音频描述中的多任务学习：句子嵌入回归损失充当正则化项

Etienne Labb´e图卢兹大学 *IRIT*´图卢兹CNRS，UT3，法国图卢兹 etienne.labbe@irit.fr

Julien Pinquier图卢兹大学*IRIT*´图卢兹CNRS，UT3，法国图卢兹 julien.pinquier@irit.fr

Thomas Pellegrini图卢兹大学*IRIT*´图卢兹CNRS，UT3，法国图卢兹 thomas.pellegrini@irit.fr

arXiv:2305.01482v1 [cs.SD] 2 May 2023

摘要——**本文提出研究在自动音频描述任务中，采用包含句子嵌入回归损失项训练的模型的性能。该任务旨在构建能够用自然语言写成的单个句子描述音频内容的系统。大多数系统采用标准交叉熵损失进行训练，但这种损失不会考虑句子的语义相似度。我们发现，加入句子嵌入损失项可以减少过拟合；在 AudioCaps 语料库上的首个实验设置中，SPIDEr 得分也从 0.397 提高到 0.418。提高权重衰减值后，我们发现模型的表现更接近当前最先进的方法，SPIDEr 得分最高达到 0.444，而对比得分为 0.475。此外，该模型使用的可训练参数少了八倍。在这种训练设置下，句子嵌入损失不再影响模型性能。**

索引词——**声音事件描述、多任务学习、音频语言任务、过拟合、句子嵌入回归损失、语义损失**

I. 引言

近年来，文本处理、生成和理解领域的机器学习系统取得了显著进展，使自然语言成为人与机器之间的通用接口。自由文本所包含的信息远多于预先定义的一组类别，这有助于提升机器对我们所处世界的理解。在音频领域，大多数任务都集中于声音事件的分类和定位。基于这一思路，自动音频描述（AAC）任务于 2017 [1]提出，旨在创建能够生成自然语言句子来描述音频文件的系统。音频可能包含各种不同长度的声音事件（人声、自然声、家庭环境声、城市环境声、音乐、音效等），这些声音是在不同场景中由不同设备录制的。描述可以涵盖音频中的各种细节，包括声音之间的时间或空间关系（先后发生、位于背景中等），也可以包含不同的特征描述（音调高、短促、重复等）。由于描述由人撰写，我们需要考虑多种情况：用不同词语描述相似的声音（鸟儿在鸣叫/啁啾/歌唱/啼叫）、不同的句子结构（一扇需要上油的门/一扇铰链吱呀作响的门）、主观性（一个男人用外语说话）、高层次描述（一个粗俗的男人在说话/听不懂的交谈）以及含糊性（有人在说话而非一个男人通过带有混响的麦克风发表演讲）。

在 AAC 中，大多数方法使用以标准交叉熵（CE）损失训练的深度学习模型。然而，这种损失往往会生成重复且笼统的内容 [2]，也没有考虑同义词、不同的句子结构或语义相似度。一些研究引入了另一种准则——自我批判序列训练 [3]（SCST），这是一种用于强化学习的技术，可直接依据某项指标对模型进行微调，而不是依据损失进行微调。该技术通过采样下一个词来生成新句子。如果这个句子的得分高于原句，模型便会得到奖励，并提高新句子对应的输出概率。然而，这种技术会导致句子退化 [4],，出现重复的 n 元语法，且语法不正确。

受交叉熵和 SCST 局限性的启发，我们尝试在 [5] 中加入句子嵌入回归（SER）损失，以改进模型。本文首先介绍基线系统，然后说明如何加入 SER 损失。我们介绍相关工作并进行比较，随后详细说明超参数。最后，我们展示结果并讨论差异。

