基于 Patchout 和文本引导的高效音频描述 Transformer

技术报告

Thodoris Kou zelis[1,2], *GrigorisBastas* [1,2], *Athanasios Katsamanis*[2], *Alexandros Potamianos*[1] ,

[1] 语言与语音处理研究所，雅典娜研究中心，希腊雅典，{theodoros.kouzelis, g.bastas, nkatsam}@athenarc.gr[2] 电子与计算机工程学院，雅典国立技术大学，希腊雅典，{potam}@central.ntua.gr

arXiv:2304.02916v1 [cs.SD] 6 Apr 2023

摘要

自动音频描述是一项多模态翻译任务，旨在为给定的音频片段生成文本描述。本文提出了一种完整的 Transformer 架构，采用*Patchout* ，如 [1], 所提出的方法，从而显著降低计算复杂度并避免过拟合。描述生成部分以预训练分类模型提取的文本 AudioSet 标签为条件；该模型经过微调，以最大化 AudioSet 标签与真实描述之间的语义相似度。为缓解自动音频描述的数据稀缺问题，我们引入了来自上游音频相关任务的迁移学习，并扩充了域内数据集。此外，我们提出了一种将 Mixup 数据增强应用于自动音频描述的方法。我们开展了消融研究，以探究*Patchout* 和文本引导对最终性能的贡献。结果表明，所提出的技术提升了系统性能，同时降低了计算复杂度。我们提出的方法获得了 2022 年 DCASE 挑战赛 Task6A 的评委奖。

索引词— 自动音频描述, Transformer 编码器-解码器, 文本条件化,预训练, Mixup

1. 引言

自动音频描述（自动音频标注）是一项多模态任务，旨在为给定的音频片段生成文本描述。为了生成有意义的描述，方法需要捕捉音频片段中存在的声音事件，并识别它们之间的时空关系。它在多种应用场景中具有巨大的实用潜力，例如帮助聋人或听力障碍者（DHH）。

自动音频标注面临的主要挑战之一是数据不足。为解决这一问题，近期许多方法采用 PANNs 和 VGGish 网络等预训练模型，从而提升最终性能 [2, 3]。Mei 等人 [4]使用预训练的 Transformer编码器，因为近期研究表明，Transformer 在音频分类任务中的表现优于 CNN [5]。Transformer编码器的一个局限是，自注意力机制的复杂度会随输入序列长度呈平方级增长，因此难以在相对较长的音频样本上进行训练，例如 Clotho数据集中最长达 30 秒的音频。为解决这些问题，我们提出了一种自动音频标注 Transformer 模型，采用 Patchout faSt Spectrogram Transformer (PaSST) [1]作为编码器。PaSST 编码器与 Mei 等人 [4] 提出的编码器之间的主要区别在于当前的切块提取方法包含：(1) 一个卷积层，用于从输入频谱图中提取特征图。(2) 解耦的时间和频率位置编码。(3) *Patchout*，即在训练期间丢弃 Transformer 输入序列中的部分内容，促使模型在序列不完整时仍能保持良好表现。

为解决自动音频标注的词语选择不确定性问题，许多方法通过预训练编码器来预测从真实描述中提取的关键词 [6] ，或通过输入文本来控制描述生成 [7, 8]。在本研究中，我们从预训练的PaSST模型中推断 AudioSet类别标签，并将其用作引导输入文本。为了获得与真实描述在语义上相似且能够发挥引导文本作用的文本，我们在从 Clotho 中提取的音频-标签对上对 PaSST 进行微调。

此外，我们提出了一种将 Mixup 增强方法 [9] 用于自动音频标注的方法。

本报告其余部分安排如下：第 2 节讨论我们提出的系统。第 3 节详细介绍我们的实现。第 4 节展示结果，第 5 节总结本报告。

2. 系统描述

我们提出的模型骨干架构基于传统的序列到序列结构，由PaSST编码器和Transformer解码器组成。编码器从输入中提取抽象嵌入序列，然后将该序列馈送至解码器，由解码器输出文本描述。我们提出了一种多模态条件化方案，其中馈送至编码器的输入序列同时包含音频和文本信息。输入文本由一个冻结的、经过微调的 PaSST 模型获取。

