举例规则:利用逻辑规则来解释仇恨言论检测

克里斯托弗·克拉克:马修·霍尔;高拉夫·米塔尔;叶宇;
桑德拉·萨吉夫;杰森·马斯陈梅;

:密歇根大学,密歇根州安娜堡
;微软,华盛顿州雷德蒙德
{csclarke,教授}@umich.edu
{mathall、gaurav.mittal、yu.ye、ssajeev、mei.chen}@microsoft.com

摘要

内容审核的经典方法通常采用基于规则的启发式方法
标志内容。虽然规则很容易定制并且人类可以直观地解释,但它们
本质上是脆弱的,缺乏灵活性或
调节大量数据所需的稳健性
今天在网上发现的不良内容。深度学习的最新进展已经证明了使用高效

深度神经模型可以克服这些挑战。然而,尽管性能有所提高,这些数据驱动的模型缺乏透明度和可解释性,常常导致日常用户的不信任和缺乏采用

通过许多平台。在本文中,我们提出
Rule By Example (RBE):一种新颖的基于示例的对比学习方法,用于从文本任务的逻辑规则中学习

内容审核。 RBE 能够提供基于规则的预测,允许

更可解释和可定制的预测
与典型的基于深度学习的方法相比。我们证明我们的方法
能够仅使用少量数据示例来学习丰富的规则嵌入表示。

3种流行仇恨言论的实验结果
分类数据集表明 RBE 能够
超越最先进的深度学习分类器以及在监督和无监督环境中使用规则,同时通过规则基础提供可解释的模型预测。

1 简介

内容审核是在线社交平台安全面临的重大挑战

例如 Facebook、Twitter、YouTube、Twitch 等。
（Vaidya 等人， 2021）。大型科技公司越来越多地分配宝贵的资源
致力于开发自动化系统

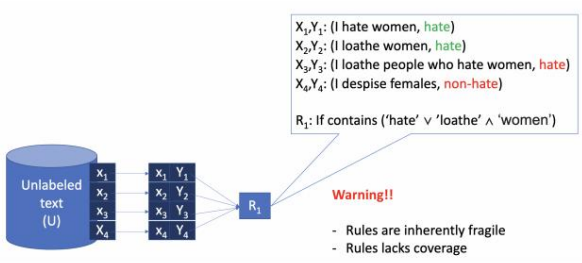

图 1:规则的泛化问题。逻辑规则,
虽然很容易解释,但本质上很容易受到细微差别的影响
自然语言。
检测和调节有害内容

除了聘请和培训专家人类主持人来应对日益增长的消极威胁
和在线毒性（Wagner 和 Bloomberg， 2021；
刘等人， 2022）。
尽管深度学习方法很流行,但产品中使用的许多实用解决方案
今天由基于规则的技术组成
基于专业策划的信号,例如阻止列表、密钥
短语和正则表达式（Gillespie， 2018；
张， 2019;达达等人， 2019）。检索方式
因其透明性、易于操作性而被广泛使用
定制和可解释性。然而,他们
缺点是难以维护
和规模,除了本质上脆弱之外
并且嘈杂（Zhang， 2019； Davidson 等人， 2017； Lee，
2022 年;赖等人， 2022）。图1显示了一个示例
其中逻辑规则虽然本质上是可以解释的,
面临对环境不灵活的问题
自然语言中的使用。虽然给定的规则可能
过于具体,无法捕捉在线内容中常见的不同用法变化,

规则也可能过于宽泛并且错误地阻止
词汇相似的内容。
与基于规则的挑战相比
方法,数据驱动的深度学习方法
在广泛的领域表现出了巨大的希望
内容审核任务和模式（Malik
等， 2022； Shido 等人， 2022;赖等人， 2022）。
由大量数据和深度神经网络推动
网络,这些复杂的模型能够

*这项工作是作为克里斯托弗的实习项目完成的微软。

计算语言学协会第 61 届年会论文集
第一卷:长论文,第 364-376 页
2023 年 7 月 9-14 日

学习更丰富的表示,更好地泛化到未见过的数据。令人印象深刻的表演
这些模型导致了行业对内容审核即服务的大量投资尝试。
Google [3] OpenAI 和 Microsoft 等多家科技公司提供了帮助内[1],[2]容审核的服务。然而,尽管他们投入 使用这些模型
巨资,他们仍面临

由于客户无法理解这些复杂模型的原理,采用面临挑战

他们做出决定的原因(Tarasov, 2021; Haimson等人, 2021; Juneja 等人, 2020)。此外,
随着人们对在线内容的日益关注
消费者之间的节制和干扰、可解释性和透明度是最重要的

要求(Kemp 和 Ekins, 2021; Mukherjee 等人, 2022)。这提出了一个具有挑战性的开放性问题
我们如何利用复杂深度学习模型的稳健性和预测性能
同时允许透明度、可定制性,
以及基于规则的方法提供的可解释性。

先前的作品,例如Awasthi 等人。 (2020);搜索引擎优化等人。 (2021);普里赞特等人。 (2022)探索过
从控制等任务的规则中学习
神经网络学习,辅助人工标注,提高自我监督学习

低数据场景。阿瓦斯蒂等人。 (2020)提议
使用规则进行噪声监督的规则示例训练方法。在去噪方面表现出色的同时
网络中通过软规则过度概括的规则
蕴含损失,与其他机器学习方法类似,
该方法缺乏在推理时解释模型预测的能力。普里赞特等人。 (2022)

提出一个用于符号规则自动发现和集成的通用框架

到预先训练的模型中。然而,这些象征性的
规则源自低容量 ML 模型
在减少的特征空间上。虽然不太复杂
与大型深度神经网络相比,这些低容量
模型仍然不容易被人类解释。
因此,结合可解释性的任务
规则和深度学习的预测能力
模型仍然是一个悬而未决的问题。

为了解决这个问题,我们引入
Rule By Example (RBE):一种新颖的基于示例的规则

1https://perspectiveapi.com/
2https://openai.com/blog/
新的和改进的内容审核%
2D 模具/
3https://azure.microsoft.com/
zh-cn/产品/认知服务/
内容主持人/

用于从文本内容审核任务的逻辑规则中学习的对比学习方法。
RBE 由两个神经网络组成,规则
编码器和文本编码器,共同学习
仇恨内容的丰富嵌入表示
以及管理它们的逻辑规则。通过
使用对比学习,我们的框架使用
语义相似性目标,将可恶的示例与管理的规则示例集群配对
它。通过这种方法,RBE 能够提供
通过考虑什么来做出更可解释的预测
我们定义为规则基础。这意味着我们的
模型能够通过显示来证实其预测
相应的可解释的逻辑规则和
构成该规则的示例。

