使用可控文本生成模型的集成大语言模型自动仇恨言论识别

Ryutaro Oshima* 和 Yuya Hosoda † 和 Youji Iiguni*
* 日本大阪大学，电子邮件：
oshima@sip.sys.es.osaka-u.ac.jp
† 日本立命馆大学，电子邮件：
y-hosoda@fc.ristumei.ac.jp

arXiv:2601.04654v1 [eess.AS] 8 Jan 2026

摘要——本文提出一种使用大语言模型（LLM）的仇恨言论自动语音识别（ASR）模型。所提出的方法将自动语音识别模型的编码器与大语言模型的解码器相结合，从而能够同时执行转录和审查任务，防止有害内容曝光。要通过指令微调使大语言模型用特定词元掩蔽仇恨相关词语，需要带标注的仇恨言论数据集，而这类数据集十分有限。我们使用大语言模型，在文化语境和示例的引导下，采用思维链（CoT）提示技术生成文本样本，再通过文本转语音（TTS）系统将其转换为语音样本。然而，其中一些样本包含带有仇恨相关词语的非仇恨言论，这会降低审查性能。本文筛选出被文本分类模型正确标记为仇恨内容的样本。通过调整正确判定的模型数量阈值，我们可以控制生成的数据集中的仇恨程度，从而以循序渐进的方式通过课程学习训练大语言模型。实验结果表明，所提出的方法对仇恨相关词语的掩码准确率达到58.6%，优于此前的基线方法。我们还证实，课程训练有助于提高转录和审查任务的效率。

本文包含冒犯性或仇恨内容，但未指向任何特定群体或个人。

I. 引言

YouTube 和 TikTok 等基于音频的社交网络服务（SNS）平台上的流媒体系统，塑造了全球实时通信的方式。情感联系会影响观众的购买意愿，这凸显了流媒体在数字营销中的重要地位 [1], [2]。除了这些益处，平台还必须应对有害内容可能迅速跨境传播这一挑战。仇恨言论针对具有种族和宗教等属性的个人或群体，仍然是一个社会问题。接触仇恨言论会对受害者造成心理伤害，也会削弱直接受害者以外的旁观者的同理心[3], [4]。如果缺乏适当的防护措施，这类行为可能会在日常交流中被视为可以接受 [5]。针对仇恨言论的审查机制对于维护更健康商业环境的完整性和保护人权至关重要。

研究人员开发了多种方法来检测不当文本内容，例如讽刺 [6], [7]和隐私信息 [8], [9]。由于语境理解至关重要，因此，自然语言处理模型已被用于检测仇恨言论 [10], [11]。基于基于变换器的双向编码器表示（BERT）的模型能够在基于文本的社交网络服务平台上识别多种语言中的仇恨内容[12]。Sabry 等人提出了一种基于文本到文本迁移变换器（T5）的模型进行仇恨语言识别的方法，并强调了利用模型可解释性检测仇恨相关词语的潜力 [13]。大语言模型更擅长理解更广泛的语境结构，并能够通过指令微调生成语义连贯的句子，从而使其能够专精于特定任务 [14], [15]。据报道，引入大语言模型能够提高仇恨言论检测的准确率 [16], [17]。

本文始终关注口语话语中的仇恨内容检测；因此，除非另有说明，“仇恨言论”一词指包含此类表达的音频（语音）。基于音频的社交网络平台上的仇恨言论检测方法是基于文本的检测方法的延伸。Nandwana 等人 [18] 提出了一种多模态系统，用于识别视频中的仇恨言论，重点在于基于文本的检测。An 等人 [19]使用自动语音识别模型转录语音内容，并使用 BERT 模型对其进行分类。分类模型的局部可解释模型无关解释（LIME）能够揭示单个词语如何影响分类结果，从而识别仇恨相关词语。然而，LIME 难以正确识别习语表达或细微语境。此外，由于多个仇恨相关词语对分类结果的影响分散在各处，它无法在单个话语中检测出这些词语。