2.1. 分块提取

每个频谱图 $\mathcal{C}$ 通过以下流程转换为输入特征序列 $X_{in} \in \mathbb{R}^{L \times h}$： (1) 频谱图经过一个二维卷积层，该卷积层的核大小为 16、步长为 10，输出维度为 d，生成特征图 $X_m \in \mathbb{R}^{d \times F_m \times T_m}$。(2) 添加位置嵌入。按照 PaSST 的实现，我们向特征图中添加频率嵌入$e_p^F \in \mathbb{R}^{d \times F_m \times 1}$ 和时间嵌入 $e_p^T \in \mathbb{R}^{d \times 1 \times T_m}$ ，以注入位置信息。(3) 应用结构化 *Patchout*[1] ，即从提取的特征图 X_m 中随机删除若干频率 bins p_f和时间帧 p_t，从而得到

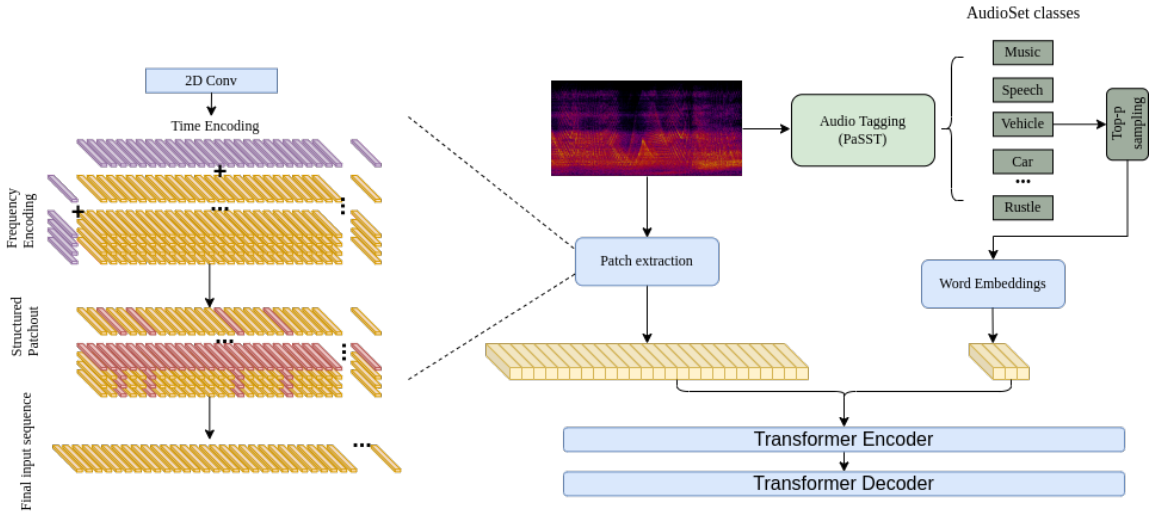

图 1：我们的系统概览。蓝色和绿色分别表示已学习和冻结的模块。为清晰起见，省略了解码器的输入。

在 $X_p \in \mathbb{R}^{d \times F_p \times T_p}$，其中 $F_p = F_m - p_f$，并且 $T_p = T_m - t_f$。（4）最后，将 X_p 展平为 $X_{in} \in \mathbb{R}^{L \times d}$，其中 $L = F_p \cdot T_p$。

2.2. 引导文本

PaSST 最近在音频分类任务中取得了最先进的性能 [1]。使用预训练的 PaSST 模型，我们从输入音频中推断AudioSet类别标签。标签中的每个词都使用可训练的嵌入映射到输入空间，并与提取的音频块拼接。与 [7], 类似，为了使我们的系统对 PaSST的预测错误更加鲁棒，我们从输出分布中采样每个标签。为了避免分布中“不可靠的尾部”，我们使用核采样（top- p）[10]。在推理过程中，我们选择概率最高的输出标签，而不是进行采样。

由于我们在模型中加入了词级信息，因此希望输入标签在语义上与真实描述文本相似，并发挥引导文本的作用。我们发现，PaSST 倾向于输出能够捕捉音频中概括性、高层次信息的标签，而不是出现频率较低、更加具体的标签。此类标签在语义上更可能与真实描述相似，甚至会在真实描述中原样出现。例如，某个音频片段的描述为“不远处，一群人正在交谈，但听不清他们在说什么。”，但它被分类为 语音，而实际上，AudioSet 类别 交谈声在语义上会更加准确。基于这一观察，我们对 PaSST 进行微调，使其预测与对应真实描述的句子相似度最高的标签。为此，我们使用预训练的 BERT 模型 [1] 对 AudioSet 的所有类别以及 Clotho数据集中的所有描述进行编码。然后，根据 BERT 嵌入的余弦相似度，为每条描述选出最相似的类别标签。我们使用从 Clotho 构建的这些音频-标签对，通过优化标准二元交叉熵损失对 PaSST 进行微调。

