Replacement Selection

实验设计　数据资源

- 数据集
  - MR：二元情感分类的电影评论数据集，短文本
  - AG：世界、体育、商业、科学四类新闻分类数据集
  - Yahoo：包含十类的问答主题分类数据集
  - Yelp：二元情感分类的商户点评数据集
  - IMDB：二元情感分类的电影评论数据集，长文本
- 对比方法
  - HLGA(2021 CCF A)
  - TextHoaxer(2022 CCF A)
  - LeapAttack(2022 CCF A)
- 评价指标：
  - Pert：单词替换率，替换单词占全部单词的百分比
  - Sim：语义相似度，使用USE模型计算
  - Acc：模型的分类准确率
  - Number of Query：查询次数

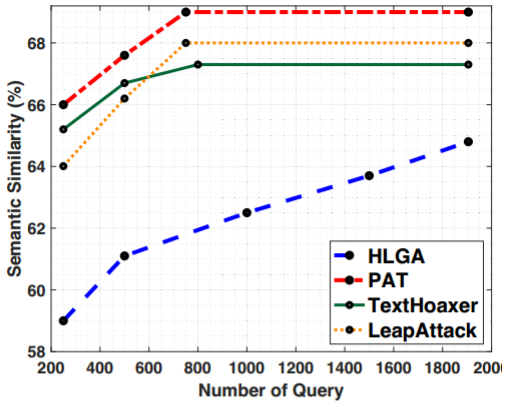
实验设计　对比实验

- 收敛性能比较
  - PAT和对比算法具备相同的攻击成功率
  - PAT在所有数据集上语义相似度都优于对比算法，效果提升较大
  - PAT在所有数据集上单词替换率都优于对比算法，效果提升较大

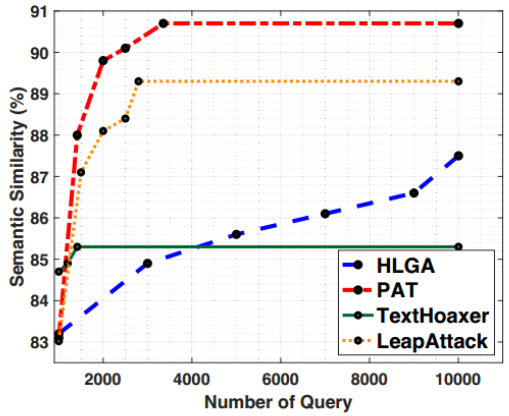
Dataset	Method	BERT			WordCNN			WordLSTM		
		Acc (%)	Sim (%)	Pert (%)	Acc (%)	Sim (%)	Pert (%)	Acc (%)	Sim(%)	Pert (%)
MR	Random	1.0 (← 85.0)	15.2	41.264	0.7 (←76.5)	18.0	39.127	0.7 (← 78.0)	17.2	39.475
	TextHoaxer		67.3	11.812		68.4	12.076		67.7	12.320
	HLGA		64.8	13.360		66.6	12.976		65.4	13.574
	LeapAttack		68.0	10.420		68.6	10.742		68.1	10.959
	PAT		**69.0**	**10.364**		**69.9**	**10.635**		**69.0**	**10.903**
AG	Random	2.8 (← 93.0)	23.0	48.517	1.4 (← 90.4)	30.3	43.494	5.7 (← 90.2)	24.4	47.546
	TextHoaxer		63.8	15.721		74.2	12.493		64.6	16.122
	HLGA		68.9	12.750		78.2	10.250		70.9	12.864
	LeapAttack		69.9	10.852		78.9	8.987		71.8	11.208
	PAT		**72.4**	**9.731**		**81.6**	**7.924**		**74.9**	**9.959**
Yahoo	Random	0.5 (← 79.1)	29.7	30.239	0.8 (← 71.1)	27.6	32.302	1.9 (← 73.7)	24.6	33.291
	TextHoaxer		70.9	6.726		75.0	7.616		67.2	8.396
	HLGA		71.6	5.865		76.2	6.492		68.4	6.999
	LeapAttack		72.0	4.851		78.2	5.629		69.6	5.952
	PAT		**74.3**	**4.382**		**80.1**	**5.110**		**71.1**	**5.586**
Yelp	Random	5.2 (← 96.5)	17.0	39.344	0.6 (← 92.9)	16.6	39.286	3.2 (← 94.8)	17.6	38.966
	TextHoaxer		74.6	9.271		81.3	8.543		80.8	7.942
	HLGA		78.4	7.081		83.8	6.675		83.0	6.166
	LeapAttack		80.5	5.985		86.2	5.845		84.9	5.484
	PAT		**82.9**	**5.246**		**88.0**	**5.329**		**86.3**	**4.974**
IMDB	Random	0.1 (← 90.3)	34.0	30.714	0.0 (← 87.8)	34.2	13.164	0.3 (← 89.3)	33.5	32.020
	TextHoaxer		85.3	4.860		90.2	4.268		89.1	4.098
	HLGA		87.5	3.307		91.3	3.001		90.1	2.916
	LeapAttack		89.3	2.894		92.9	2.881		91.3	2.681
	PAT		**90.7**	**2.299**		**93.7**	**2.613**		**92.3**	**2.526**