本文提出了一种利用大语言模型的自动仇恨言论识别模型，该模型能够检测单个句子中的多个仇恨相关词语。由于内容过滤器的存在，直白的仇恨言论表达通常会受到限制，因此难以将包含仇恨相关词语的转录文本直接输入大语言模型。所提出的方法聚焦于一种多任务模型，该模型将自动语音识别模型的编码器集成到大语言模型中作为解码器 [20], [21]。由于仇恨相关词语以隐式方式输入大语言模型，该集成模型能够同时执行转录

和审查任务，并使用特定词元将这些词语屏蔽。由于参数规模较大，指令微调需要大量语音样本，但带标注的音频数据集数量有限。我们使用大语言模型生成文本样本，再通过文本转语音（TTS）模型将其转换为语音样本。然而，这些样本中可能包含带有仇恨相关词语的非仇恨言论样本，从而降低审查性能。我们使用文本分类模型对仇恨程度进行评分，以此将这些样本排除。

本文的贡献如下：
1）我们通过思维链提示技术，利用示例句子、歧视背景和目标群体来定义仇恨言论的概念，从而能够生成更真实、包含指定仇恨相关词语的仇恨言论文本。2）所提出的方法可以通过阈值处理控制生成数据集的仇恨程度，使我们能够通过课程学习逐步训练集成自动语音识别模型。3）实验结果表明，所提出的方法能够同时实现转录和审查处理，并凸显了编码器和解码器的微调过程对其性能的影响。

II. 方法

A. 带标注仇恨言论数据集生成

我们采用数据增强来解决仇恨言论带标注数据集不足的问题。HateXplain [22]包含来自文本类社交网络服务平台 X（原 Twitter）和 Gab 的 20,148 个句子，用于分析促成仇恨言论感知的语言表达。三名标注员分别对每个句子进行标注，以判定其类别（“仇恨”“冒犯性”“正常”或“未决定”）。当句子包含仇恨内容时，他们还会标注目标群体（如“女性”和“非裔”），并突出显示支持其判断的词语。三名标注员一致将 5,945 个句子标记为“仇恨”，这些句子是生成仇恨言论样本的核心数据。然而，一些样本包含俚语和表情符号等视觉符号，自动语音识别模型难以准确转录。我们手动筛选出其中 602 个不含此类内容的句子，作为原始数据集。

首先，我们从仇恨句子中提取高频词，然后使用五种最先进的文本分类模型 [12], [19], [23], [24] 来筛选关键词。这些模型在不同的数据集上进行了预训练，用于识别带有“仇恨”和“正常”两个标签的主题。为提高所选关键词的可靠性，本文将至少被五种模型中的三种判定为仇恨的词定义为仇恨相关关键词。其次，我们使用 Qwen2.5-1.5B-Instruct 生成仇恨言论样本，包括包含指定关键词的样本。

表 I 用于生成仇恨言论样本的思维链提示

Step	目标	提示
0	分配 Role	你是一名专门从事 仇恨言论检测的人工智能助手。你的目标是 生成高质量的训练数据。
1	理解 Hate 语音	仇恨言论是任何形式的交流， 这些交流会基于某些特征歧视个 人或群体 例如种族、宗教、族裔、性别、 性取向、残疾或其他 受保护特征。
2	审阅 示例	以下是两个仇恨言论示例。 示例 1： [仇恨言论 1],示例 2： [仇恨言论 2]
3	生成 New 示例	根据上述定义和示例 生成一条新的仇恨言论 严格遵循以下 仇恨言论特征的陈述。生成的 句子必须包含以下 关键词。必选关键词: [关键词]
4	确认 Text	务必在生成的文本中包含指定的关键字- 词。不要使用 被审查的词语或不完整的短语。

