序列到序列的钢琴转录与变压器

Curtis Hawthorne Ian Simon Rigel Swavely EthanManilow[?]
Jesse Engel
谷歌研究
{峡湾,iansimon, rigeljs、 emanilow jesseengel} @google.com

摘要。
近年来，通过在大型数据集上训练自定义深度神经网络，自动音乐转录取得了重大进展。然而，这些模型需要广泛的特定领域的网络架构设计、输入/输出表示和复杂的解码方案。在这项工作中，我们证明了使用具有标准解码方法的通用编码器-解码器变压器可以实现等效的性能。我们证明，该模型可以学习将谱图输入直接翻译为类似midi输出事件，用于多个转录任务。这种序列到序列的方法通过联合建模音频特征和类似语言的输出依赖关系来简化转录，从而消除了对特定任务架构的需求。这些结果指出了通过关注数据集创建和标记而不是自定义模型设计来创建新的音乐信息检索模型的可能性。

1. 介绍

音乐自动转录(AMT)是音乐信息检索(MIR)的核心任务之一。AMT的目标是将原始音频转换为适当的符号表示。在本文中，我们考虑将钢琴音频转录为一系列音符事件的问题，这些音符事件指示精确的开始/偏移时间和速度，而不是与格律网格对齐的乐谱。

钢琴转录的最新进展在很大程度上是由两个因素驱动的:包含对齐的钢琴音频和MIDI的数据集的构建和发布(最著名的是MAPS[1]和MAESTRO[2])，以及使用专门为钢琴转录设计的架构的深度神经网络(例如，分别为音符开始和音符存在建模的Onsets和Frames架构[3])。虽然特定领域的模型已经导致了基准数据集的改进，但目前尚不清楚这些模型是否有效方法可以转化为其他领域和MIR任务。

同时，采用自关注的Transformer模型[4]已经证明了一种惊人的能力，可以通过简单地改变输入和输出表示，在具有相同核心架构的各种领域中获得最先进的结果[5-14]。

在本文中，我们证明了一个通用的transformer模型可以实现最先进的钢琴转录，而无需任何特定于领域的调整。使用“现成的”组件(T5论文[6]中基本上未修改的编码器-解码器配置)和简单的贪婪解码策略，我们训练了一个模型来编码原始谱图帧，并直接解码到由原始MIDI协议[15]中的消息(例如，note on和velocity消息)启发的一系列音符事件。因此，在本文的其余部分中，我们将模型的输出称为“类似midi”。我们在3.2节中提供了模型词汇表的详细信息。

此外，我们证明了这种领域不可知的方法使我们能够通过仅更改训练标签而不修改输入或模型来训练转录任务的几种变体(例如，仅转录笔记发作)。

总之，这项工作说明了使用通用序列到序列转换器进行钢琴转录的价值，而不需要特定域的适应，并指出了将类似方法扩展到各种MIR任务的潜力。

2. 相关工作

2.1 钢琴乐谱

利用基于音频和MIDI数据集训练的深度神经网络模型在钢琴转录方面取得了很大进展。2012年，Boulanger-Lewandowski等人[16](在Nam等人[17]的声学模型的基础上构建)训练了一个循环神经网络(RNN)转录模型来输出二进制钢琴卷。Böck和Schedl[18]训练了一个类似的基于rnn的模型，仅用于钢琴奏响。Hawthorne等人[3]通过使用单独的基于卷积的模型*堆栈*来检测音符开始、音符存在和音符速度，从而提高了转录精度。模型输出使用硬先验(hard prior)解码为离散音符，除非开始预测器给出的概率大于0.5，否则不启动音符。

? Work done as a Google Brain Student Researcher.

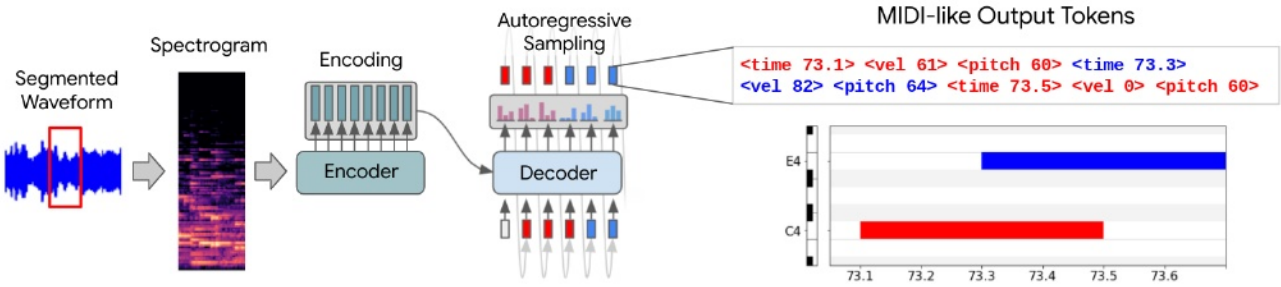

图1所示。我们的模型是一个通用的编码器-解码器Transformer架构，其中每个输入位置包含单个频谱图帧，每个输出位置包含来自midi类词汇表的事件。输出令牌从解码器中自回归采样，在每一步以最大概率取令牌。