II. 基线系统描述

我们采用自动音频描述系统中广泛使用的编码器—解码器架构，并使用在 AudioSet 上预训练的编码器 [6]来提取声音事件的强表征。具体而言，我们使用预训练音频神经网络（PANN）研究中的 CNN14_DecisionLevel_Att 音频编码器 [7],，其预训练权重可在 Zenodo[1]上获取。与其他可用的 PANN 架构相比，该架构在音频描述数据集部分的声音事件分类任务中取得了最佳结果。我们发现，冻结权重不会降低性能，同时还能显著加快训练过程。该编码器可为时长为十秒的录音生成维度为31×2048 的嵌入序列。在此基础上，我们添加一个投影层，得到 256 维嵌入，以匹配解码器的输入维度 d_{model}。

解码器是一个标准的 Transformer 解码器 [8]，包含 6 层，每层有 4 个注意力头，全局嵌入维度d_{model} 设为 256，全局 dropout 概率为 0.2。我们还在解码器中使用了 GELU [9]激活层。

[1]https://zenodo.org/record/3987831

解码器通过教师强制进行训练，即 将真实值中的前序参考词元提供给模型，以预测下一个词元。基线准则是对整个序列的输出概率与参考词元类别之间计算标准交叉熵损失。

在推理过程中，我们使用束宽设为 2 的束搜索算法，因为更大的值并不会带来改进。为提升性能和整体描述质量，我们对句子生成过程施加了限制：将预测长度限定为至少 3 个词元、至多 30 个词元，并禁止模型重复生成同一词元，但 [10]package 中预先定义的停用词词元除外。这里的限制减少了无效句子和重复内容，并使模型性能略有提升。

III. 添加句子嵌入回归损失

A. Sentence-BERT 模型

Sentence-BERT [11] (SBERT) 模型是一种基于 Transformer 的模型，它将 BERT [12] 模型与池化层和投影层结合起来，为给定句子生成一个由 768 个值组成的固定大小的嵌入。

现有的 SBERT 模型使用两个文本数据库进行训练：斯坦福自然语言推理（SNLI） [13] 和多类型自然语言推理（MultiNLI） [14]。这些数据集包含成对的句子，并标注为矛盾、蕴含或中性标签。为学习句子的语义，两个 SBERT 嵌入被输入分类层，由该层预测句子对的标签。

B. SER损失

为了利用 SBERT 模型改进我们的模型，我们需要使用相同的词元单位，以保持序列长度一致。BERT 使用 WordPiece词元 [15] 而非单词；在我们的实验中，这些词元构成了一个包含 30522 个不同单位的词汇表。

图 1概述了整个流程和所用层。在训练阶段，我们使用音频特征和真实值中的前序词元来生成下一个词元嵌入，记作 $\hat{e}_t$。这些嵌入会用于模型的两个不同部分。首先，通过分类器将它们投影为 logits，以计算标准交叉熵损失 $\mathcal{L}_t$。此外，词元嵌入还会从 256 维投影到 768 维，以匹配 SBERT 嵌入的输入形状。得到的嵌入 $\hat{e}_s$ 与真实值嵌入一起作为 SER 损失组件 $\mathcal{L}_s$的输入。为了使用 SBERT 模型训练我们的模型，我们需要移除

SBERT 将词元 ID 映射为嵌入向量（图中标为“嵌入”），因为这一层不可微。在推理时，仅使用分类器分支。

我们尝试了几种用于 $\mathcal{L}_s$ 损失的回归准则：余弦嵌入损失、L1 损失、均方误差损失和平滑 L1 损失。我们得到的最佳结果是平滑 L1 损失 [16],，这是一种结合了均方误差和 L1 损失的回归函数，如公式 (1) 所示：

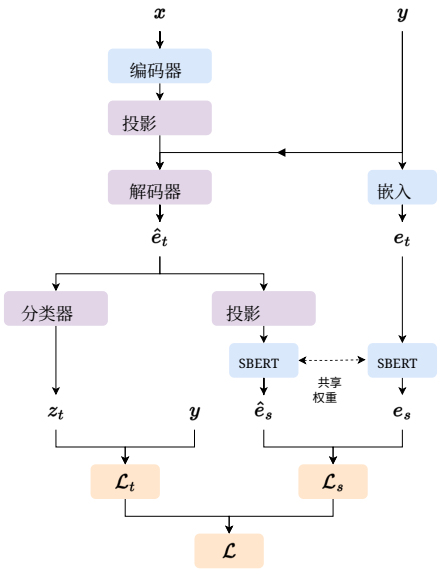


图 1. 我们提出的训练方法概览。蓝色方框表示预训练冻结层，紫色方框表示可训练层，橙色方框表示函数。SBERT 模块包含 SBERT 模型，但移除了其首个嵌入层。