实验设计　对比实验

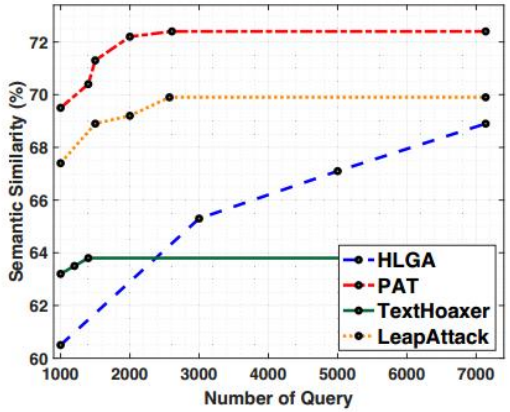
- 攻击效率比较
  - PAT在语义相似度上，对短文本、长文本和多标签分类都有更高的攻击效率
  - PAT在单词替换率上，对短文本、长文本和多标签分类都有更高的攻击效率



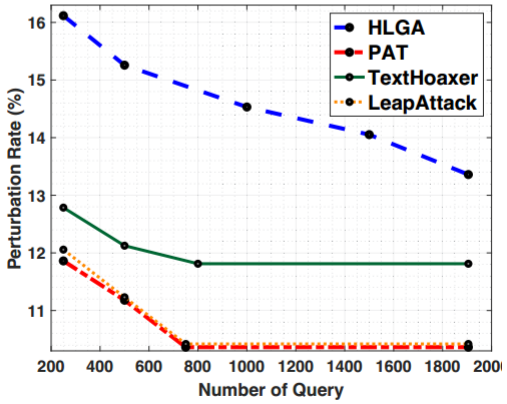
(a) MR (Sim)



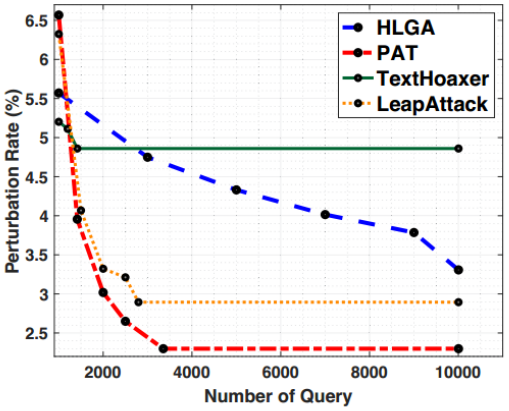
(e) IMDB (Sim)



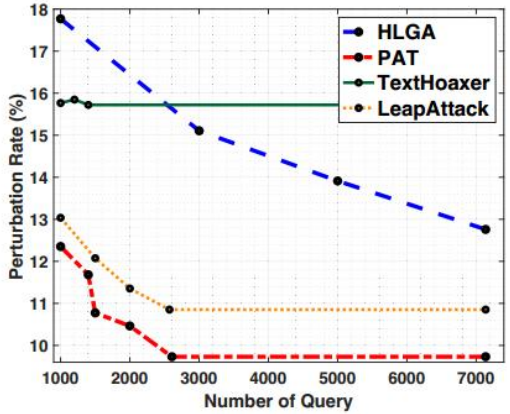
(b) AG (Sim)



(f) MR (Pert)



(j) IMDB (Pert)



(g) AG (Pert)