我们使用一套规则集在有监督和无监督的环境中评估 RBE。我们的
结果表明,每个副本只需一份
根据规则,RBE 能够在三个基准测试中超越最先进的仇恨文本分类器
两种设置中的内容审核数据集。在
总结一下,本文的贡献是:

·示例规则 (RBE):一种新颖的基于示例的对比学习方法

来自文本内容审核任务的逻辑规则。4

·我们演示了如何轻松集成 RBE,以将模型 F1 分数提高最多

三种流行的仇恨言论分类为 4%
数据集。

·对可定制性和可解释性特征的详细分析和见解

RBE 旨在解决新出现的仇恨内容和模型透明度问题。

2 示例规则框架

在本节中,我们概述了示例规则
框架,定义其操作术语,并描述其端到端架构。我们首先正式

描述中使用的两个主要操作术语
我们的框架:1)规则集- 规则集由以下部分组成
一系列可执行函数,当给定时
当且仅当输入满足规则中定义的所有条件时,文本作为输入“fire”。
图1
显示触发的简单规则的示例
如果给定文本包含关键字“讨厌”或

4https://github.com/ChrisIsKing/
举例规则

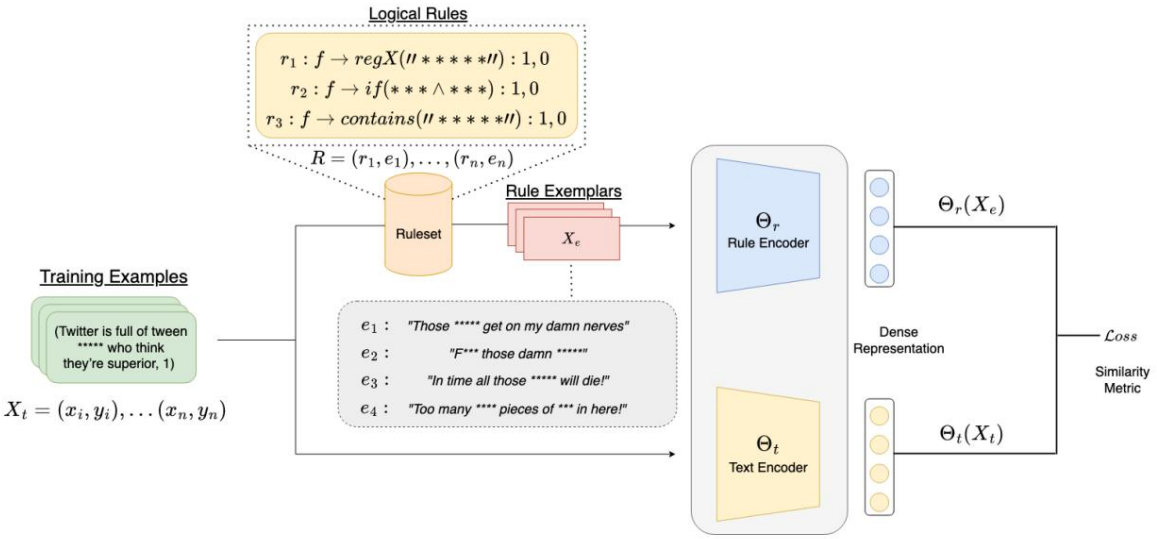

图 2:示例规则框架： RBE 由两个神经网络、一个规则编码器和一个文本编码器组成,它们共同学习仇恨内容的丰富嵌入表示以及管理它们的逻辑规则。通过对比学习,RBE 利用语义相似性目标,将可恶的示例与管理它的规则示例集群配对。

“厌恶”又包含“女人”。规则可以是作用于文本的任何可编程函数,例如正则表达式、阻止列表、关键字等。在这项工作的范围内,我们只考虑人类可以轻松解释的简单规则。因此,鉴于其黑盒性质,机器学习模型不能被视为规则。 2)范例- 范例是给定的文本示例,它很好地定义了规则所管辖的内容类型。例如,图1中的X1和X2可以被视为规则R1的示例,因为它们正确匹配 R1的条件。

考虑规则示例对R tpr1, e1q,pr2, e2q, ...,prn, enqu的规则集,其中ri de 表示定义的规则， ei表示ri正确触发的示例。对于包含标记示例X tpx1, y1q,px2, y2q, ...,pxm, ymqu 的给定语料库 X,每个规则ri可以用作黑盒函数Ri : x Ñ tyi , Hu来嘈杂地标记每个实例x使其分配标签y或根本不分配标签。一个实例可能被多个规则覆盖或根本没有规则。此外,覆盖集C表示X中触发规则 ri 的实例集。当规则被大量应用时出现的泛化问题有两个方面。当规则太宽泛时,封面集C就会很大,并且会错误地标记大量非仇恨内容。同样,当规则过于严格和脆弱时,覆盖集C就太小,词汇和语义上相似的令人讨厌的内容最终会被忽略。我们的目标是利用这些规则及其示例来促进可解释的模型学习。

算法 1有监督双编码器训练要求:规则编码器θr文本编码器θt输入:训练数据X px1, y1q...pxn, ynq,规则集R pr1, e1q, ...,prn, enq输出:更新参数θr, θt 1: 初始化θr和θt 2: 未收敛时do 3: 获取小批量Xb 4:对于Xb中的每个实例xi do 5: 获取样本ei “ doRulesetpR, xiq 6: 连接样本ei 7:结束为8: 9：

获取θrpEbq和θtpXbq
计算L” pYbD$\frac{1}{2}$ ^{2} ` p1 Ybqmaxpmargin D、 0q^{2} q
10： 更新θr和θt的参数11: end while

2.1 双编码器架构

如图2所示,双编码器架构通常用于密集检索系统和多模态应用（Clarke et al., 2022； Reimers and Gurevych, 2019； Xu et al., 2022）。我们的架构由规则编码器θr和文本编码器θt组成。

这是两个类似 Bert 的双向 Transformer 模型(Devlin et al., 2018) ,每个模型负责学习各自输入的嵌入表示。这种双编码器架构可以对示例进行预索引,从而可以在训练后运行时更快地进行推理。