$$\mathcal{L}_s(\hat{e}_s, e_s) = \begin{cases} (\hat{e}_s - e_s)^2 \cdot \frac{1}{2\beta} & \text{if } |\hat{e}_s - e_s| < \beta \\ |\hat{e}_s - e_s| - \frac{\beta}{2} & \text{otherwise} \end{cases} \quad (1)$$

β 超参数用于控制应使用均方误差还是 L1 损失。我们保留了标准交叉熵作为第一个组成部分 $\mathcal{L}_t$，以帮助模型生成语法有效的句子。最终损失由公式 (2) 给出，是 $\mathcal{L}_t$ 和 $\mathcal{L}_s$的加和，并由一个系数 λ加权：

$$\mathcal{L} = \mathcal{L}_t + \lambda \cdot \mathcal{L}_s \quad (2)$$

IV. 相关工作

目前，AudioCaps [17] 上最先进的方法是名为音频描述 Transformer（ACT） [18], 的完整 Transformer 架构，该架构在 AudioSet 上进行预训练，类似于 PANN 模型。该系统在训练期间对音频波形和频谱图进行 mixup [19] ，并拼接相应的字幕，以提升泛化能力。在推理时，系统使用高斯噪声和 SpecAugment [20] 生成同一样本的多个变体，并将其输入模型。随后，对同一示例的中间表示取平均，以生成更好的句子。

在 [21], 中，作者提出使用一个预训练的 Transformer 解码器 BART [22] 来生成更好的句子。他们的第一个编码器（YAMNet）会预测 AudioSet 类别的名称，以改进音频表示。这些类别名称作为 BART 嵌入层的输入，并与 PANN 编码器的音频嵌入相加。解码器是预训练 BART Transformer 解码器部分。为简洁起见，我们在表中将他们的模型命名为 BYP，即 BART+YAMNet+PANNs。

文献 [5]提出了一种与我们方法相似的方案，但存在显著差异（音频编码器、优化器，以及用于优化和生成的超参数），因此提供了更强的基线。与他们不同，我们的模型只通过一个阶段进行训练（交叉熵+SER损失），而不是两个阶段（先使用交叉熵，再使用交叉熵+SER损失）。

V. 实验设置

A. 数据集

我们在 AudioCaps [23]数据集上训练和评估模型。该数据集是目前已知最大的人类生成字幕音频语言数据集。音频文件是来自 AudioSet [6] 的 10 秒片段，提取自 YouTube 视频。由于部分原始视频已被删除或无法访问，我们所用的数据集中，训练子集有 46230 个文件（原有 49838 个），验证子集有 464 个文件（原有 495 个），测试子集有 912 个文件（原有 975 个）。训练子集中的每段音频配有一条字幕，验证子集和测试子集中的每段音频则配有五条字幕。

B. 指标

我们仅关注字幕生成指标，因此决定不采用翻译指标（BLEU、ROUGE、METEOR），因为这些指标主要基于 n 元语法重叠度。CIDEr-D [24]计算候选项和参考标注中共有 n 元语法的 TF-IDF 分数之间的余弦相似度。SPICE [25]使用解析器和语法规则，从句子中提取语义命题，并计算表示这些命题的图边的 F1 分数。SPIDEr [26]是 CIDEr-D 和 SPICE 的平均值，主要用于对自动音频描述系统进行排序。由于我们还在研究句子相似度损失，因此决定加入来自 [27]的三种基于模型的指标：SBERT、FluErr 和 FENSE。SBERT 指标对应于使用 SBERT 模型提取的句子嵌入的余弦相似度。FluErr 是由一个模型检测出的流畅度错误率，该模型经过训练，能够识别字幕生成系统常见的错误，例如不完整句子、重复事件、重复副词、缺失连词和缺失动词。FENSE 指标是每个句子的 SBERT 分数；如果检测到 FluErr 指标中的错误，则该分数除以 10。最后一个指标“#Words”表示整个子集中候选项所用的唯一词汇数量。