文本生成过程采用思维链提示策略，如表 I 所示。首先，该方法将人工智能助手设定为生成高质量训练数据的角色，以检测仇恨言论内容。为清晰说明任务，我们向大语言模型提供基于种族、宗教和性别等属性的仇恨言论定义。随后，提示中给出了原始数据集中两个目标群体相同的示例。文本生成任务要求包含一个指定词作为“必选关键词”。最后，提示强调该关键词的重要性，同时避免使用经过审查或不完整的表达。

我们将由此生成的共 21,368 个句子称为仇恨言论候选句子。请注意，这些句子也包含非仇恨言论内容，因为大语言模型会根据训练数据中的统计模式选择词语，有时会优先考虑一般相关性，而非严格遵循特定任务。本文使用仇恨言论检测模型对其进行筛选。图 1 展示了筛选仇恨言论候选句子的过程。这些模型会对每个示例进行分类，其中“正常”表示无问题的标签，“仇恨”则表示各种不当内容，例如“仇恨”或“冒犯性”内容。根据“仇恨”标签的数量，我们将仇恨程度划分为 0 至 5 级。例如，在图 1 中，Ex.1 的五个模型中有两个将其判定为“仇恨”，因此其仇恨程度为 2。基于程度的标注使我们能够根据模型间共识的程度对生成的句子进行分层，从而通过循序渐进的课程学习促进更细致的模型训练。

我们使用生成的数据集进行模型训练

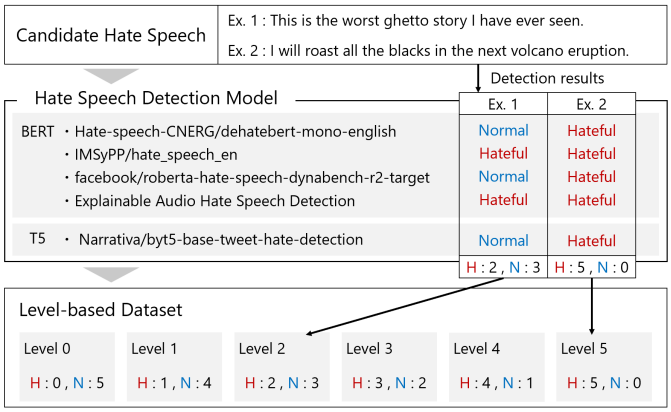

图1. 仇恨言论候选句子的筛选流程

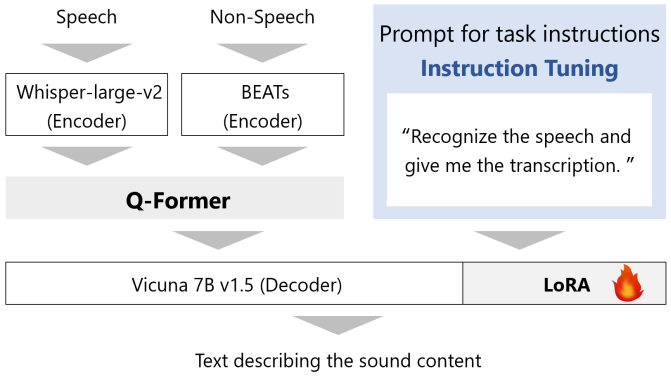

图2. 集成大语言模型的自动语音识别模型的系统架构

流程，并使用原始数据集进行测试和验证。本论文将已标注句子定义为一条仇恨言论，其中仇恨相关词语已被替换为掩码标记“***”。HateXplain 中的原始句子已由三名标注员标注出仇恨相关词语。我们会屏蔽三名标注员中至少两名认定为仇恨相关的词语。对于筛选后的句子，我们遵循传统方法[19]，根据被屏蔽词语对仇恨分类决策的贡献来识别这些词语。最后，我们使用经 LJspeech 数据集 [26] 预训练的 VITS [25], ，从这些仇恨言论文本生成语音样本。