实验设计　消融实验

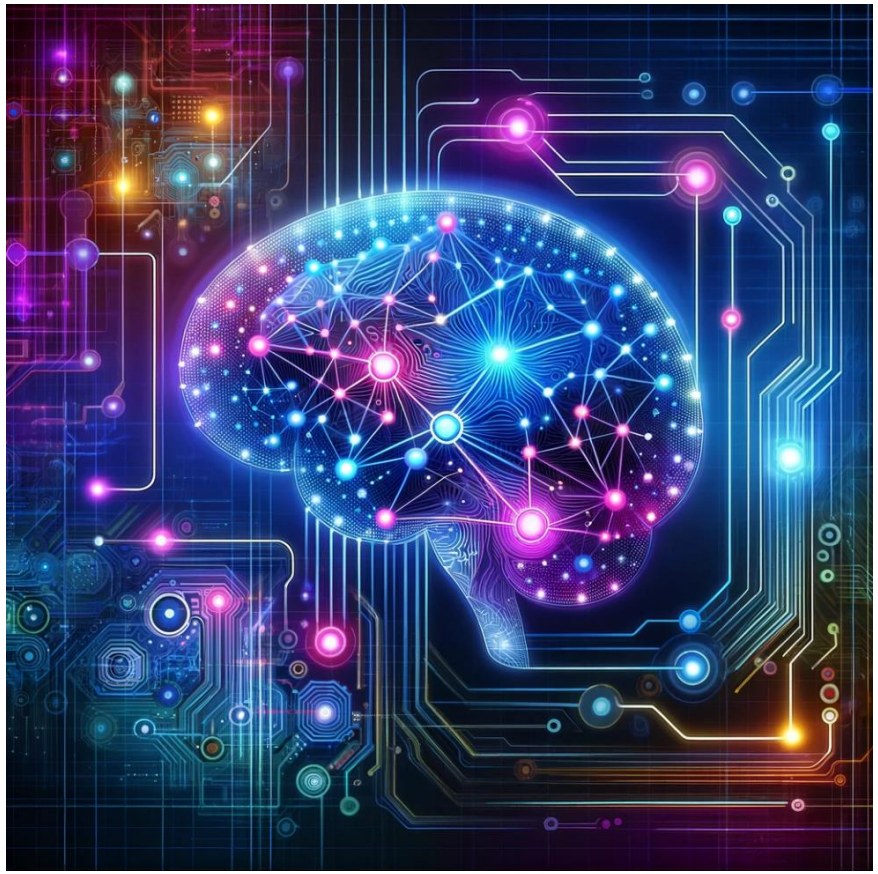
- 消融实验
  - w/o Replacing：不包含原词替换步骤
  - w/o Selecting (Random)：随机选择最佳同义词
  - w/o Selecting (Most Similar)：在词嵌入空间中选择与原始同义词最相似的同义词
- 实验结论
  - 原词替换步骤有助于降低单词替换率
  - 原型估计方法选择最佳同义词是方法的关键

Table 2: Ablation study results on module design.

Dataset	Variant	Sim (%)	Pert (%)
AG	w/o Replacing	81.0	11.647
	w/o Selecting (Random)	70.9	13.077
	w/o Selecting (Most Similar)	68.8	14.541
	PAT	**81.6**	**7.924**

算法总结 PAT

- 算法流程
  - 随机同义词替换生成初始对抗样本
  - 迭代优化
    - 原词替换
    - 同义词替换
- 算法优势
  - 针对文本分类硬标签黑盒模型，适用面广
  - 生成的对抗样本单词替换率低，语义相似度高
  - 攻击效率高
- 算法不足
  - 随机同义词替换生成初始对抗样本的效率较低
  - 忽略了语义相似度在同义词选择中的指引作用



【TIFS】
FastTextDodger: Decision-Based Adversarial Attack Against Black-Box NLP Models With Extremely High Efficiency