C. 超参数

超参数对深度学习系统的训练至关重要。我们发现，使用不同的超参数组合进行训练时，模型可能会获得截然不同的分数。我们对超参数进行了优化，以最大化 AudioCaps 验证子集上的 FENSE 分数。我们总共训练模型 100 个训练轮次 K ，并在单个图形处理器上以 512 个样本的批量大小进行训练。首次实验中，我们使用 AdamW 优化器 [28] ，权重衰减（wd）为 10^{-6} ；第二种设置中，将其设为 2 以限制过拟合。网络的偏置权重不应用权重衰减。我们还发现，使用标准 Adam [29] 优化器时，网络在较大的 wd 下无法收敛。训练开始时，初始学习率 lr_0 设为 $5 \cdot 10^{-4}$，而 β_1 和β_2 的值分别设为 0.9 和 0.999。我们采用余弦调度器衰减，并在每个训练轮次结束时更新 k ，规则如下：$\mathrm{lr}_k = \frac{1}{2}\left(1 + \cos\left(\frac{k\pi}{K}\right)\right)\mathrm{lr}_0$。字幕均转换为小写，并删除所有标点符号。为稳定训练，我们将梯度的 L2范数裁剪至 10，并在交叉熵损失部分加入设为 0.1 的标签平滑。在各训练轮次中选择最佳模型时，我们采用 FENSE 分数最高的模型，而不是依据验证子集上的交叉熵损失。使用 FENSE 可以选取比依据验证损失函数所选结果更靠后的训练轮次，从而获得更好的结果。

对于句子嵌入回归方法，我们使用了 paraphrase-TinyBERT-L6-v2 模型[2]，因为 SBERT 和 FENSE 这两个指标使用的正是该模型。我们也尝试了更大的模型（all-mpnet-base-v2 和 all-mpnet-base-v1），但并未带来任何改进。所选模型包含 6700 万个参数，其权重在训练期间被冻结。我们在 $\mathcal{L}_s$ 函数中将 β 设为 1。我们尝试了 λ 的多个取值（1、10、100、1000 和 10000），发现 100 是句子嵌入回归的最佳参数。较高的取值会降低性能，而较低的取值与基线相比，对多任务学习没有影响。

VI. 结果

我们在表 I 中报告了基线方法（使用词分词器）的分数、使用 SBERT 分词器的方法的分数（基线+SBERT 词元），以及使用 SER损失方法的分数（基线+SBERT 词元+SER损失）。所有分数均为 5 个随机种子下的平均值。我们还列出了另一种 SER 方法 CNN10-trans 的分数、当前最先进方法（Multi-TTA 和 BYP）的分数，以及交叉引用的分数。对于交叉引用，我们从每个音频文件的五条字幕中排除一条，并将其用作候选句子，其余四条作为真实参考文本。重复此过程五次，以计算平均人工一致性分数，我们将其称为“交叉引用”。

A. 讨论

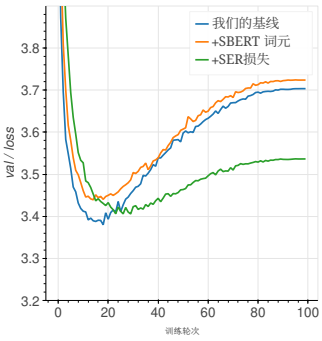
使用较小的权重衰减值时，SER损失使 FENSE 和 SPIDEr 得分略有提升，分别从 0.595 增至 0.607、从 0.397 增至 0.418。此外，当我们引入较大的权重衰减值以防止过拟合时，可以看到基线的 FENSE 和 SPIDEr 得分显著提升，分别从 0.595 增至 0.619、从 0.397 增至 0.445。尽管 SER损失也受益于较大的权重衰减值，但其得分与同样采用该权重衰减值的新基线非常接近，因此 SER损失带来的正则化效果似乎不再显著。