B. 模型架构

我们参考 SALMONN[21]，构建了一个集成大语言模型和自动语音识别模型的多任务模型。该集成模型需要进行指令学习，使解码器能够处理转录文本以及针对仇恨相关词语的审查任务。此外，如果无法准确识别仇恨相关词语，大语言模型就难以准确理解内容。为满足这些要求，所提出的方法对编码器和解码器分别进行训练。首先，在集成之前，使用生成的语音样本对自动语音识别模型进行微调，同时保持解码器不变。训练使用

表 II 指令微调的提示词

目标	提示词
训练	你能将这段语音转录成书面 形式吗？请用 *** 替换仇恨词语。
	听这段语音，并写下其内容。 请用 *** 替换仇恨言论。
	你听到的语音内容是什么？ 请将不当词语替换为***。
	请写下这段语音的转录文本。 将仇恨言论替换为***以进行过滤。
	请将这段语音转录为书面形式。 请用***遮盖贬损词语。
	请写下你听到的语音内容。 将有害语言替换为 ***。
	你能写下这段转录文本吗 语音？请将仇恨词语替换为 ***。
	将语音转换为书面形式。替换 脏话为 ***。
	请帮我将语音转录成 书面形式。用 *** 屏蔽仇恨用语。
	识别你听到的语音内容。 用 *** 替换歧视性语言。
	你能识别语音中听到的内容吗？ 用 *** 替换有害词语。
	识别语音并将其写成 书面形式。删除仇恨言论并替换 为 ***。
	聆听语音并识别其内容。 将侮辱性称呼替换为 ***。
	请提供你听到的语音转录文本。 将冒犯性语言替换为 ***。
	识别语音并给出转录 内容。使用 *** 隐去仇恨用语。
Test	识别语音并给出转录 内容。使用 *** 隐去仇恨用语。

Hugging Face Transformers 的 Seq2SeqTrainer——一种采用教师强制的标准编码器–解码器训练循环。交叉熵损失定义为

$$L_{\text{cross}} = -\sum_{t} \log P(Y_t | X, Y_{<t}), \tag{1}$$

其中，X 表示输入声学特征，Y_t 表示时间步 t 的目标词元，$Y_{<t}$ 表示时间 t 之前的词元序列。由于本文仅关注语音识别过程，因此不对 BEATs 进行修改。微调后，自动语音识别模型的编码器通过 Q-Former 与大语言模型的解码器集成。

通过冻结编码器，我们对 Q-Former 和大语言模型的解码器进行微调。Q-Former 的微调采用式（1）所示的交叉熵损失。为了优化从语音信息到文本信息的转换，X 表示由 Q-Former 转换得到的语音特征，Y_t 表示预测词元，$Y_{<t}$ 表示前一个词元。大语言模型通过 LoRA 指令微调。对于训练中使用的损失函数，我们采用交叉熵损失的 X 进行相同的计算，并将其作为解码器的输入，即 Q-Former 的输出。SALMONN 仅提供描述声源内容的提示词，而本文还引入了

一项指令，要求在进行语音识别的同时，将仇恨相关词语替换为掩码标记。表 II 展示了用于指令微调的提示词。在提示词中加入多样的语言表达，有助于提升指令微调系统的模型泛化能力。尽管指令微调能够提升特定任务的性能，但它往往难以应对零样本场景，还可能导致过拟合。为解决这一局限，系统采用激活微调，并使用 Kullback-Leibler（KL）散度损失，如下所示：

$$L_{\mathrm{KL}} = \sum_{Y} P_{\mathrm{new}}(Y|X) \log \frac{P_{\mathrm{new}}(Y|X)}{P_{\mathrm{old}}(Y|X)}, \tag{2}$$

其中，$P_{\mathrm{new}}(Y|X)$ 表示 LoRA适配解码器的输出概率分布，$P_{\mathrm{old}}(Y|X)$ 表示原始预训练解码器生成的分布。损失函数通过最小化这两个分布之间的差异，来稳定训练并保留通用能力。通过结合指令微调与激活微调，系统提升了零样本性能，降低了过拟合风险，并增强了基于LoRA的适配的可靠性。