最近，钢琴转录方面的进展主要涉及添加更多特定领域的深度神经网络组件和修改解码过程。在大多数情况下，这种额外的复杂性已经面向提高piano转录精度的特定目的。

Kong等人[19]使用与Hawthorne等人[3]相似的网络架构，通过使用回归预测精确的连续on-set/offset时间，实现了更高的转录精度。Kim和Bello[20]在转录输出上使用对抗性损失来鼓励转录模型输出更可信的钢琴滚动。我们的序列到序列方法通过自回归解码器明确地建模这种输出间依赖关系，该解码器与提取有意义的音频特征的编码器端到端进行训练。

Kwon等人[21]使用各种语言模型来模拟每个音高的音符状态转换，而不是具有单独的起始、帧和偏移堆栈。然而，解码过程相当复杂，特别是处理不同音高之间的交互。类似地，Kelz等人[22]在基于攻击-衰减-持续释放(ADSR) en-velopes的音符状态上使用隐马尔可夫模型进行解码。

在一个非常彻底的特定领域处理中，Elows-son[23]构建了一个分层模型，该模型从频谱图中提取基本频率轮廓，并使用这些轮廓来推断音符的起跳和偏移。虽然对于许多应用来说，像Engel等人[24]那样具有这样的中间表示可能是有用的，但在这项工作中，我们将从音频到离散音符的复调转录视为端到端的问题。这具有概念简单的优点，我们的评估(第4.2节)表明它也是有效的。

2.2 变形金刚

最近，一种通用的Transformer架构[4]被用于跨多个领域来解决序列到序列的问题，取代了以前使用的特定于任务的架构。在变形金刚最初出现并被广泛使用的自然语言处理领域之外(例如Brown等人的GPT-3[25]和rafael等人的T5[6])，变形金刚已被用于计算机视觉任务，如物体检测[8]、基于字幕的图像生成[9]和姿态重建[10]，以及音频相关任务，包括语音识别[11,12]、语音合成[13]和音频事件分类[14]。

请注意，Transformer的上述许多用途都利用了*预训练*阶段，其中模型使用自我监督在大量未标记的数据上进行训练。虽然这样的预训练阶段也可能有助于音乐转录，但在这项工作中，我们探索了一种更简单的设置，即以普通的监督方式在标记的转录上从头开始训练Trans-former架构。

2.3 Sequence-to-Sequence转录

使用变形金刚进行音乐转录的想法也被考虑过。Awiszus在2019年[26]探索了音乐转录作为序列到序列问题的几种公式，使用LSTM[27]和Transformer模型使用各种输入和输出表示(包括类似于我们自己的表示)。然而，该论文无法证明明显的成功，这似乎是由于在钢琴转录中使用了帧式多f0评估而不是基于音符的评估标准，使用了相对时移而不是绝对时移(参见3.2节)，并且在MAPS数据集上进行训练，该数据集比我们使用的MAESTRO数据集小得多。早些时候，Ullrich和van der Wel[28]似乎是第一个将mu- music转录作为序列到序列的问题(使用LSTMs而不是变形金刚)，但他们的系统只能处理单音音乐。

3. 模型

如上所述，我们的模型是一个通用的编码器-解码器Transformer架构，其中每个输入位置包含单个频谱图帧，每个输出位置包含来自midi类词汇表的事件。我们的模型和我们的输入和输出设置的概述如图1所示。

输入通过一堆编码器自注意层进行处理，从而产生与原始输入长度相同的嵌入序列。一叠解码器层-

然后Ers在解码器输出上使用随机屏蔽的自注意，并在编码器堆栈的完整输出上使用交叉注意。至关重要的是，这允许符号-ken输出是可变长度的，仅取决于描述输入音频所需的令牌数量。

3.1 模型架构

模型配置基于T5[6]中的“小”模型，并根据T5.1.1配方[1]的建议进行修改。具体来说，我们的模型使用嵌入大小d_{model}= 512，前馈输出维数d_{ff}= 1024，键/值维数d_{kv}= 64,6头注意力，编码器和解码器各8层。