从图2中可以看出，较小的权重衰减配合 SER 损失，降低了验证学习曲线的上升幅度，

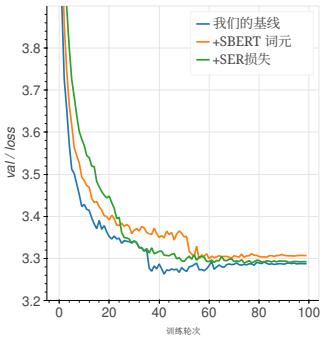
[2]https://www.sbert.net/docs/pretrained_models.html

表 I AudioCaps 测试子集上的 AAC 结果**测试** 字幕生成指标。箭头 ↑ 表示该列数值越高越好，而 ↓ 表示该列数值越低越好。Optim. 和 WD. 分别代表优化器和权重衰减。

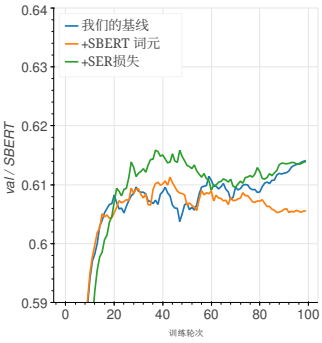
方法	优化器	Wd.	CIDEr-D ↑	SPICE ↑	SPIDEr ↑	FENSE ↑	SBERT ↑	FluErr ↓	#词数 ↑	可训练参数	冻结参数
交叉引用	N/A	N/A	.901	.217	.559	.680	.682	.005	952.2	0	0
Multi-TTA [17]	AdamW	10^{-6}	**.769**	**.181**	**.475**	N/A	N/A	N/A	N/A	108M	0
BYP [21]		N/A	.753	.176	.465					408M	
CNN10-trans [5]	Adam	N/A	.573	.158	.365	N/A	.545	N/A	N/A	14M	0
+SER 余弦损失			.573	.166	.370		.555				N/A
我们的基线	AdamW	10^{-6}	.628	.165	.397	.595	.601	.034	**485.8**	12.4M	79.7M
+SBERT 词元			.659	.167	.413	.602	.607	.034	445.0	25.7M	79.7M
+SER损失			.665	.170	.418	.607	.614	.027	465.6	25.9M	146.7M
我们的基线	AdamW	2	.712	**.176**	.444	.619	**.621**	.004	387.2	12.4M	79.7M
+SBERT 词元			**.715**	.175	**.445**	**.620**	**.621**	**.002**	390.2	25.7M	79.7M
+SER损失			**.715**	.172	.443	.619	.620	**.002**	348.8	25.9M	146.7M



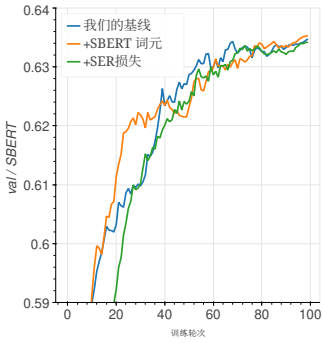
权重衰减设为 10^{-6}时的损失。



权重衰减设为 2 时的损失。



权重衰减设为 10^{-6}时的 SBERT。



权重衰减设为 2 时的 SBERT。

图2. 验证集上各训练轮次的交叉熵损失和 SBERT 余弦相似度。

这意味着它限制了模型的过拟合，也解释了表格中获得的提升。图中的验证损失和 SBERT 余弦相似度表明，较大的权重衰减所带来的正则化对模型非常有效。

我们的基线模型与我们的基线模型+SBERT 词元之间，可训练参数数量从 12.M 增至 25.7M，这是因为词汇表从 4724 扩大到 30522，导致解码器部分的分类器和输入嵌入层中的参数数量大幅增加。根据 SPIDEr 和 FENSE，采用 SBERT 词元和较大权重衰减值的方法已成为我们目前表现最好的模型，尽管采用相同衰减值的其他方法紧随其后。我们的方法也与当前最先进的方法非常接近；后者的得分为 0.475，而我们最佳的 SPIDEr 得分为 0.445 和 0.444。尽管如此，我们的基线模型和基线模型+SBERT 词元的可训练参数分别比后者少八倍和四倍。