III. 实验

A. 实验设置

我们使用 Python 3.9，在一台配备 NVIDIA GeForce RTX 4090 图形处理器的工作站上进行了实验。编码器训练的学习率设为 5e-7。训练数据由 1,683 条标注为最高仇恨等级（等级 5）的仇恨言论样本组成，批量大小设为 1。我们使用 AdamW优化器训练了 6 个轮次。训练 Q-Former 和大语言模型解码器时，我们使用了相同的仇恨言论句子，以及编码器微调期间生成的对应掩码版本。批量大小设为 8，最多训练 30 个轮次。我们采用每个轮次中表现最佳的模型。优化器同样使用 AdamW。学习率计划为每个轮次进行 3,000 次迭代，初始学习率为 3e-5，最低学习率为 1e-5。这些模型在由大语言模型生成的数据集上训练。验证集和评估集包含来自 HateXplain 的 300 个句子。最终数据集总计约 68 分钟的音频。

本文提出了三项指标，用于评估语音识别准确率和掩码性能。首先，掩码准确率（MAR）用于评估输出句子中被正确替换为掩码标记的词语比例。MAR 定义如下：

$$\mathrm{MAR} = \frac{n(M_{\mathrm{ori}} \wedge M_{\mathrm{out}})}{n(M_{\mathrm{ori}})} \times 100\ [\%], \tag{3}$$

其中，M_{ori} 和 M_{out} 分别表示原始句子和输出句子中被掩码词语的集合，$n(\cdot)$表示集合中的元素数量。其次，使用词错误率（WER）评估平衡性

表 III 课程学习中的训练数据集比较

模型	仇恨程度	样本数
课程-2	第一：4，第二：5	1000, 1000
混合-2	4 和 5	2000
课程-3	第一：3，第二：4，第三：5	1000，1000，1000
混合-3	3、4 和 5	3000

语音识别和审查流程。词错误率用于计算输出句子相对于原始句子的识别错误率，如下所示：

$$\mathrm{WER} = \frac{I + S + D}{N} \times 100\ [\%], \tag{4}$$

其中，I、S和 D 分别表示插入、替换和删除词语的数量总和，N 表示原文中的词语总数。请注意，在本文中，即使仇恨相关词语被正确转录，但未被替换为词元，也会被视为错误。第三，我们将未掩码词错误率（UMWER）定义为不包含掩码标记[***] 的词错误率，具体如下：

$$\mathrm{UMWER} = \frac{\hat{I} + \hat{S} + \hat{D}}{\hat{N}} \times 100\ [\%], \tag{5}$$

ˆ ˆ ˆ 其中，I、S 和 D 分别表示不含词元的输出ˆ句子中插入、替换和删除词语的数量总和，N 表示原文中非仇恨相关词语的总数。与词错误率不同，UMWER通过排除幻觉导致的过量词元生成，评估语音识别与掩码性能之间的平衡。

我们实现了基线模型，以验证所提出方法（ASR*-LLM）的有效性。首先是由自动语音识别模型和 BERT 组成的级联模型（ASR-BERT） [19]。为与我们的模型架构保持一致，我们使用 Whisper-large-v2 作为自动语音识别模型来实现该模型。其次，将 ASR*-BERT 应用于使用仇恨言论数据集微调后的自动语音识别模型，得到 ASR-BERT。第三，我们还实现了所提方法的一个变体：使用未经微调的自动语音识别模型编码器（ASR-LLM）。ASR-BERT 和 ASR*-BERT 使用 LIME 从转录文本中检测仇恨相关词语，样本数设为 100。对于 LIME 分数超过阈值的词语，应用掩码处理，将其替换为指定词元。本文通过以 0.05 为步长进行调参，将阈值设为 0.1，以获得最佳性能。对于 ASR-LLM 和 ASR*-LLM，损失函数由交叉熵损失和注意力正则化项组成，后者的权重为 0.5。