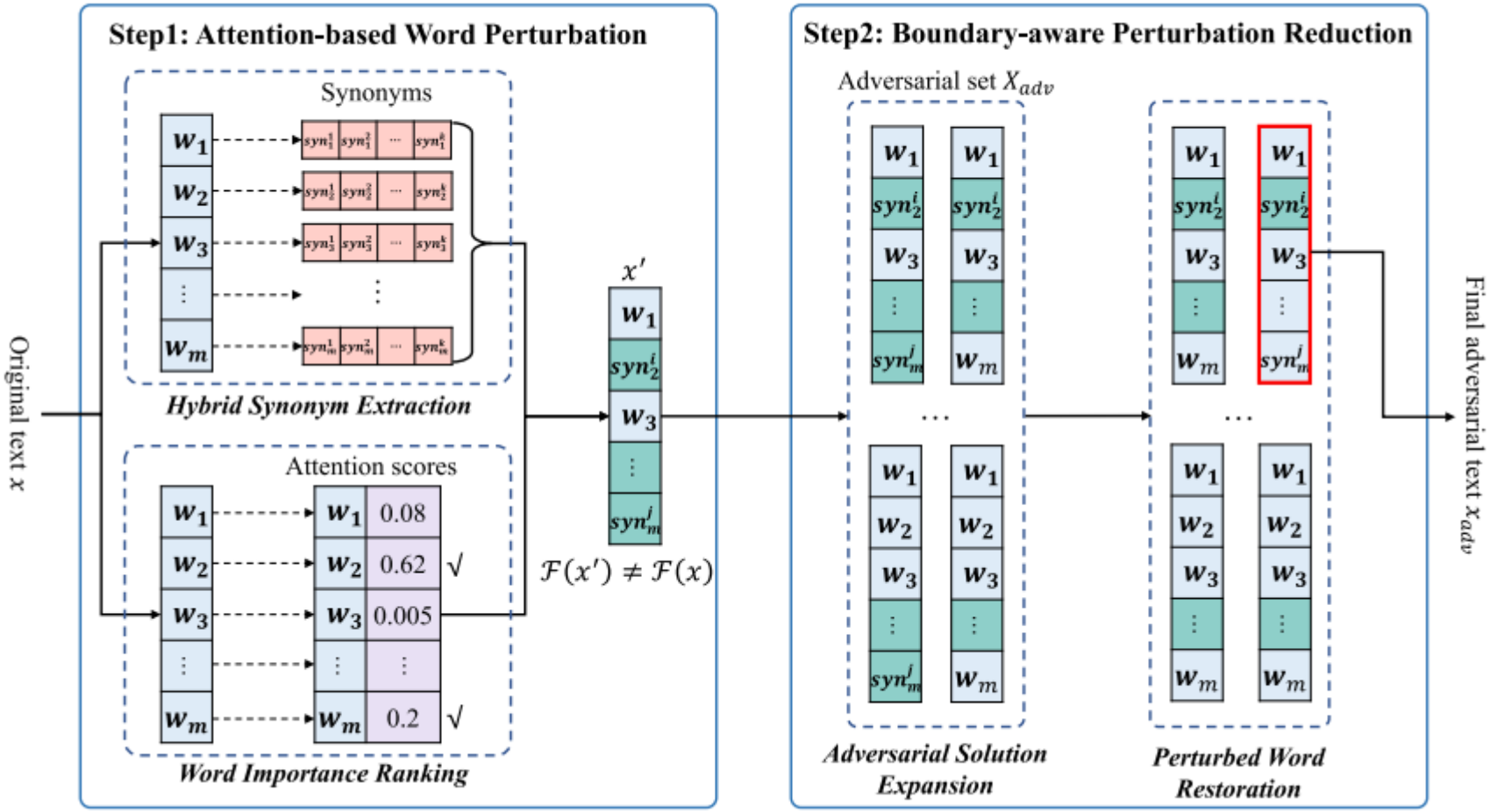
FastTextDodger TIPO

T	目标	针对文本分类硬标签黑盒模型，实现查询高效的对抗性样本攻击
I	输入	原始样本、同义词集
P	处理	1. 基于注意力的初始化 2. 对抗解扩展 3. 干扰词恢复
O	输出	对抗样本

P	问题	现有的方法要么需要大量查询，要么攻击成功率低，无法满足实际需求
C	条件	目标模型硬标签黑盒设置
D	难点	1. 如何在少查询下保持攻击成功率和单词替换率
L	水平	TIFS 2024 CCF A

FastTextDodger 算法原理图