B. 定性分析

表II和表III展示了我们的模型在不同训练流程下生成的句子示例。采用较小权重衰减值的基线系统似乎尝试使用更多同义词，但更常无法给出良好的描述，如表II所示。采用较大权重衰减值时，系统使用的词更少，句式也更笼统，尽管准确度更高。在这种情况下，我们还注意到，词数在测试子集中从采用较小和较大权重衰减值的基线系统的平均485.8词降至387.2词。加入SER损失后，词数进一步显著减少，平均仅为348.8词。

表II 使用权重衰减值 10^{-6} 的不同方法对 AudioCaps 测试文件（ID：“ARFFw0e_jig”）生成的字幕。

候选项	方法	SPIDEr	FENSE
一个人打嗝	我们的基线	.070	.208
一个人大声打嗝	SBERT 词元	.135	.291
一个人连续大声打嗝几次	SER损失	**.252**	**.398**
参考标注			
大声打嗝并尖叫			
响亮的打嗝声反复出现			
一声响亮且失真的嗝声，随后是一连串打嗝声			
几声失真的嗝声，随后是未经失真处理的打嗝声			
一连串失真的打嗝声，随后是未经失真处理的打嗝声			

VII. 结论

在本研究中，我们探讨了加入句子嵌入回归损失组件，以提升自动音频描述系统。我们通过优化超参数、对生成过程进行条件化，并使用更强的预训练编码器，为基线寻找最优配置。我们发现，SER 损失组件似乎

表 III 不同权重衰减下基线模型对一个 AudioCaps 测试文件（ID：“JZloTOdIY_c”）生成的字幕。

候选项	wd	SPIDEr	FENSE
一匹马嘶鸣并沉重地喘气	10^{-6}	.284	**.658**
一匹马在小跑	2	**.337**	.452
参考标注			
马低吼着，蹄声嗒嗒			
一匹马嘶鸣，随后小跑并喷鼻息			
马在草地上小跑，同时嘶鸣并喷鼻息			
马嘶鸣，接着喷鼻息并在泥土地面上小跑			
马嘶鸣并在地面上跺蹄			

能够减少自动音频描述系统的过拟合，但与较大的权重衰减值等更强正则化方法结合时，不再带来额外提升。我们还注意到，尽管正则化方法改善了主要指标（SPIDEr、FENSE），这些指标并未考虑所用词语的多样性。未来的字幕生成指标可以纳入所用词语的多样性，也可以像 [30] 中那样，在学习过程中由模型直接考虑这一点。