编码管道给定输入文本xt ,首先从规则集R中提取适用规则集 我们及其各自的范例。然后,我们将每个提取的范例连接起来形成xe 。如果没有适用的规则,我们会从 en 中随机抽取样本

xt ,

轮胎规则设置为xe。使用形式xe rCLSs, e1然后使用规则编码器θr将xe编码为隐藏状态rCLSs , vrSEPs（其中e是第 n 个样本的第 k 个标记， rSEPs rCLS是特殊的标记。类似地，使用文本编码器θt ，我们对xt进行编码k

为了获得密集的表示，我们对隐藏状态应用均值池化操作并导出固定大小的句子嵌入。获得样本xe和文本xt的表示后，我们使用余弦函数来衡量它们之间的相似度：

$$simpxe, xtq \quad “ \} \frac{\theta rpxeq ¡ \theta tpxtq}{\theta rpxeq\} \}\theta tpxtq\}} \qquad (1)$$

我们采用对比损失（Hadsell et al., 2006)来学习我们的规则和文本编码器的嵌入表示。对比学习鼓励模型最大化相同标签示例之间的表示相似性，并最小化不同标签示例之间的表示相似性。这使得我们的编码规则集的嵌入表示能够匹配覆盖集 C 正确覆盖的文本的表示。同样，对于规则不正确覆盖的良好示例，我们的对比学习目标会增加这些表示之间的距离，从而限制规则集中某些规则的过度概括。

令Yt为文本Xt的正确标签， D为pxe的余弦距离， xtq ， m为边距，我们的对比学习损失函数定义如下：

$$“L” \frac{1}{2} pYtD \quad ^2 \quad 、 p1 \quad Ytqmaxpm \quad D, 0q \quad ^2_q \qquad (2)$$

算法1详细介绍了训练循环以及编码管道和对比损失步骤。

2.2 规则基础

通过采用基于嵌入的方法来学习表示，RBE 实现了我们定义的规则基础。规则基础使我们能够将模型预测追溯到可解释的规则集以及定义每个规则的示例。对于任何已被我们的双编码器标记为正的输入xt ，我们执行规则搜索以查找在该输入上触发的规则，并执行嵌入相似性搜索以查找最近的样本以及这些样本所属的规则。表2显示了一个示例。

3 实验设置

训练我们使用 AdamW 优化器训练所有模型，并在所有数据上使用 0.01 的权重衰减。我们采用早期停止，上限为 10 个 epoch，学习率为2e-5，批量大小为 8，并在前 10% 步长上使用余弦计划进行线性学习率预热。我们的模型使用 Azure 机器学习工作室使用 NVIDIA Tesla V100 32GB GPU 进行训练。我们预处理数据并在多次运行中使用不同的随机种子训练所有模型。我们的 RBE 实现基于 Huggingface Transformers (Wolf et al., 2020)和 Sentence Transformers (Reimers and Gurevych, 2019)。 RBE 使用两个基于 Bert 的网络，每个网络包含 1.1 亿个参数。

训练所有 3 个测试集的 RBE 的所有超参数变体以及Bert 基线需要大约 2,000 个 GPU 小时。

基线我们在监督和无监督环境下评估我们的训练算法。我们将按原样应用逻辑规则的基线与当前训练基于变压器的序列分类器的SOTA 方法进行比较（Mathew 等人， 2020）。

3.1 数据集

我们评估该任务的三个数据集的 RBE

仇恨言论分类。在每个数据集中，我们将问题框架为检测给定文本是可恨的还是非可恨的二元分类任务。我们使用手动管理的规则集来扩充每个数据集。下面提供了有关每个数据集和规则集的更多信息。

HateXplain （Mathew 等人， 2020)是一个用于可解释的仇恨语音检测的大型基准数据集，涵盖仇恨语音检测的多个方面。它由“20k 个样本组成，分为“可恨”、“攻击性”和“正常”3 个标签。此外，每个样本都附有相应的目标群体和可解释的理由。在我们的实验中，我们将仇恨和攻击性的输出类别合并为一个，分别产生“8k/1k/1k 仇恨样本和”6k/781/782 个非仇恨样本用于训练/验证/测试。此外，我们利用随附的规则集构建原理。

Jigsaw5是来自维基百科的大型数据集
由人类评估者标记为有毒行为的评论。定义的毒性类型是“有毒”，
“严重有毒”、“淫秽”、“威胁”、“侮辱”和
“身份仇恨”。每个评论可以有任何一个
或更多这样的标签。总共包含“230k
样品。在我们的实验中，我们定义了示例
“身份仇恨”阶层的人是可恨的，其余的人是
非仇恨导致数据集为 1405/100/712
可恨样本和“分别用于训练/验证/测试的158k/1k/63k
非可恨样本。

上下文滥用数据集 (CAD) (Vidgen
et al., 2021)是“25k Red dit 条目标记为六个概念上不同的条目”的带注释数据集
主要类别为“身份导向”、“人导向”、“隶属关系导向”、“反言论”、
“非仇恨诽谤”和“中立”。在我们的实验中，我们定义了“身份导向”的例子
类为可恨的，并将其余的例子视为
非仇恨导致数据集为 1353/513/428
用于训练/验证/测试的可恨样本和“12k/4k/4k 非可恨样本”。

3.2 规则集构建

仇恨+滥用列表我们利用规则集定位
身份仇恨，我们将其称为仇恨+滥用
列表。它由代表有害语言（例如诽谤或仇恨动词）的 n 元语法列表组成。
仇恨+滥用列表类似于网上常见的公开不良词汇列表。我们

将仇恨+滥用列表中的每个 n-gram 条目视为其
如果 n gram 存在于输入文本中，则提出正标签的自己的规则。总共，仇恨+虐待列表
由 2957 条不同的身份仇恨规则组成。

HateXplain 合理规则集使用 HateX 中包含的带标签注释器基本原理
plain 数据集，我们以编程方式为 HateXplain 生成规则集。为此，我们提取 1、2 和
来自注释者基本原理的 3-gram 子字符串和
按注释者确定的目标人口群体对它们进行聚类。然后我们取前 N 个 n 元语法
每个人口统计群体并自动创建
他们每个人的规则。这会导致类似的规则
在我们的仇恨+虐待清单中。使用默认值
25 个目标类别的聚类大小为 100
在 HateXplain 中定义，我们总共生成了 670 个
HateXplain 的独特规则。

5https://www.kaggle.com/competitions/拼图有毒评论分类%2挑战

上下文滥用合理规则集类似于
我们派生的 HateXplain 规则集，我们以编程方式生成上下文滥用的规则集
使用注释者标记的基本原理的数据集。按照之前概述的相同过程，这
结果总共有 2712 条不同的 CAD 规则。