- 算法原理图
  - 初始化
    - 混合同义词提取
      - 词嵌入空间
      - 义原
    - 单词重要性排序
      - 分层注意力网络
  - 优化
    - 对抗解扩展
    - 干扰词恢复



初始化 优化

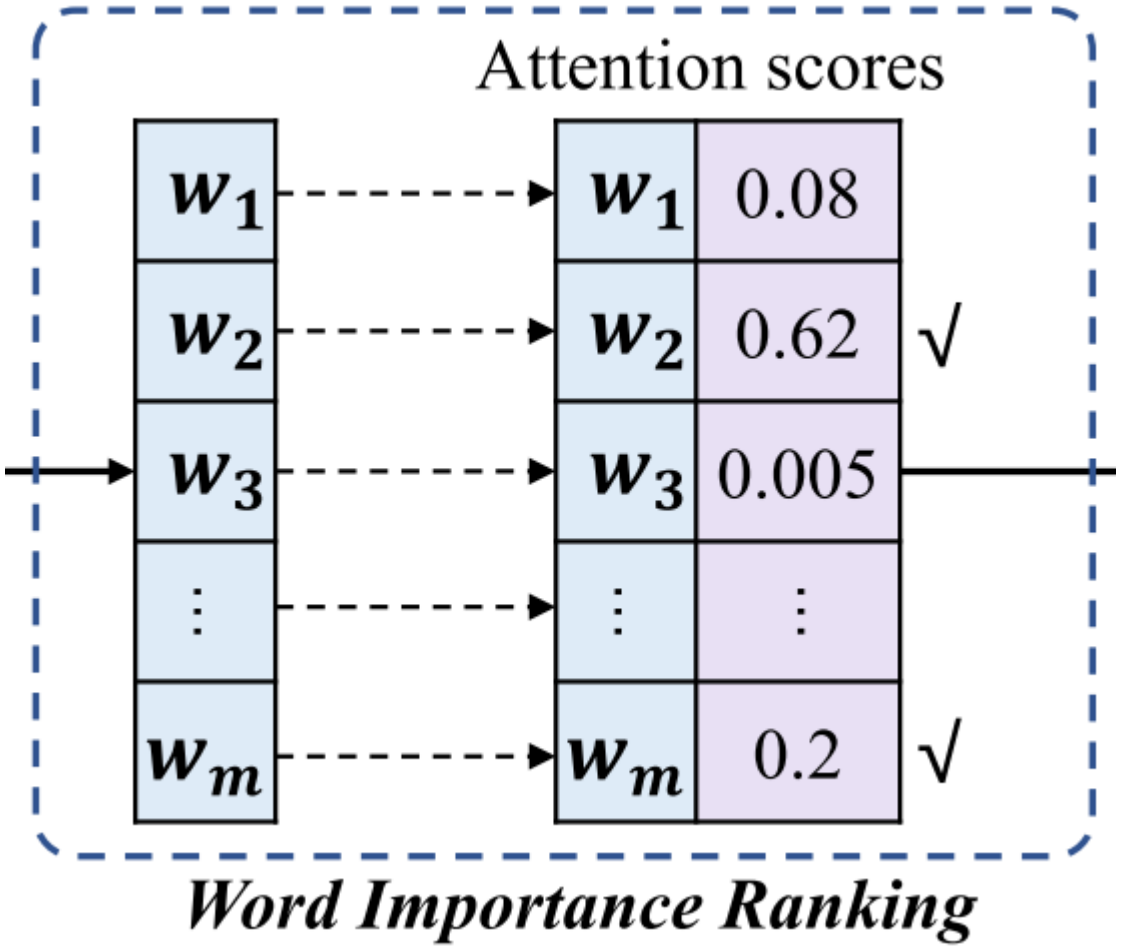
- 混合同义词提取
  - 义原
    - 即原子语义，是语言学意义上的最小的、不可再分的语义单位
    - 基于义原的同义词具备更高的质量
    - 部分单词无法通过义原找到同义词
  - 词嵌入空间