参考标注

[1] K. Drossos、S. Adavanne 和 T. Virtanen，“基于循环神经网络的自动音频描述”，载于*2017*年*IEEE*音频与声学信号处理应用研讨会（*WASPAA*），2017年，第374–378页。[2] A. Holtzman、J. Buys、L. Du、M. Forbes 和 Y. Choi，“神经文本退化的奇特案例”，2019年。 [在线]。可用：https://arxiv.org/abs/ 1904 09751 。S. J. Rennie、E. Marcheret、Y. Mroueh、J. [3] Ross 和 V. Goel，“用于图像描述的自我批判序列训练”，2016年。 [在线]。可用：https://arxiv.org/abs/ 1612 00563 。X. Mei、Q. [4] Huang、X. Liu、G. Chen、J. Wu、Y. Wu、J. Zhao、S. Li、T. Ko、H. L. Tang、X. Shao、M. D. Plumbley 和 W. Wang，“一种基于编码器-解码器、结合迁移学习与强化学习的音频描述系统”，2021 [ ] 2108 02752 。在线。可用：[5] https://arxiv.org/abs/。R. Mahfuz、Y. Guo 和 E. Visser，“使用语义相似度指标改进音频描述”，2022年。 [在线]。可用：https: //arxiv.org/abs/ 2210 16470 。J. F. Gemmeke、D. P. W. Ellis、D. Freedman、A. [6] Jansen、W. Lawrence、R. C. Moore、M. Plakal 和 M. Ritter，“Audio Set：用于音频事件的本体和人工标注数据集”，载于*2017*年*IEEE*声学、语音与信号处理国际会议（*ICASSP*）。美国路易斯安那州新奥尔良：IEEE，2017年3月，第776–780页。 [在线]。可用：http://ieeexplore.ieee.org/document/7952261/ [7] Q. Kong、Y. Cao、T. Iqbal、Y. Wang、W. Wang 和 M. D. Plumbley，“PANNs：用于音频模式识别的大规模预训练音频神经网络”，*arXiv:1912.10211 [cs，eess]*，2020年8月，arXiv： 1912 10211 [ ] 1912 10211 。。在线。可用：[8] http://arxiv.org/abs/。A. Vaswani、N. Shazeer、N. Parmar、J. Uszkoreit、L. Jones、A. N. Gomez、L. Kaiser 和 I. Polosukhin，“Attention is all you need”，2017年。 [ ] 1706 03762 在线。可用：https://arxiv.org/abs/。D. Hendrycks 和 K. Gimpel，“高斯误差线性单元（gelus）”， 2016 [ ] 1606 08415 。在线。可用：[10] https://arxiv.org/abs/。S. Bird、E. Klein 和 E. Loper，使用*Python*进行自然语言处理：使用自然语言工具包分析文本。“O’Reilly Media, Inc.”，2009年。[11] N. Reimers 和 I. Gurevych，“Sentence-BERT：使用孪生BERT网络的句子嵌入”，2019年。 [在线]。可用：https: //arxiv.org/abs/ 1908 10084 。J. Devlin、M. Chang、K. Lee 和 K. Toutanova，“BERT：用于语言理解的深度双向Transformer预训练”，*CoRR*，第abs/1810.04805卷，2018年。 [在线]。可用：[12] http://arxiv.org/abs/ 1810.04805[13]S. R. Bowman、G. Angeli、C. Potts 和 C. D. Manning，“用于学习自然语言推理的大型标注语料库”，载于*2015*年实证方法会议论文集 自然语言处理。葡萄牙里斯本：计算语言学协会，2015年9月，第632–642页。 [在线]。可用：https://aclanthology.org/D 15 1075 - A. [14] Williams、N. Nangia 和 S. Bowman，“通过推理进行句子理解的广覆盖挑战语料库”，载于《2018年计算语言学协会北美分会会议：人类语言技术》第一卷（长篇论文）。美国路易斯安那州新奥尔良：计算语言学协会，2018年6月，第1112–1122页。 [ ] 18 1101 在线。可用：[15] https://aclanthology.org/N - Y. Wu、M. Schuster、Z. Chen、Q. V. Le、M. Norouzi、W. Macherey、M. Krikun、Y. Cao、Q. Gao、K. Macherey、J. Klingner、A. Shah、M. Johnson、X. Liu、L. Kaiser、S. Gouws、Y. Kato、T. Kudo、H. Kazawa、K. Stevens、G. Kurian、N. Patil、W. Wang、C. Young、J. Smith、J. Riesa、A. Rudnick、O. Vinyals、G. Corrado、M. Hughes 和 J. Dean，“谷歌神经机器翻译系统：弥合人工翻译与机器翻译之间的差距”，2016年。 [在线]。可用：https://arxiv.org/abs/ 1609 08144 。R. B. Girshick，“Fast R-CNN”，CoRR，第abs/1504.08083卷，2015年。 [16] [ ] 1504 08083 在线。可用：[17] http://arxiv.org/abs/。E. Kim、J. Kim、Y. Oh、K. Kim、M. Park、J. Sim、J. Lee 和 K. Lee，“使用MixGen和多级测试时增强改进音频语言学习”，2022年。 [在线]。可用：https://arxiv.org/abs/ 2210 17143 。X. Mei、X. Liu、Q. Huang、M. D. Plumbley 和 W. Wang，“音频描述 Transformer”，2021年。 [在线]。可用：https: //arxiv.org/abs/ 2107 09817 。H. [18] Zhang、M. Cisse、Y. N. Dauphin 和 D. Lopez-Paz，“mixup：超越经验风险最小化”，2018年。[20] D. S. Park、W. Chan、Y. Zhang、C.-C. Chiu、B. Zoph、E. D. Cubuk 和 Q. V. Le，“SpecAugment：一种简单的数据增强方法，用于自动语音识别”，载于《Interspeech 2019论文集》，2019年，第2613–2617页。[21] F. Gontier、R. Serizel 和 C. Cerisara，“通过使用AudioSet标签微调BART实现自动音频描述”，载于DCASE 2021——第6届声学场景与事件检测和分类研讨会，西班牙，虚拟会议，2021年11月。 [19] 2021 [ ] 03522488 在线。可用：[22] https://hal.inria.fr/hal- M. Lewis、Y. Liu、N. Goyal、M. Ghazvininejad、A. Mohamed、O. Levy、V. Stoyanov 和 L. Zettlemoyer，“BART：面向自然语言生成、翻译和理解的去噪序列到序列预训练”，2019年。 [在线]。可用：https: // arxiv.org/abs/ 1910 13461 。C. D. Kim、B. Kim、H. Lee 和 G. Kim，“AudioCaps：在开放环境中生成音频字幕”，载于《2019年计算语言学协会北美分会会议：人类语言技术》第一卷（长篇与短篇论文）。美国明尼苏达州明尼阿波利斯：计算语言学协会，2019年6月，第119–132页。 [在线]。可用：[23] https://aclanthology.org/N 19 1011 - R. Vedantam、C. L. Zitnick 和 D. Parikh，“CIDEr：基于共识的图像描述评估”，arXiv:1411.*5726 [* cs*]*, 2015年6月，arXiv: 1411 5726 [ ] 1411 5726 。。在线。可用：[25] http://arxiv.org/abs/。P. Anderson、B. Fernando、M. Johnson 和 S. Gould，“SPICE：语义命题图像描述评估”，arXiv: [24] *1607 08822* 。cs，7月，*[ ]*， 2016， 1607 08822 [ ] arXiv：。。在线。可用：http: //arxiv.org/abs/ 1607 08822 。S. Liu、Z. Zhu、N. Ye、S. Guadarrama 和 K. Murphy，“通过SPIDEr的策略梯度优化改进图像描述生成”，2017年IEEE国际计算机视觉会议（ICCV），第873–881页，2017年10月，arXiv：1612.00370。 [在线]。可用：http://arxiv.org/abs/ 1612 00370 。Z. Zhou、Z. Zhang、X. Xu、Z. Xie、M. Wu 和 [26] K. Q. Zhu，“ 能否使用图像描述指标评估音频描述？ ”，2022 年 1 月，编号：arXiv:2110.04684 arXiv:2110.04684 [cs，eess]。 [在线]。可用：http://arxiv.org/abs/ 2110 04684 。I. Loshchilov 和 F. Hutter，“ 解耦权重衰减正则化 ”， [27] 2017 [ ] 1711 05101 。在线。可用：[28] https://arxiv.org/abs/。D. P. Kingma 和 J. Ba，“Adam：一种随机优化方法”，arXiv:1412.*6980 [*cs*]*, 2017年1月，arXiv：1412.6980。 [29] [ ] 1412 6980 在线。可用：[30] http://arxiv.org/abs/。S. Welleck、I. Kulikov、S. Roller、E. Dinan、K. Cho 和 J. Weston，“采用非似然训练的神经文本生成”，2019年。 [在线]。可用：https://arxiv.org/abs/1908.04319