样本选择对于每个数据集，我们通过配对每个规则来完成规则集构建
并附有示例。为了达成这个，
我们首先在数据集训练集上运行我们的规则集，
提取规则正确触发的实例。
对于每个正确触发的规则，我们随机
选择N个实例作为样本。此外，限制可能过于笼统的规则

我们强制执行的条件是不能有两条规则
映射到同一样本。除非另有说明，否则我们仅使用一份副本来报告结果
规则在我们的实验中。

3.3 无监督设置

除了在监督环境中评估 RBE 之外，我们还研究了 RBE 在不存在标记数据的非监督环境中的适用性。

在这种设置中，我们面临着一个大型的未标记语料库T和给定的规则集R。由于固有的规则集，这种设置特别具有挑战性。

规则的泛化问题。在此设置中松散地应用 ing 规则会产生模型

过度拟合规则集的分布，如图所示
如表3 所示。为了解决这个问题，我们设计了三种
不同的基于语义聚类的策略
在无人监督的环境中确定规则质量：
均值、连续和距离聚类。给定
未标记的语料库T “ tt1, t2, ..., tnu,规则集
R “ tpr1, e1q, ...,prn, enqu和阈值k,我们
首先使用预训练的方法对整个语料库T进行编码
句子嵌入模型Eθ。就我们而言，我们
使用 MPNet 的微调版本（Song 等人，
2020)来自句子转换器库。收到我们的编码语料库
EθpTq后，对于
均值和concat，我们构造一个规则嵌入
$\overline{\theta}$ 对于规则集中的每个规则ri 。在平均策略中，这是通过取所有规则的平均值获得的
示例_ $\overline{\theta}$q “p $\frac{1}{米}$ R M$_{米}$ e$_{平方米}$。对于康卡特来说，这
通过连接所有规则示例来计算
µpriq “ Eθpe $_{1}$ }...}e $_{mq}$并对 con 进行编码，
串联表示。一旦r $\overline{\theta}$
然后，我们标记语料库中余弦值的每个文本
相似度在阈值 k 内：

使用规则进行内容审核（完全监督）												
	仇恨解释				拼图				计算机辅助设计			
模型	精确召回F1			Acc 精确召回 F1				Acc 精确召回 F1				加速器
仇恨解释规则	0.609	0.983 0.752 0.615			-	-	-	-	-	-	-	-
仇恨+滥用规则	0.755	0.687 0.719 0.682			0.164	0.361 0.226 0.972			0.586	0.193 0.290 0.909		
CAD 规则	-	-	-	-	-	-	-	-	0.110	0.842 0.194 0.325		
伯特`	0.808	0.841 0.824 0.787 0.854			0.459	0.729 0.563 0.987 0.674			0.445	0.421 0.433 0.893		
MPNet^	0.795	0.823 0.783			0.510	0.581 0.989			0.519	0.417 0.463 0.906		
举例`△ 0.758 举例^△ 0.790 举例`°举		0.903 0.824 0.771 0.891			0.581	0.625 0.602 0.991 0.746			0.416	0.478 0.445 0.885		
例^°举例`;		0.837 0.795			0.508	0.604 0.989			0.484	0.468 0.476 0.900		
	0.738	0.912 0.816 0.756			-	-	-	-	-	-	-	-
	0.779	0.893 0.832 0.786			-	-	-	-	-	-	-	-
	-	-	-	-	-	-	-	-	0.512	0.378 0.435 0.905		
举例规则^;	-	-	-	-	-	-	-	-	0.508	0.448 0.476 0.905		

表 1:仇恨言论分类数据集在完全监督环境下的实验结果。 `使用 BERT （Devlin 等人，2018)作为基础模型。 ^使用 MPNet (Song et al., 2020)作为基础模型。 °使用 HateXplain 规则集。 △使用仇恨+辱骂规则集。 ;使用 CAD 规则集。注意： HateXplain 规则集和上下文滥用数据集 (CAD) 规则集仅适用于他们各自的数据集。

$$fptiq \quad \#\ 1, \text{如果} \ simpr \ \theta, \ \text{埃塔普蒂克} \check{e}\ k \quad (3)$$

$$\#\ 0, \text{否则}$$

与 Mean 和 Concat 策略相反

ies,距离策略采用规则消除方法。给定一个未标记的语料库T

tt1, t2, ..., tnu,规则集R “ tpr1, e1q, ...,prn, enqu,和阈值k,我们首先使用规则集Ri来嘈杂地标记整个语料库： xt Ñ t1, Hu

这样每个规则都与一个覆盖集配对

R “ tpr1, e1, c1q, ...,prn, en, cnqu其中ci是ri涵盖的文本集。接下来,对于每条规则,我们在其封面集EThepciq中编码文本并计算每个嵌入之间的平均余弦距离及其在ci中的相邻示例。

$$avgDistpE\theta pciqq \ “ \ \frac{1}{n} \ 在 \ 迪普 \ _{j,j}^{我} \ 词_1 \quad 问 \quad (4)$$

最后,一旦每条规则的平均距离为计算,使用定义的阈值k,我们翻转封面集中任何弱标记的示例,如果该规则的平均距离高于阈值C:

$$fptiq \quad \#\ 1, \text{如果} \ avgDistpriq \check{e}\ k \quad (5)$$

$$\#\ 0, \text{否则}$$

4 结果与讨论

我们分析我们的实验结果,详细我们的见解,并讨论应用RBE 进行可解释的仇恨言论检测的影响。

评估指标:准确率、召回率和监督环境中每个数据集的 F1 分数报告于表1 中。由于高度偏斜班级分布,我们倾向于宏观 F1 分数作为我们的主要评价指标。我们还报告准确性分数（完整集的条目分数标签匹配)作为另一个指标。

4.1 监督绩效

表1报告了我们对三种仇恨言论的结果监督环境中的分类数据集。我们观察 RBE 能够优于 SOTA基于 Transformer 的模型 BERT 和 MPNet F1 得分为 1.3/1.4%、4.1/2.3% 和 4.3/1.3%分别是 HateXplain、Jigsaw 和 CAD。这改进凸显了杠杆作用的影响我们框架的训练过程中的规则。此外,值得注意的是,这种增加中的每个规则仅使用 1 个副本即可实现规则集。这些例子也被用来训练比较基线模型,确保所有方法接受了相同数量的训练样品。这进一步展示了轻量化灵活的RBE是集成到内容修改工作流程中。对于 HateXplain,我们的实验表明 MPNet 作为初始化编码器与 HateXplain的组合仇恨+滥用规则集提供最佳性能。经过更深入的分析,我们发现这是由于两个主要因素：