样本	We		love	science
义原	huaman	1stPerson	FondOf	knowledge
候选词	men	I	**like**	technology
	people			logic
对抗样本	People		like	science

$$\mathrm{Syn}(w_i) = \begin{cases} \mathrm{se}_i, & \mathrm{se}_i \neq \emptyset, \\ \mathrm{em}_i, & \text{others.} \end{cases}$$

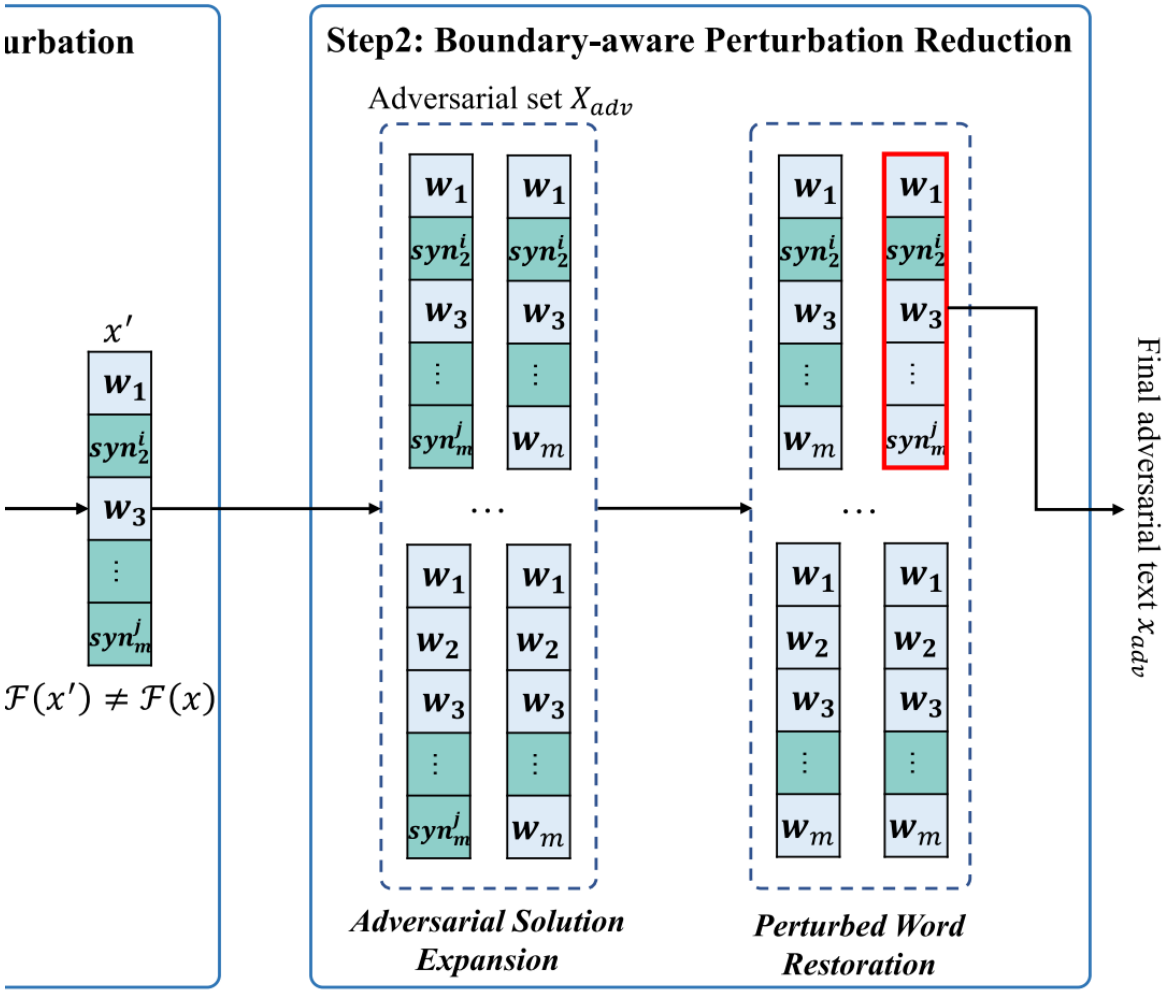
FastTextDodger 单词重要性排序

- 单词重要性排序
  - 句子过滤
    - 过滤与样本标签不同的句子
  - 单词过滤
    - 过滤掉其词性不是形容词、副词、动词或名词的单词
    - 两重过滤获得单词集W
  - 计算注意力分数
    - 使用分层注意力网络计算W中单词的注意力分数，作为重要性分数
  - 单词重要性排序



FastTextDodger 优化

- 优化
  - 对抗解扩展
    - 超参数
      - K：对抗样本集的大小
      - σ：抽取单词的比例
    - 从W中按照比例σ抽取单词
    - 改变抽取的单词
      - 将被替换的单词替换回原词
      - 将未被替换的单词替换为同义词
    - 重复上述操作直到获取一个大小为K的对抗样本集
  - 干扰词恢复



实验设计　数据资源

- 数据集
  - MR：二元情感分类的电影评论数据集，短文本
  - AG：世界、体育、商业、科学四类新闻分类数据集
  - Yahoo：包含十类的问答主题分类数据集
  - Yelp：二元情感分类的商户点评数据集
  - IMDB：二元情感分类的电影评论数据集，长文本
- 对比方法
  - TextFooler(2020 CCF A)
  - PWWS(2019 CCF A)
  - Textbugger(2019 CCF A)
  - PNLA(2021 CCF A)(HLGA)
  - TextDecepter(2020 未发表)
  - LeapAttack(2022 CCF A)
- 评价指标：
  - ASR：攻击成功率
  - Pert：单词替换率，替换单词占全部单词的百分比
  - NoQ：查询次数
  - Sim：语义相似度，使用USE模型计算
  - GmErr：语法错误率，使用LanguageTool计算

实验设计　对比实验

- 对比实验
  - FastTextDodger在所有数据集上查询次数都远少于其余方法
  - FastTextDodger在所有数据集上单词替换率都差于PNLA和LeapAttack
  - FastTextDodger的攻击成功率在IMDB、MR上与PNLA和LeapAttack相当，在Yelp上优于PNLA和LeapAttack，在AG和Yahoo上较差