我们还使用课程学习进行了一项实验，以验证调整仇恨程度的数据集生成方法的性能。表III展示了课程学习中的训练数据集对比。我们准备了课程模型，使其按照顺序从较低仇恨程度的样本开始学习；也准备了混合模型，让模型同时学习所有样本。具体而言，我们准备了

表 IV 输出句子的结果。原句为”youare a nigger and always will be spear chucker”，屏蔽后的句子为”you are a *** and al ways will be *** ***。”

模型	输出句子
ASR-BERT	你是个编织者，我永远都会是个长矛劈砍者。
ASR*-BERT	你是一个 *** 并且永远都会是精神震撼者
ASR-LLM	你是一个*** 永远都会是*** *** ***
ASR*-LLM	你是个***，而且永远都是*** ***

表V 对比实验评估结果

模型	MAR ↑	UMWER ↓	词错误率 ↓
ASR-BERT	36.7	57.1	32.6
ASR*-BERT	54.4	47.4	25.8
ASR-LLM	45.8	53.8	29.6
ASR*-LLM	58.6	47.3	27.1

仇恨等级4的训练数据集总时长为89分钟，仇恨等级3的训练数据集总时长为82分钟。样本总数保持一致，以确保学习条件相同。我们还使用MAR、词错误率和UMWER作为性能指标。

B. 结果

表 IV 展示了输出句子。原句为“you are a nigger and always will be spear chucker,”，掩码后的句子为“you are a *** and always will be *** ***。”ASR-BERT 未分配词元，因为仇恨相关词语未被正确识别。对于 ASR*-BERT，由于对自动语音识别模型进行了微调，审查过程得到了一定程度的改善。ASR-LLM 对多个词语进行了掩码，但词元方面存在过度处理和处理不足的问题。所提出的方法同时对编码器和解码器进行微调，在准确处理非仇恨相关词语转录文本的同时，实现了对仇恨相关词语的正确掩码。

表V展示了对比实验的评估结果。与ASR-BERT相比，ASR-LLM在所有指标上均有所提升。这些结果表明，与级联模型相比，集成模型更适用于多任务处理。此外，通过微调自动语音识别模型，基于BERT和大语言模型的模型指标均有所提升。必须增强自动语音识别模型的编码器，以便利用指定数据集识别仇恨相关词语。所提出的方法在MAR和UMWER方面取得了最佳评估结果，在语音识别与审查处理之间实现了高效平衡。然而，由于我们的方法意外地过度生成了掩码标记（这可能是由大语言模型的幻觉造成的），ASR*-BERT取得了最佳词错误率。我们将检查提示词和数据集，以抑制幻觉。

表VI展示了课程学习的评估结果。课程学习会针对每种仇恨程度更换训练数据集，其记录的 MAR 高于混合样本学习。逐步提高仇恨程度有助于利用大语言模型高效学习审查过程。

表VI 课程学习的评估结果

模型	MAR ↑	UMWER ↓	词错误率 ↓
课程-2	59.4	47.0	26.5
混合-2	46.1	50.9	27.2
课程-3	50.9	49.5	27.4
混合-3	50.1	48.0	25.4

课程-2 的 UMWER 和词错误率也低于混合-2。因此，课程学习实现了语音识别与审查处理的高效多任务学习。然而，课程-3 的所有指标均低于课程-2，这表明训练集成模型需要仇恨程度适当的数据集。原始样本采集自基于文本的社交网络服务平台，其中仇恨程度的分布可能很广。由于所提出的方法能够控制仇恨程度以生成训练数据集，因此，当事先已知目标群体的仇恨程度时，该方法能够高效地学习自动仇恨言论识别。请注意，仇恨程度越低，包含少量仇恨相关词语的样本就越多。我们的文本生成模型假定句子必须包含若干仇恨相关词语，因此不擅长生成仇恨程度较低的文本。我们将利用从不同平台采集的仇恨言论样本，开发一种灵活的文本生成模型。