1)规则集大小和对齐方式- 如上所述在第3.2节中， HateXplain 理性规则集使用基本原理标签自动制作来自专家注释者。这产生了强大的能够识别大量的规则集HateXplain 数据集中的仇恨内容如图所示

举例规则 规则基础			
数据集	文本	解雇规则	复制
HateXplain完全同意每一次个人互动	与这些酷儿一起强化知道我已经准备好 严重的精神疾病和 令人讨厌的启动	if contains(酷儿) Ñ 1, 0	是的,但是酷儿们太自以为是了,以至于不能让他们最初被包括在内 同性恋和恋童癖社区正在共同努力,nambla 的发起者是 现在喜欢男孩的男同性恋者的所有恋童癖子集都是假设的成员 南布拉仍然存在
拼图	为什么他们让这么多同性恋者上台 该死的节目从什么时候开始就还好 成为同性恋。	if contains(so ^”gay”q Ñ 1, 0停止恢复我的编辑 你的so f**** gay get af**** life your f**** 或者去get 放置或其他什么	
计算机辅助设计	多么小啊****	if contains(b****) Ñ 1, 0	不,今天是金枪鱼 b****

表 2:规则基础解释示例。该表显示了 RBE 生成的追踪模型预测的示例。
通过显示负责的规则和范例,规则作者和用户能够更好地理解模型预测,并可以
自动调整规则集以进一步提高模型性能。

HateXplain 理由的高召回率
表1 中的规则集。此外,当应用于
HateXplain 数据集、HateXplain 基本原理
与之前的相比,规则集总共生成 577 条规则
源自仇恨+滥用规则的 377 条规则
集,允许更多的规则表示
模型进行对比。

2)嵌入初始化- 开箱即用,
预训练的 BERT 不会产生有意义的结果
不同的句子表示。在实践中,
BERT [CLS] 令牌以及平均 BERT 输出在下游微调后可以包含有
用的信息。 BERT 表明了这一点

表1中的性能。然而,当预训练的模型输出在所有维度上汇集并用于
计算语义相似度时,

即使对于完全不同的输入文本,这也会产生相似的表示。结果,如果
应用
到 HateXplain 数据集,无需任何微调,
BERT 嵌入获得了精度、召回率和
F1分数分别为59%、100%和75%,
每个例子都被标记为可恨的。这
缺乏多样化的句子表示加上
详细的规则集,例如 HateXplain 基本原理
规则集导致最初偏向可恨的示例,如高召回分数所示。

因此,利用预先训练的句子嵌入器,
例如 MPNet,其预训练任务针对语义嵌入进行了更优化,从而获
得更好的结果
表现。当使用CAD 的派生规则集时,我们观察到类似的趋势。注:当

随着训练时间的延长,BERT 模型的偏差会随着句子表示形
式的变化而减少
学到了。

关于拼图和上下文滥用数据集使用
仇恨+滥用列表和派生的 CAD 规则集,
RBE 的表现优于 SOTA,幅度更大
分别为 4.1/2.3% 和 4.3/1.3%。相反
对于 HateXplain 来说,这两个数据集对于非仇恨示例更加不平
衡,
因此更能代表现实世界的情况
考虑大多数内容的内容审核

是良性的。这种增强的性能亮点
结合逻辑规则来协助的力量
模型学习以及 RBE 更好的能力
概括规则。如表 1 所示,单独
仇恨+滥用规则集在每个方面都表现不佳
数据集的精确度和召回率。尽管 RBE 的
依赖这个规则集来指导模型学习,
当与标记的训练数据结合时,RBE 是
能够限制过于笼统的规则
并利用其对语义相似性的理解来扩展脆弱的规则,无论基础如何

模型。此外,当使用严重过度拟合 CAD 数据集的CAD 规则集时,

如偏斜召回分数所示,RBE 仍然是
能够超越基线。

域外规则集我们的仇恨+滥用规则
set 是与任何规则无关的通用规则集
评估数据集,从而得出域外
规则集。这提供了域外的示例
使用并非源自 tar get 数据集的规则的性能。我们观察到,即使在申
请时
RBE 通过仇恨+滥用规则集,我们能够
优于每个数据集的基线。什么时候
将 RBE 应用于新的域设置,所有这一切
需要为此制定附加规则
新域名。这可以手动完成,也可以更多
通过自动派生规则来可扩展
新域数据。

4.2 可解释性

除了性能的提升之外,另一个
RBE 的优势在于其执行能力
规则接地。正如第2.2 节中所解释的,规则接地使我们能够
将模型预测追溯到其各自的规则,并伴随

定义该规则的示例。表2显示
从每个中提取的规则基础示例
我们测试的数据集。从本质上讲,规则基础在 RBE 中
实现了两个主要功能:

1)可定制性/规则集适应:给定
广泛的在线应用程序、内容模组

使用规则进行内容审核（无监督）												
	仇恨解释				拼图				计算机辅助设计			
模型	精确召回F1			Acc 精确召回 F1				Acc 精确召回 F1				加速器
仇恨解释规则	0.609	0.983 0.752 0.615 0.687			-	-	-		-	-	-	
仇恨+滥用规则	0.755	0.719 0.682			0.164	0.361 0.226 0.972			0.586	0.193 0.290 0.909		
CAD 规则	-	-	-	-	-	-	-	-	0.110	0.842 0.194 0.325		
伯特`°	0.606	0.990 0.752 0.613			-	-	-	-	-	-	-	-
伯特`△	0.747	0.717 0.732 0.688			0.234	0.461 0.310 0.977			0.587	0.205 0.303 0.909		
伯特`;	-	-	-	-	-	-	-	-	0.107	0.865 0.191 0.290		
MPNet`°	0.611	0.991 0.756 0.621			-	-	-	-	-	-	-	-
MPNet`△	0.652	0.850 0.738 0.641			0.247	0.501 0.331 0.977			0.642	0.199 0.304 0.912		
MPNet`;	-	-	-	-	-	-	-	-	0.111	0.840 0.196 0.335		
举例规则（距离)°	0.614	0.983 0.756 0.623 0.955			-	-	-	-	-	-	-	-
举例规则（距离)△	0.629	0.758 0.639			0.358	0.284 0.317 0.986			0.280	0.322 0.299 0.854		
举例规则（距离）；	-	-	-	-	-	-	-	-	0.166	0.522 0.252 0.701		
举例规则（Concat)°	0.621	0.950 0.751 0.626 0.985			-	-	-	-	-	-	-	-
举例规则（Concat)△	0.612	0.755 0.621			0.189	0.052 0.081 0.987			0.175	0.437 0.250 0.747		
示例规则（Concat）；	-	-	-	-	-	-	-	-	0.178	0.437 0.253 0.750		
举例规则（平均值)°	0.612	0.983 0.754 0.620			-	-	-	-	-	-	-	-
举例规则（平均值)△	0.636	0.944 0.760 0.646			0.188	0.124 0.149 0.984			0.294	0.273 0.283 0.866		
举例规则（平均值）；	-	-	-	-	-	-	-	-	0.189	0.411 0.259 0.772		
无监督预训练												
举例规则（平均值)△	0.641	0.954 0.767 0.656 0.968			0.166 .626		0.262 0.961		0.260	0.320 0.287 0.846		
举例规则（距离)△	0.617	0.753 0.624			0.203	0.465 0.283 0.974			0.484	0.236 0.317 0.902		