Dataset	Attack	Bert			WordCNN			WordLSTM		
		NoQ ↓	Pert ↓	ASR ↑	NoQ ↓	Pert ↓	ASR ↑	NoQ ↓	Pert ↓	ASR ↑
IMDB	TextFooler [7]	737.7	6.2	99.6	587.0	4.2	100.0	607.7	4.4	100.0
	PWWS [8]	1460.6	3.4	99.7	6569.0	1.3	100.0	6481.3	1.3	100.0
	TextBugger [9]	473.4	27.8	92.7	316.2	49.4	99.7	461.2	46.3	99.8
	PNLA [10]	10025.0	3.3	99.9	9403.9	3.0	100.0	8717.7	2.9	99.7
	TextDecepter [11]	1531.2	1.9	64.9	1071.0	3.0	80.0	1332.7	2.9	70.1
	LeapAttack [12]	3696.9	2.9	99.9	3523.6	2.9	99.7	3319.1	2.7	99.7
	Ours	251.1	3.4	99.7	256.9	3.0	99.5	250.0	2.9	99.6
MR	TextFooler [7]	451.9	26.3	76.5	241.8	16.2	98.6	364.6	22.6	88.5
	PWWS [8]	149.5	13.5	91.4	567.0	10.6	99.4	562.1	10.2	99.6
	TextBugger [9]	55.9	14.9	62.7	48.2	27.0	78.4	46.8	25.4	80.0
	PNLA [10]	1905.2	13.4	98.8	1824.6	13.0	99.1	1864.1	13.6	99.1
	TextDecepter [11]	117.2	12.2	69.2	103.2	12.2	76.5	102.2	12.6	76.7
	LeapAttack [12]	747.1	10.4	99.1	795.0	10.6	99.2	775.7	10.9	99.4
	Ours	83.7	14.6	98.0	60.8	14.7	99.3	100.7	14.7	99.0
Yelp	TextFooler [7]	161.5	16.5	86.9	124.7	14.5	96.3	128.3	15.2	96.2
	PWWS [8]	1028.9	7.0	94.0	3735.8	3.9	99.7	3749.3	3.5	100.0
	TextBugger [9]	333.5	23.4	83.8	222.7	51.2	95.6	218.9	49.1	96.5
	PNLA [10]	10344.3	7.1	94.6	9852.7	6.7	99.4	8936.1	6.2	96.6
	TextDecepter [11]	1248.1	4.8	41.6	1093.4	5.5	49.7	1092.6	5.4	53.9
	LeapAttack [12]	3925.3	6.1	95.3	3832.8	6.0	99.7	3376.2	5.4	96.1
	Ours	282.2	8.4	96.7	239.8	7.4	99.7	227.8	6.9	98.8
AG	TextFooler [7]	636.1	13.6	91.4	465.1	11.4	96.6	704.4	15.0	91.7
	PWWS [8]	369.7	16.9	53.5	315.0	14.9	87.8	1502.9	11.4	99.1
	TextBugger [9]	187.5	33.4	51.8	114.9	57.4	86.4	133.2	56.5	75.1
	PNLA [10]	7136.6	12.7	97.0	5319.6	10.3	98.5	6800.3	12.9	93.7
	TextDecepter [11]	478.3	10.3	23.5	455.3	7.7	31.1	482.3	9.7	26.8
	LeapAttack [12]	2768.2	10.8	96.8	2247.7	8.8	98.5	2700.5	11.0	93.1
	Ours	334.0	15.3	94.4	206.7	11.9	96.1	331.0	15.5	92.9
Yahoo	TextFooler [7]	796.7	13.6	95.0	465.2	8.2	98.8	546.4	9.5	97.7
	PWWS [8]	2911.6	3.8	100.0	3072.8	4.5	100.0	3074.3	3.8	100.0
	TextBugger [9]	234.3	24.2	91.3	215.9	46.7	93.2	241.6	42.6	93.4
	PNLA [10]	6401.2	5.8	99.4	8207.6	6.5	98.9	8350.7	7.0	97.4
	TextDecepter [11]	745.6	5.8	64.6	814.1	5.2	72.6	767.2	6.0	70.6
	LeapAttack [12]	2168.2	4.7	99.4	2955.2	5.7	98.9	2890.3	5.9	98.0
	Ours	197.4	6.5	96.8	238.3	6.8	92.7	303.9	7.5	95.4

实验设计　消融实验

- 消融实验
  - 注意力
    - 注意力分数对攻击成功率和单词替换率影响微小
    - 注意力分数小幅降低了查询次数，大约降低10%
  - 优化
    - 优化步骤显著降低了单词替换率和语法错误率，显著提升了语义相似度

Dataset	Attack	Bert			WordCNN		
		NoQ ↓	Pert ↓	ASR ↑	NoQ ↓	Pert ↓	ASR ↑
IMDB	w/ attention	201.1	3.8	99.9	215.2	3.1	99.8
	w/o attention	221.6	3.9	99.8	249.7	3.2	99.7
MR	w/ attention	77.0	14.6	98.1	75.6	14.4	99.3
	w/o attention	83.9	14.4	98.5	75.8	14.0	99.4
Yelp	w/ attention	224.2	8.7	96.9	198.6	7.6	99.6
	w/o attention	269.1	8.8	97.2	213.5	7.6	99.4
AG	w/ attention	259.2	15.2	95.1	208.1	12.1	96.9
	w/o attention	228.8	15.8	93.8	206.8	12.5	96.6
Yahoo	w/ attention	228.3	6.5	98.2	238.0	7.2	94.8
	w/o attention	239.0	7.1	97.7	262.8	7.6	94.9

Ablation Study	IMDB			Yelp		
	Pert	Sim	GmErr	Pert	Sim	GmErr
w/o BPR	15.2	59.2	3.6	24.5	40.5	3.3
w/ BPR	3.0	85.8	1.2	7.5	74.8	1.1