[1]https://www.sbert.net/

2.3. 编码器

为了利用预训练模型，编码器架构与 PaSST 相同，包含 12 个编码器块和 12 个注意力头，隐藏维度为 $d = 768$。每一层包含两个子层：一个多头自注意力层和一个逐位置全连接前馈层。前馈网络包含两个线性层，中间使用 GELU 激活函数。由于使用 *Patcout* 来解决过拟合问题，我们在 Transformer编码器中不使用 dropout。如图 1 所示，输入序列由提取的块和嵌入后的引导文本拼接而成。与 ViT [11] 和 AST [5], 一样，全局可学习的类别标记 $x_{cls} \in \mathbb{R}^{1 \times d}$ 被添加到输入序列的开头。PaSST 的分类层被省略，最后一个隐藏层则传入解码器的交叉注意力层。

2.4. 解码器

解码器由词嵌入层、Transformer 解码器块和线性层组成。输入序列中的每个词都通过词嵌入层转换为向量 $x_i \in \mathbb{R}^{512}$，并馈送到 Transformer 解码器中。词向量由在 Clotho数据集语料上训练的预训练 Word2Vec 模型 [12] 提取。

Transformer 解码器包含 6个模块和8个头，嵌入维度为 512。每个模块由一个自注意力层、一个交叉多头注意力层和一个前馈层组成。前馈层的维度为2048。编码器的输出经过一个前馈层和一个非线性函数调整后，馈送至解码器的交叉多头注意力层。位置嵌入被添加到输入序列中，并对其应用掩码。最后，解码器的输出经过一个线性层和 softmax 函数，以获得描述词的输出概率。

模型的训练目标是最小化交叉

模型	BLUE-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE _L	CIDEr	SPICE	SPIDEr
基线	0.555	0.358	0.239	0.156	0.164	0.364	0.358	0.109	0.233
PACT_no_s	0.576	0.384	0.261	0.176	0.166	0.385	0.453	0.130	0.292
PACT_s	0.578	0.384	0.262	0.176	0.177	0.387	0.454	0.133	0.293
PACT_s_clr	0.575	0.384	0.262	0.174	0.178	0.386	0.457	0.133	0.295
PACT_2s	0.579	0.386	0.262	0.173	0.178	0.387	0.457	0.134	0.296

表 1：不同模型在 Clotho 评估集上的性能。

熵（CE）损失：

$$\mathcal{L}_{\mathcal{CE}}(\theta) = -\frac{1}{T}\sum_{t-1}^{T} logp(y_t|y_{<t},\theta) \tag{1}$$

其中 θ 是模型参数， y_t 是时间步 t的真实词。

2.5. 迁移学习

由于本任务允许使用外部数据，我们尝试了两种迁移学习方案：利用在 ImageNet和 AudioSet 上训练的 PaSST编码器，以及在更大的域内数据集上对我们的模型进行预训练。

2.5.1. 预训练编码器

PaSST 以在 ImageNet 上预训练的 ViT 作为其骨干网络。该模型在 AudioSet 数据集上训练，在不同的下游音频模式识别任务中展现出强大的音频特征提取能力 [1]。与 AST 不同，PaSST 使用独立的嵌入进行时间和频率位置编码。将时间和频率嵌入解耦的优势在于，无需微调或插值即可处理可变长度的输入。

2.5.2. 使用域内数据集进行预训练

此外，我们还创建了一个扩大的域内数据集，将 AudioCaps 数据集中的约 46,000 个单描述音频样本 [13] 和 MACS 中的 3930 个多描述音频样本 [14] 加入 Clotho 开发-验证集划分。我们首先在该数据集上对模型进行预训练，然后在 Clotho 上对其进行微调。

2.6. 数据增强

为避免过拟合并对数据进行正则化，我们使用 SpecAugment [15], 标签平滑 [16], 和 Mixup [9]。由于自动音频标注不是分类任务，因此在标签上应用 Mixup 并非易事。不同于 [17] 通过拼接字幕来应用 Mixup 的做法，我们在嵌入空间中混合字幕。对于采样得到的两个频谱图 x_i、 x_j 及其对应的嵌入字幕 y_i^e、 y_j^e，我们按如下方式应用 Mixup：

$$x = \lambda x_i + (1-\lambda)x_j$$
$$y^e = \lambda y_i^e + (1-\lambda)y_j^e$$

其中， $\lambda = Beta(a,a)$和 $a = 0.3$。我们还混合了嵌入的引导文本。尽管在计算损失时使用未混合的字幕，这种Mixup实现提升了我们模型的结果。