与标准配置相比，我们的模型有一些小的变化。最重要的是，为了使用连续的频谱图输入，我们添加了一个密集层，将每个频谱图输入帧投影到Transformer的输入嵌入空间。我们还使用固定的绝对位置嵌入，而不是T5中使用的对数缩放的相对位置桶嵌入，以确保所有位置都能以相同的分辨率进行处理。最后，我们使用float32激活来获得更好的训练稳定性，因为我们的模型足够小，我们不需要像大型T5模型中通常使用的不太精确的bfloat16 for-mat[29]那样的内存效率。

该模型是使用T5X框架2实现的，该框架[2]是建立在flex[30]和JAX[31]之上的。我们还使用SeqIO [3]进行数据预处理和评估。我们实现的代码将在https://goo.gl/magenta/seq2seq-piano-transcription-code上提供。

虽然最近一些使用变形金刚的研究倾向于非常大的模型，例如具有175B参数的GPT-3[25]，但我们发现相对较小的模型足以完成这些任务。在上面描述的配置下，我们的模型只有54M个参数，大约是Onsets和Frames[2]的两倍，后者有28M个pa参数。

3.2 输入和输出

该模型使用谱图帧作为输入，每个输入位置有一帧。为了匹配T5设置，我们使用可学习的EOS(End of se - sequence)嵌入来终止输入序列。每一步的模型输出是一个离散事件词汇表上的softmax分布，如下所述。这个词汇表很大程度上受到MIDI规范中最初定义的消息的启发[15]。使用事件作为输出表示而不是钢琴滚动矩阵具有更稀疏的优点，因为只有在事件发生时才需要输出，而不需要对每一帧都进行注释。词汇表由以下标记类型组成:

Note[128个值]表示128个MIDI音高之一的音符开启或音符关闭事件。为了灵活，我们使用了完整的MIDI音高范围，但在这些实验中，实际上只使用了与钢琴琴键相对应的88个音高。

Velocity[128个值]表示要应用于所有后续Note事件的速度变化(直到下一个Velocity事件)。有128个速度值，包括0，这是一个特殊的值，它导致后续的Note事件被解释为noteoff事件。

Time[6000个值]表示时间段内的绝对时间位置，量化为10ms的bin。这个时间将应用于所有后续的Note事件，直到下一个time事件。时间事件必须按时间顺序发生。为了灵活起见，我们将时间定义为60秒以内，但由于每段时间都会重置，因此在实践中我们只使用这种类型的前几百个事件。

EOS [1 value]表示序列的结束。

以前使用这种midi类事件词汇表的工作[32]使用事件之间的相对时间移位，表示自上次时间移位以来经过的时间量。然而，在序列到序列的场景中，输出中早期的单个相对时移错误会导致所有后续的输出步骤不正确，并且随着序列长度的增加，此类错误会逐渐累积。为了调整这种漂移，Transformer模型必须学会对所有先前的时移执行累积和，以便及时确定当前的位置。我们转而使用绝对时间，其中每个时间事件表示从片段开始的时间量，如图1所示。这让模型更容易独立地确定每个时间戳;我们还在4.4节中对这种选择进行了实证研究，并发现使用绝对时移而不是相对时移可以获得更好的性能。

对于我们的时间事件，我们使用10毫秒的时间分辨率，因为一些实验发现，这个位移大约是人类感知的极限[33](尽管其他人报告了更小的值，例如Handel的5毫秒[34])。我们保留了一种可能性，即我们的结果可以通过更精细的事件分辨率得到进一步改善，例如通过预测Kong等人[19]中的连续时间。

在推理过程中解码模型输出是用简单的贪婪自回归算法完成的。我们在每一步选择最大概率事件，并将其作为该步骤的预测事件反馈到网络中。我们继续这个过程，直到模型预测到一个EOS to-ken。

使用事件序列作为我们的训练目标，而不是钢琴滚动矩阵或其他基于帧的格式，可以实现显著的灵活性。例如，我们在4.4节中演示了与

1 https://github. com/google-research/text-to-text-transfer-transformer/blob/master/released_checkpoints.md#t511