表 3:所有集群策略的无监督性能。 °使用 HateXplain 规则集。 △使用仇恨+滥用规则集。
;使用 CAD 规则集。注意： HateXplain 规则集不适用于 Jigsaw 和上下文滥用数据集 (CAD)。

生成系统需要能够轻松适应不断出现的仇恨内容趋势。特别是在
在线社交环境,这些平台的专家用户不断寻找新的有趣的方式

绕过审核系统。此外,新
每天都会引入术语和俚语。 RBE 能够无缝寻址

通过促进规则引导的学习来解决这些问题。
通过定义一条新规则并添加至少一条
例如,RBE 能够捕获新兴内容,而无需重新训练。此外，

RBE的用户可以轻松修改现有规则
可能太宽泛并添加更多示例
以可控的方式进一步细化预测。

2)预测透明度:通过促进
通过规则基础的模型解释,用户
在线系统应提供切实的指导
他们的内容将被标记,可能会增加用户
对系统的信任。此外,这可以直接指示规则作者的内容类型

想要适度。

4.3 无监督表现

表3报告了我们在无监督环境下的结果。我们观察到
RBE 能够跑赢大市
SOTA 在噪声规则标记样本上进行训练
HateXplain 和 Jigsaw 数据集,同时也像在所有三个数据集上一样执行规则集。
在每个数据集中,我们发现 RBE 的距离
基于策略产生最一致的性能,在 HateXplain 上优于 SOTA 和

CAD 在曲线锯上的性能与 SOTA 相当。我们观察到这种性能稳定性
这是由于该策略的规则消除目标。
与 Mean 和 Concat 策略相反
其重点是导出规则表示
自我监督的方式,距离策略反而侧重于消除过度概括的规则

其封面示例集在语义上不同。这在以下情况下特别有用：

由于数量较多,精度分数较低
误报。

对于 Jigsaw,我们观察到与 SOTA 相比性能略有下降。经过进一步分析,我们认为这是 RBE 在这种情况下过度依赖规则集的结果,特别是

对于均值和连接策略。这是因为
由于其覆盖集C的标签,规则集直接影响导出的规则嵌入。

当规则集过于笼统时，
Jigsaw 上的仇恨+滥用规则的情况下,RBE 很可能
以匹配规则集的分布。我们发现
执行自监督模型预训练
(Gao et al., 2021)关于目标语料库的规避
这种趋势适用于 Mean 和 Concat 策略。高手
因此,通过更精细的规则集,预计性能会提高,如 HateXplain 和 CAD 中所示。

5 相关工作

在检测仇恨方面一直在积极开展工作
语言中的言语（Poletto 等人， 2021； Al Makhadmeh 和 Tolba， 2020； Schmidt 和 Wie

盛大， 2017）。仇恨语音检测已被证明是一项细致且困难的任务，导致针对问题各个方面的方法和数据集的开发（Vidgen 等人， 2021； Mathew 等人， 2020； Mody 等人， 2023）。然而，很少有人尝试关注这些模型的可解释性，这是对其在线使用越来越关注的领域（Tarasov， 2021； Haimson 等人， 2021），从而导致持续使用功能较弱但更易于解释的方法，例如规则。先前的工作已经探索将逻辑规则纳入模型学习中。阿瓦斯蒂等人。 （2020）提出通过将规则与示例配对并训练去噪模型来从规则中进行弱学习。

然而，这需要为所有输出类定义规则，使其不适用于仇恨言论检测任务。此外，该方法仅侧重于减少规则范围来解决过度泛化问题。它不能同时解决图 1 中演示的过度特异性问题。最后，该方法没有提供在推理过程中解释模型预测的方法。徐等人。 （2021）提出了一种通过规则控制神经网络训练和推理的方法，然而，他们的框架将规则表示为需要复杂扰动的可微函数，使其更适合数字规则，例如医疗保健和金融中定义的规则语言的复杂细微差别。普里赞特等人。 （2022）提出了一个从一小组标记数据自动归纳符号规则的框架。然而，这些规则源自低容量的机器学习模型，因此不适合人类阅读或解释。

六，结论

我们引入了 Rule By Example，这是一种基于示例的对比学习框架，可以从逻辑规则中学习，以实现准确且可解释的仇恨语音检测。具体来说，我们提出了一种新颖的双编码器模型架构，旨在产生有意义的规则和文本表示。 RBE 利用一种新颖的基于范例的对比学习目标，该目标融合了相似类别的规则表示和文本输入。我们分享了仇恨言论检测的三个公共数据集的结果，这些结果验证了示例规则框架不仅可以大大优于初始规则集，而且可以在两种监督方式中优于基线 SOTA 分类方法和无监督的设置。此外，RBE 还支持规则基础，从而提供SOTA分类方法所没有的更可解释的模型预测优势，以及通过规则集适应实现的额外灵活性。

7 限制

在本节中，我们将讨论示例规则方法的一些限制。

7.1 对监管的依赖

在我们的示例规则方法中，需要一组规则和每个规则一个示例，这意味着即使对于“无监督”的实验设置，也需要一定程度的专家监督。在某些情况下，这可能是一个令人望而却步的成本。有一些潜在的方法可以以无监督的方式为每个规则选择一个示例，例如对规则触发的示例进行聚类，这可以在未来的工作中进行探索。然而，规则的创建本身意味着某种形式的专家监督，它将有关分类任务的知识提炼成可解析的函数。