3. 实验

3.1. 数据集

3.1.1. Clotho

最新的 Clothov2 数据集包含训练集中的3839个音频片段，以及验证集和评估集中的1045 个音频片段。音频片段采集自 Freesound，时长从 15 秒到 30秒不等。标注人员通过亚马逊机械土耳其人受聘，以众包方式生成描述。每个音频片段都有五条对应的描述，长度从 8个词到20 个词不等。为防止产生偏差的标注，标注人员只能获取音频信号。Clotho 的数据集划分通过分层过程创建。

3.1.2. AudioCaps

AudioCaps [13] 是规模最大的音频描述数据集，包含约 46000 个样本。所有音频片段均来自 AudioSet，时长为 10 秒。标注工作由通过亚马逊机械土耳其人平台招募的众包工作者完成。训练集中的音频样本各对应一条描述，而验证集和测试集中的音频样本各对应五条描述。

3.1.3. MACS

MACS [14] 由来自 TAU 城市声学场景 2019数据集的音频片段组成。它包含3930 个时长为10秒的音频片段，但未提供子集划分。每个音频样本的描述数量从两条到五条不等。

3.2. 数据预处理

所有音频片段均转换为32kHz。使用 1024 点汉宁窗并采用 50% 重叠提取对数梅尔频谱图，并使用 128个梅尔频带作为输入特征。字幕经过分词处理并转换为小写，同时移除标点符号。<SOS> 和 <EOS> 标记分别添加到每条字幕的开头和结尾。在增强数据集上进行预训练期间，不在 Clotho 词汇表中的词将替换为 <UNK>。

3.3. 实验设置

我们按照 [18]将预训练阶段分为两部分。首先，我们冻结 PaSST编码器，并以学习率 1×10^{-4} 在扩展后的数据集上进行训练。然后解冻编码器，并使用 1×10^{-5} 的学习率。在微调期间，初始学习率 1×10^{-5} 通过线性预热逐渐增加至 1×10^{-4}。所有阶段的批量大小均为 32。在推理阶段，我们采用束宽为 3 的束搜索。

在预训练期间，我们应用 *Patchout*，丢弃 4 个频率区间和 80 个时间帧；在 Clotho 上微调时，丢弃 120 个时间帧

。这意味着微调期间，输入序列中几乎有一半会被丢弃，从而大幅降低复杂度。为进一步缓解过拟合，我们在解码器中使用 dropout，比率设为 0.2。标签平滑因子设为 0.1。

我们使用创建的音频-标签对对预训练的 PaSST 模型进行微调，训练 1 个 epoch，学习率为 1×10^{-5}。开发-验证集划分的输入引导文本在训练前获得。

4. 结果

我们的提交包含以下四种模型 s:

- PACT80 $p_t =$ no s：此模型采用时间 Patchout 进行训练。输入文本选取 PaSST 的 logits 中最大值对应的文本。
- PACT_s：该模型采用时间 Patchout $p_t = 120$进行训练。输入文本通过 top-p 采样选取。
- PACT_s_clr：与 PACT_s 相同，在 Clotho 上进行微调时采用恒定学习率 1×10^{-5}，而非线性预热。
- PACT_2s：该模型采用时间 Patchout $p_t = 120$进行训练。输入文本是通过 top-p 采样选取的两个 AudioSet 类别标签的拼接结果。

表 1 将我们提交的模型与基线模型进行了性能比较。

5. 结论

本报告介绍了我们提交至 DCASE2021 挑战赛任务 6 的系统。我们提出了一种结合音频与文本条件的 Transformer 架构。我们证明，*Patchout* 可有效应用于音频描述，在降低计算复杂度的同时提升性能。为获得在语义上与真实描述相似的输入引导文本，我们对预训练分类模型进行了微调。为解决数据稀缺问题，我们在更大的域内数据集上对模型进行预训练，并使用预训练的 PaSST 初始化编码器权重。我们最佳模型在开发-测试数据集上的 SPIDEr 分数为 0.296。