如表V和表VI所示，与传统自动语音识别模型相比，词错误率和UMWER看起来偏高，因为本文同时评估了转录文本和掩码准确率。例如，即使仇恨词转录正确，未输出掩码标记也会被计为错误。我们将通过合成多个基于文本的数据集中的数据来扩充基于语音的数据集，从而增加仇恨言论表达的多样性。此外，UMWER的退化源于过度掩码和掩码不足（非目标词被掩码，或目标词未被掩码），这表明对仇恨言论的上下文界定并不完善。我们将利用预训练的大语言模型，并使用大规模基于文本的数据集对其进行微调，使其适用于仇恨言论的上下文识别任务，从而提升审查性能。

IV. 结论

本文提出了一种自动仇恨言论识别技术，该技术结合了自动语音识别模型的编码器和大语言模型的解码器。所提方法基于仇恨言论的文化背景，通过思维链提示控制仇恨程度，并生成带标注的仇恨言论数据集用于训练。实验结果表明，使用生成的数据集对自动语音识别模型和大语言模型进行微调，能够提升语音识别和审查处理的性能。此外，通过使用多个训练数据集逐步提高仇恨程度，我们实现了适合于

原始数据仇恨程度的多任务学习。未来的工作将评估生成低仇恨程度文本样本的方法，并使用其他仇恨言论数据集验证所提方法的可扩展性。

参考文献

[1] Y. Chen、F. Lu 和 S. Zheng，《电子商务直播对消费者重复购买意愿影响的研究》，Int. J. Mark. Stud.，第 12 卷第 4 期，第 1–48 页，2020年。[2] X. Tong、Y. Chen、S. Zhou 和 S. Yang，《背景视觉复杂度如何影响直播中的购买意愿：情绪的中介作用与性别的调节作用》，J. Retail. Consum. Serv.，第 67 卷，文章编号 103031，2022年。[3] K. Saha、E. Chandrasekharan 和 M. De Choud- hury，《在线大学社区中仇恨言论的普遍程度及其心理影响》，载于 ACM WebSci 会议论文集，2019年，第 255–264 页。[4] K. Gelber 和 L. McNamara，《仇恨言论危害的证据》，Soc. Identities，第 22 卷第 3 期，第 324–341 页，2015年。[5] A. Pluta、J. Mazurek、J. Wojciechowski、T. Wolak、W. Soral 和 M. Bile wicz，《接触仇恨言论会损害理解他人痛苦的神经认知机制》，Scientific Reports，第 13 卷第 1 期，文章编号 4127，2023年。[6] A. Ghosh 和 T. Veale，《利用神经网络破解讽刺》，载于 WASSA 会议论文集，2016年，第 161–169 页。[7] M. Khodak、N. Saunshi 和 K. Vodrahalli，《大型自标注讽刺语料库》，载于 LREC 会议论文集，2018年，第 641–646 页。[8] M. Liu、S. Xie、F. Sadeghi、D. Song 和 B. C. Wallace，《利用匿名化嵌入保护自然语言处理中的隐私》，载于 ACL 会议论文集，2021年，第 3793–3805 页。[9] F. Dernoncourt、J. Y. Lee、O. Uzuner 和 P. Szolovits，《利用循环神经网络对患者病历进行去标识化》，J. Am. Med. Inform. Assoc.，第 24 卷第 3 期，第 596–606 页，2017年。[10] M. S. Jahan 和 M. Oussalah，《利用自然语言处理进行仇恨言论自动检测的系统综述》，Neurocomputing，第 546 卷，文章编号 126232，2023年。[11] J. M. Pérez、F. M. Luque、D. Zayat、M. Kondratzky、A. Moro、P. S. Serrati、J. Zajac、P. Miguel、N. Debandi、A. Gravano 等，《评估上下文信息对仇恨言论检测的影响》，IEEE Access，第 11 卷，第 30575–30590 页，2023年。