7.2 与规则相比成本增加

尽管“示例规则”方法生成的双编码器模型比其派生的规则集性能要高得多，但它仍然具有其他深度学习方法的成本限制。双编码器需要昂贵得多的计算 (GPU) 来进行初始训练和随后在生产环境中的推理。即使使用昂贵的 GPU，延迟成本也不可避免地比大多数简单的逻辑规则高得多。

对于某些应用，双编码器模型的质量增益可能不值得增加运营成本。

7.3对质量规则和范例的依赖

由于示例规则方法基于规则集和相关的示例来学习，因此这些规则和示例的质量可能会影响下游双编码器模型的质量。

如果编写的规则集和选择的示例质量不高，直观上双编码器模型的质量就会受到影响。在无监督环境中尤其如此，其中规则被用作噪声标签函数。未来可能的扩展是研究规则和示例质量对派生双编码器模型性能的影响。

8 道德

仇恨言论检测是一项复杂的任务。减少编写一组简单逻辑规则的任务可能会导致规则作者在这些规则中编码硬偏差。例如,如果使用组内词或身份术语作为将内容识别为仇恨言论的规则,这可能会导致删除问题。

举例规则方法可以潜在地减少这些情况,例如通过学习更好的规则表示并识别术语何时用作群体内演讲而不是用作侮辱或诽谤。然而,衍生的双编码器也面临着传播和放大这些偏差的风险（Hall et al., 2022）,造成比原始规则集更大的意外伤害。

无论是使用规则集还是使用更复杂的模型,通过额外的 Responsible AI 工作流（例如分类器行为和测量的审查)支持分类器非常重要

的公平性。

致谢

我们感谢匿名审稿人的反馈和建议。这项工作是由Microsoft Cloud & AI 的ROAR（Responsible & Open AI Research)团队进行的。在密歇根大学,克里斯托弗·克拉克 (Christopher Clarke) 得到了国家科学基金会 NSF1539011奖的部分支持。

参考

扎菲尔·阿尔·马哈德梅和阿姆·托尔巴。 2020。使用杀手级自然语言处理优化集成深度学习方法进行自动仇恨语音检测。计算,102（2）:501-522。

Abhijeet Awasthi、Sabyasachi Ghosh、Rasna Goyal 和Sunita Sarawagi。 2020.从概括标记范例的规则中学习。

Christopher Clarke、Joseph Peper、Karthik Krishna murthy、Walter Talamonti、Kevin Leach、Walter Lasecki、Yiping Kang、Lingjia Tang 和 Jason Mars。
2022 年。一个代理统治一切:迈向多代理对话式人工智能。计算语言学协会的调查结果:ACL 2022,第 3258-3267 页,爱尔兰都柏林。计算语言学协会。

Emmanuel Gbenga Dada、Joseph Stephen Bassi、Haruna Chiroma、Shafi i Muhammad Abdulhamid、Ade bayo Olusola Adetunmbi 和 Opeyemi Emmanuel Ajibuwa。
2019.垃圾邮件的机器学习过滤:回顾、方法和开放研究问题。Heliyon,5(6):e01802。

托马斯·戴维森、达纳·沃姆斯利、迈克尔·梅西和英格玛·韦伯。 2017。自动仇恨言论检测和攻击性语言问题。

雅各布·德夫林、张明伟、肯顿·李和克里斯蒂娜·图塔诺娃。 2018. Bert:深度预训练用于语言理解的双向转换器。

高天宇、姚兴成、陈丹琪。 2021年
Simcse:床上用品中句子的简单对比学习。

塔尔顿·吉莱斯皮。 2018 年。互联网的守护者:平台、内容审核以及塑造社交媒体的隐藏决策。耶鲁大学出版社。

R·哈德塞尔、S·乔普拉和 Y·勒昆。 2006.通过学习不变映射来降维。 2006年IEEE 计算机学会计算机视觉和模式识别会议(CVPR 06),第 2 卷,第 1735-1742 页。

奥利弗·L·海姆森、丹尼尔·德尔莫纳科、聂佩佩和安德里亚·韦格纳。 2021.不成比例的拆除以及保守派、跨性别者和黑人社交媒体用户的不同内容审核体验:边缘化和适度的灰色地带。程序。

ACM 嗡嗡声计算。互动,5（CSCW2）。

梅丽莎·霍尔、劳伦斯·范德马滕、劳拉·古斯塔夫森、麦克斯韦·琼斯和亚伦·阿德科克。 2022.偏置放大的系统研究。

普雷纳·朱尼贾 (Prerna Juneja)、迪皮卡·拉玛·萨勃拉曼尼亚 (Deepika Rama Subramanian) 和塔努什里·米特拉 (Tanushree Mitra)。 2020 年。透过镜子:Reddit审核实践透明度研究。程序。 ACM 嗡嗡声计算。互动， 4（小组）。

大卫·坎普和艾米丽·艾金斯。 2021 年民意调查:75% 的人不相信社交媒体会做出公平的内容审核决定,60% 的人希望对他们发布的帖子有更多的控制权
湖。

Vivian Lai、Samuel Carton、Rajat Bhatnagar、Q. Vera Liao、Yunfeng 张和 Chenhao Tan。 2022 年。
通过条件委托进行人机协作:内容审核的案例研究。 2022 年 CHI 计算系统人为因素会议记录,CHI 22,美国纽约州纽约市。

计算机器协会。

凯文·李. 2022.规则与机器学习:为什么你需要两者都获胜:筛选。

刘毅、Pinar Yildirim 和 Z.John 张。 2022 年。收入模式和技术对内容审核策略的影响。营销科学， 41(4):831-847。

Jitendra Singh Malik、Guansong Pang 和 Anton van den Hengel。 2022.仇恨言论的深度学习检测:比较研究。

宾尼·马修、Punyajoy Saha、Seid Muhie Yimam、
克里斯·比曼、帕万·戈亚尔和阿尼梅什·穆克吉。 2020.
Hatexplain：可解释的仇恨言论检测的基准数据集。

Devansh Mody、Yi Dong Huang 和 Thiago Eustaquio
阿尔维斯·德·奥利维拉。 2023.精心策划的仇恨数据集
社交媒体文本上的语音检测。信中的数据，
46:108832。

Animesh Mukherjee、Mithun Das、Binny Mathew 和
普尼亚乔伊·萨哈。 2022。仇恨言论：检测、缓解及其他@aaai。

法比奥·波莱托、瓦莱里奥·巴西莱、曼努埃拉·桑吉内蒂
克里斯蒂娜·博斯科和维维安娜·帕蒂。 2021. 资源
和仇恨言论检测的基准语料库：a
系统审查。语言资源与评估，55(2)：477–523。