6. 参考文献

[1] K. Koutini、J. Schlüter、H. Eghbal-zadeh 和 G. Widmer，“采用Patchout 的高效音频 Transformer训练”，*CoRR*，第abs/2110.05069 卷，2021年。 [在线]。可访问：https://arxiv.org/abs/2110.05069[2] X. Mei、Q. Huang、X.Liu、G.Chen、J. Wu、Y. Wu、J. Zhao、S.Li、T. Ko、H. L. Tang、等，“一种基于编码器—解码器的音频描述系统，结合迁移学习和强化学习”，*arXiv* 预印本*arXiv:2108.02752*，2021年。[3]Z.Ye、H.Wang、D. Yang和Y.Zou，“通过整合声学信息和语义信息提升自动音频描述的性能”，*CoRR*，第abs/2110.06100 卷，2021年。 [在线]。可访问：https://arxiv.org/abs/2110.06100[4] X. Mei、X.Liu、Q. Huang、M. D. Plumbley 和 W. Wang，“音频描述 Transformer”，*arXiv* 预印本 *arXiv:2107.09817*，2021年。[5] Y. Gong、Y. Chung 和 J. R. Glass，“AST：音频频谱图 Transformer”，*CoRR*，第 abs/2104.01778 卷，2021年。 [在线]。可访问：https://arxiv.org/abs/2104.01778[6] Z. Ye、H. Wang、D. Yang 和 Y. Zou，“通过整合声学和语义信息提高自动音频描述的性能”，*arXiv* 预印本 *arXiv:2110.06100*，2021年。[7] 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 - 第六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。[在线]。可访问：https://hal.inria.fr/hal-03522488[8] Y. Koizumi、Y. Ohishi、D. Niizumi、D. Takeuchi 和 M. Yasuda，“使用由基于音频的相似描述检索引导的预训练大规模语言模型进行音频描述”，*arXiv* 预印本 *arXiv:2012.07331*，2020年。[9] H. Zhang、M. Cisse、Y. N. Dauphin 和 D. Lopez-Paz，“Mixup：超越经验风险最小化”，*arXiv* 预印本 *arXiv:1710.09412*，2017年。[10] A. Holtzman、J. Buys、L. Du、M. Forbes 和 Y. Choi，“神经文本退化的奇特案例”，*arXiv* 预印本 *arXiv:1904.09751*，2019年。[11] A. Dosovitskiy、L. Beyer、A. Kolesnikov、D. Weissenborn、X. Zhai、T. Unterthiner、M. Dehghani、M. Minderer、G. Heigold、S. Gelly、J. Uszkoreit 和 N. Houlsby，“一幅图像胜过 16x16 个词：用于大规模图像识别的 Transformer”，*CoRR*，第 abs/2010.11929 卷，2020年。[在线]。可访问：https://arxiv.org/abs/2010.11929[12] T. Mikolov、K. Chen、G. Corrado 和 J. Dean，“向量空间中的词表示高效估计”，*arXiv* 预印本 *arXiv:1301.3781*，2013年。[13] C. D. Kim、B. Kim、H. Lee 和 G. Kim，“AudioCaps：为真实环境中的音频生成描述”，载于计算语言学协会北美分会*2019*年会议论文集：人类语言技术，第*1*卷（长篇和短篇论文）。明尼阿波利斯，明尼苏达州：计算语言学协会，2019年6月，第119–132页。 [在线]。可访问：https://aclanthology.org/N19-1011[14] I. Martin Morato 和 A. Mesaros，“音频描述数据集中的多样性和偏差”，载于第六届声学场景与事件检测和分类研讨会（*DCASE 2021*）论文集，F. Font、A. Mesaros、D. P.W. Ellis、E. Fonseca、M. Fuentes 和 B. Elizalde 编，2021年11月，第90–94页，声学场景与事件检测和分类；会议日期：2021年11月15日至2021年11月19日。[15] D. S. Park、W. Chan、Y. Zhang、C.-C. Chiu、B. Zoph、E. D. Cubuk 和 Q. V. Le，“SpecAugment：一种用于自动语音识别的简单数据增强方法”，*arXiv* 预印本 *arXiv:1904.08779*，2019年。[16] C. Szegedy、V. Vanhoucke、S. Ioffe、J. Shlens 和 Z. Wojna，“重新思考用于计算机视觉的 Inception 架构”，*CoRR*，第 abs/1512.00567 卷，2015年。 [在线]。可访问：http://arxiv.org/abs/1512.00567[17] N. Kuzmin 和 A. Dyakonov，“声学场景与事件检测和分类 2020年自动音频描述技术报告”。

[18] W. Yuan、Q. Han、D. Liu、X. Li 和 Z. Yang，《DCASE 2021 挑战赛任务 6 系统：采用弱监督预训练和词语选择方法的自动音频描述》，DCASE2021 挑战赛，技术报告，2021年7月。