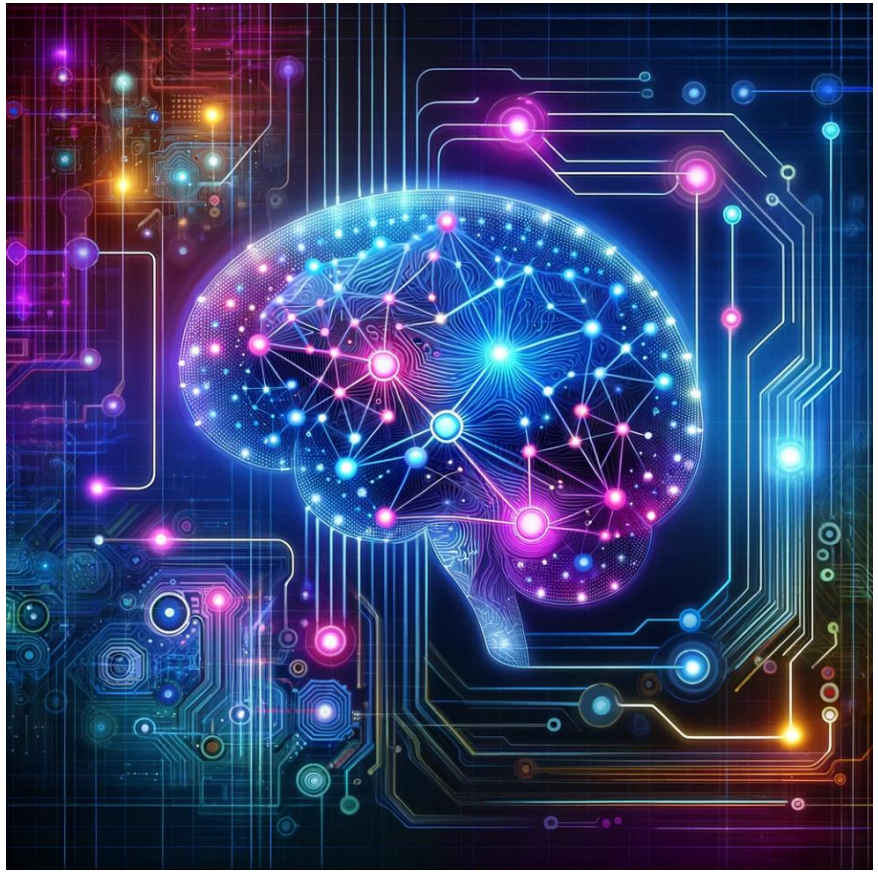
实验设计　补充实验

- 补充实验
  - 优化
    - 在短文本（MR）上，FastTextDodger在攻击效率上相比PAT不具备优势
    - 在长文本（IMDB）上，FastTextDodger在攻击效率上相比PAT具备明显的优势，只需要33%的查询次数便可以达到相近的单词替换率

Dataset	Attack	Bert	
		NoQ	Pert
MR	PAT	83.3	14.53
	FastTextDodger	83.7	14.6
IMDB	PAT	457.1	5.784
	PAT	753.9	3.36
	FastTextDodger	251.1	3.4

算法总结 FastTextDodger

- 算法流程
  - 初始化
    - 混合同义词提取
    - 单词重要性排序
  - 优化
    - 对抗解扩展
    - 干扰词恢复
- 算法优势
  - 在少查询下生成的对抗样本
- 算法不足
  - 不同的数据集需要不同的超参数K和σ
  - 评价指标欠缺，只对比了查询次数和单词替换率，没有对比语义相似度



特点总结与未来展望

特点总结与未来展望

- 特点总结
  - PAT
    - 优化方向：提升对抗样本质量，减少查询次数
    - 方法特点：使用原型估计方法估计决策边界，使用决策边界指导同义词选择
  - FastTextDodger
    - 优化方向：减少查询次数
    - 方法特点：使用义原优化同义词集，在初始化阶段使用分层注意力网络计算单词重要性，使用对抗解扩展和干扰词恢复以优化对抗样本
- 未来发展
  - 提升同义词集的质量，解决一词多义和无关词的问题
  - 设计更有效的评价指标，解决生成的对抗样本有效性不足的问题
  - 提升方法的攻击效率，提升对抗样本质量

参考文献

[1] Ye M, Chen J, Miao C, et al. PAT: Geometry-Aware Hard-Label Black-Box Adversarial Attacks on Text. Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining [C]. New York: ACM, 2023: 3093-3104.

[2] Hu X, Liu G, Zheng B, et al. FastTextDodger: Decision-Based Adversarial Attack Against Black-Box NLP Models With Extremely High Efficiency[J]. IEEE Transactions on Information Forensics and Security, 2024.

[3] Chiang C H, Lee H. Are Synonym Substitution Attacks Really Synonym Substitution Attacks? [EB/OL]. (2023-5-8)[2024-11-17]. https://arxiv.org/abs/2210.02844

知人者智，自知者明。胜人者有力，自胜者强。知足者富。强行者有志。不失其所者久。死而不亡者，寿。

谢谢！