Reid Pryzant、杨紫怡、徐一冲、朱晨光、
和迈克尔曾。 2022.自动规则归纳
用于可解释的半监督学习。

尼尔斯·雷默斯和伊琳娜·古列维奇。 2019.句子-bert：
使用 siamese bert 网络进行句子嵌入。

安娜·施密特和迈克尔·韦根。 2017.一项调查
使用自然语言处理进行仇恨言论检测。第五国际会议录

社交媒体自然语言处理研讨会，第 1-10 页，西班牙巴伦西亚。协会
用于计算语言学。

Sungyong Seo、Sercan O. Arik、Jinsung Yoon、Xiang
张基赫·索恩 (Kihyuk Sohn) 和托马斯·普菲斯特 (Tomas
Pfister)。 2021.用规则表示控制神经网络。

志户雄介、刘显奇、梅泽圭介。
2022. C2C 市场中的文本内容审核。第五届研讨会论文集

电子商务和 NLP (ECNLP 5)，第 58-62 页，
爱尔兰都柏林。计算语言学协会。

宋凯涛，谭旭，秦涛，陆剑峰，刘铁岩。 2020.Mpnet：语言理解的
屏蔽和排列预训练。

凯蒂·塔拉索夫。 2021.为什么内容审核会产生成本
数十亿，对于 facebook、twitter、youtube 来说是如此棘手
和别的。

Sahaj Vaidya、Jie Cai、Soumyadeep Basu、Azadeh
纳德里、东熙·伊维特·沃恩和阿里特拉·达斯古普塔。
2021.视觉分析干预的概念化
用于内容审核。 2021 年 IEEE 可视化
会议 (VIS)，第 191-195 页。

Bertie Vidgen、Dong Nguyen、Helen Margetts 和 Patricia
罗西尼和丽贝卡·特罗布尔。 2021.介绍
CAD：上下文滥用数据集。诉讼中
计算语言学协会北美分会 2021 年会议的内容：

人类语言技术，第 2289-2303 页。
在线的。计算语言学协会。

库尔特·瓦格纳和布隆伯格。 2021 年。Facebook这么说
已花费 130 亿美元用于安全和安保工作
自2016年以来。

托马斯·沃尔夫、莱桑德尔·迪布特、维克多·桑、朱利安
肖蒙德、克莱门特·德兰格、安东尼·莫伊、皮埃里克·西斯塔克、
蒂姆·罗特、雷米·卢夫、摩根·芬托·伊茨、乔·戴维森、萨姆·施
莱弗、帕特里克·冯·普拉滕，
Clara Ma, Yacine Jernite, Julien Plu, 徐灿文，
特文·勒·斯考、西尔万·古格、玛丽亚玛·德拉梅、
昆汀·洛斯特和亚历山大·拉什。 2020。变形金刚：最先进的自然
语言处理。
2020 年实证会议论文集
自然语言处理方法：系统
演示，第 38-45 页，在线。协会
用于计算语言学。

徐灿文、郭大亚、段南和朱利安·麦考利。
2022. Laprador：用于零样本文本检索的无监督预训练密集检索
器。

张雨辰. 2019.阻止不良内容
发布并建立更好的社区。

ACL 2023 负责任的 NLP 检查清单

A 对于每次提交：

✓ A1。您是否描述了您工作的局限性？
7

✓ A2。您是否讨论过您工作中的任何潜在风险？
8号

✓ A3。摘要和引言是否总结了论文的主要主张？
1

A4。您在写这篇论文时使用过人工智能写作助手吗？
留空。

B ✓您是否使用或创造了科学制品？
2

B1您是否引用了您使用的工件的创建者？
没有反应。

B2。您是否讨论过任何工件的使用和/或分发的许可或条款？
没有反应。

B3。您是否讨论过您对现有工件的使用是否与其预期用途一致（前提是已指定）?对于您创建的工件，您是否指定了预期用途以及是否与原始访问条件兼容（特别是，出于研究目的而访问的数据的衍生物不应在研究环境之外使用）？

没有反应。

B4。您是否讨论了检查收集/使用的数据是否包含任何命名或唯一识别个人或攻击性内容的信息所采取的步骤，以及保护/匿名化所采取的步骤？

没有反应。

B5。您是否提供了工件的文档，例如领域、语言和语言现象的覆盖范围、所代表的人口群体等？

没有反应。

✓ B6。您是否报告了您使用/创建的数据的相关统计数据，例如示例数量、训练/测试/开发拆分的详细信息等?即使对于常用的基准数据集，也要包括训练/验证/测试分割中的示例数量，因为这些为读者理解实验结果提供了必要的背景。例如，大型测试集上的准确性的微小差异可能很重要，而在小型测试集上则可能不那么重要。

留空。

C ✓您进行过计算实验吗？
3

✓ C1。您是否报告了所使用模型中的参数数量、总计算预算（例如，GPU 时间）以及使用的计算基础设施？ 3

ACL 2023使用的负责任的 NLP 检查表采纳自NAACL 2022，还增加了一个关于人工智能写作辅助的问题。

✓ C2。您是否讨论了实验设置,包括超参数搜索和最佳找到的超参数值？ 3

✓ C3。您是否报告了有关结果的描述性统计数据（例如,结果周围的误差线、实验组的汇总统计数据）,以及您是否报告最大值、平均值等或仅报告单次运行是否透明？ 4

✓ C4。如果您使用现有的包（例如,用于预处理、标准化或评估）,您是否报告了所使用的实现、模型和参数设置（例如,NLTK、Spacy、ROUGE等）？ 3

D 您是否使用人类注释者（例如众包工作者)或与人类参与者一起进行研究?

留空。

D1。您是否报告了向参与者提供的说明的全文,包括屏幕截图、参与者或注释者的任何风险免责声明等?

没有反应。

D2。您是否报告了有关如何招募（例如,众包平台、学生)和付费参与者的信息,并讨论了考虑到参与者的人口统计（例如,居住国家）,此类付款是否足够?

没有反应。

D3。您是否讨论过是否以及如何获得您正在使用/管理其数据的人的同意?例如,如果您通过众包收集数据,您向众包工作者发出的指示是否解释了如何使用这些数据?

没有反应。

D4。数据收集方案是否得到伦理审查委员会的批准（或确定豁免）？
没有反应。

D5。您是否报告了作为数据来源的注释者群体的基本人口统计和地理特征?

没有反应。