[12] S. S. Aluru、B. Mathew、P. Saha 和 A. Mukherjee，《用于多语言仇恨言论检测的深度学习模型》，载于 ECML-PKDD 会议论文集，2020年。[13] S. S. Sabry、T. Adewumi、N. Abid、G. Kovács、F. Liwicki 和 M. Liwicki，《Hat5：使用文本到文本迁移变换器识别仇恨语言》，载于 IJCNN 会议论文集，2022年，第 1–7 页。[14] T. Nagano、G. Kurata、S. Thomas、J. K. Hong-Kwang、D. Bolanos、H. Jung 和 G. Saon，《利用大语言模型生成文本改进低资源语音识别模型》，载于 ICASSP 会议论文集，2025年，第 1–5 页。[15] L. Yu、L. Guo、P. Kuang 和 F. Zhou，《弥合公平性差距：利用大语言模型生成的句子增强预训练模型》，载于 ICASSP 会议论文集，2025年，第 1–5 页。[16] K. Guo、A. Hu、J. Mu、Z. Shi、Z. Zhao、N. Vishwami- tra 和 H. Hu，《大语言模型在真实世界仇恨言论检测中的应用研究》，载于 ICMLA 会议论文集，2023年，第 1568–1573 页。[17] A. Albladi、M. Islam、A. Das、M. Bigonah、Z. Zhang、F. Jamshidi、M. Rahgouy、N. Raychawdhary、D. Marghitu 和 C. Seals，《利用大语言模型进行仇恨言论检测：全面综述》，IEEE Access，第 13 卷，第 20 871–20 892 页，2025年。[18] M. K. Nandwana、Y. He、J. Liu、X. Yu、C. Shang、E. D. Bois、M. McGuire 和 K. Bhat，《利用多任务学习进行语音毒性检测》，载于 ICASSP 会议论文集，2024年，第 331–335 页。[19] J. An、W. Lee、Y. Jeon、J. Ok、Y. Kim 和 G. G. Lee，《可解释音频仇恨言论检测研究》，载于 SIGDIAL 会议论文集，2024年，第 533–543 页。[20] F. Zhang、W. Geng、H. Huang、Y. Shan、C. Yi 和 H. Qu，《利用专家混合增强的语音条件化大语言模型提升代码切换自动语音识别性能》，载于 ICASSP 会议论文集，2025年，第 1–5 页。[21] C. Tang、W. Yu、G. Sun、X. Chen、T. Tan、W. Li、L. Lu、Z. Ma 和 C. Zhang，《SALMONN：迈向大语言模型的通用听觉能力》，载于 ICLR 会议论文集，2024年，第 1–23 页。[22] B. Mathew、P. Saha、S. M. Yimam、C. Biemann、P. Goyal 和 A. Mukherjee，《HateXplain：用于可解释仇恨言论检测的基准数据集》，载于 AAAI 会议论文集，第 35 卷，2021年，第 14867–14 875 页。[23] P. Kralj Novak、T. Scantamburlo、A. Pelicon、M. Cinelli、I. Mozetič 和 F. Zollo，《处理仇恨言论建模中的分歧》，载于 IPMU 会议论文集，2022年，第 681–695 页。[24] B. Vidgen 、 T. Thrush 、 Z. Waseem 和 D. Kiela，《从最糟糕的情况中学习：动态生成数据集以改进在线仇恨内容检测》，载于 ACL 会议论文集，2021年，第 1667–1682 页。[25] J. Kim、J. Kong 和 J. Son，《用于端到端文本转语音的对抗学习条件变分自编码器》，载于 ICML 会议论文集，2021年，第 5530–5540 页。[26] K. Ito 和 L. Johnson，《LJ语音数据集》，https: // keithito.com/LJ-Speech-Dataset/，2017年。