2 https://goo.gle/t5x

3 http://github.com/google/seqio

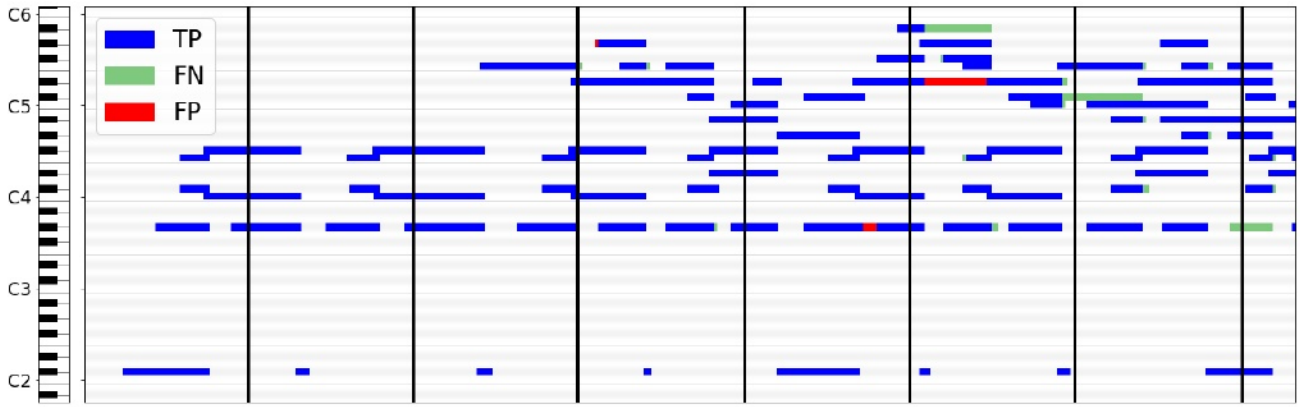

图2。来自MAESTRO验证集与地面事实对比的肖邦降*d大调Berceuse Op. 57*模型事件输出的钢琴滚动渲染部分。黑色竖线表示推理过程中的段边界。真阳性(TP)帧用蓝色标记，假阴性(FN)帧用绿色标记，假阳性(FP)帧用红色标记。请注意，该模型成功地预测了注记事件发生在不同片段中的笔记的注记事件。

同样的输入可以训练为只预测起跳(仅使用Note、Time和EOS事件)或起跳、偏移和速度(使用上面的完整词汇表)。唯一需要改变的是使用一组不同的令牌作为训练目标。这与之前的工作形成对比，在之前的工作中，预测一个新特征需要添加新的输出头(或整个堆栈)，为这些输出设计损失，并修改(通常是不可微的)解码算法，将所有模型输出组合成最终所需的表示。

通过使用序列到序列的方法，我们的模型可以通过在完全可微的端到端训练集中联合建模音频特征和类似语言的输出依赖关系来直接输出我们想要的表示。添加新的输出特征或更改任务定义仅仅是更改用于描述目标输出的标记的问题。

3.3 序列长度注意事项

变形金刚可以在每一层处理序列中的所有标记，这特别适用于需要细粒度信息的转录任务，例如每个事件的音高和时间。然而，这种注意机制相对于序列长度n的空间复杂度为$O(n^2)$。实际的后果是，用于转录的大多数音频序列无法装入内存。为了解决这个问题，我们在训练和推理过程中将音频序列及其相应的符号描述拆分为更小的片段。

在训练期间，我们对批处理中的每个序列使用以下过程:

1. 从完整的序列中选择一个随机的音频片段作为模型输入。所选段的长度可以从单个输入帧变化到最大输入长度，并且从均匀随机分布中选择起始位置。

2. 为所选音频段对应的训练目标选择符号段。是- - - - - -
因为注释可能从一个片段开始，在另一个片段结束，模型被训练成能够预测未观察到注释事件的情况下的注释事件。
3. 计算所选音频的谱图，并将符号序列映射到我们的词汇表中(参见3.2节)。计算符号段内的绝对时移，使时间0为段的开始。

4. 提供连续谱图输入和一次热编码的midi类事件作为Transformer架构的训练示例。

在推理过程中，使用以下过程:

1. 在可能的情况下，使用最大输入长度将音频序列分割成不重叠的片段，然后计算谱图。

2. 依次为每个片段提供频谱图作为Transformer模型的输入，并通过在每个步骤中根据模型输出贪婪地选择最可能的令牌进行解码，直到预测出EOS令牌。超出音频段长度的时移之后出现的任何令牌都将被丢弃。

3. 将所有片段的解码事件连接成单个序列。在连接之后，可能仍然存在没有相应注释的noteoff事件;我们将这些去掉。如果我们遇到一个已经开始的音调的注释事件，我们结束这个注释并开始一个新的。在序列的最后，我们结束任何缺少音符事件的活动音符。

该模型在预测相应事件在不同片段中的笔记或笔记事件方面具有惊人的能力，如图2所示。这种能力也通过模型在第4.2节中关于起始、偏移和速度F1分数的结果得到了实证证明。

4. 实验

我们使用Adafactor优化器[35]训练我们的模型，批量大小为256，恒定学习率为1e−3，子层输出和嵌入式输入的dropout设置为0.1。选择批大小是为了最大限度地提高训练吞吐量，因为我们在初始实验中尝试的其他批大小似乎对最终性能没有影响。学习率和dropout值被设置为T5用于微调任务的相同值。

使用Tensor-flow [36] *tf计算输入谱图。图书馆的信号*。我们使用的音频采样率为16000 kHz, FFT长度为2048个样本，跳宽为128个样本。我们将输出缩放到512梅尔箱(以匹配模型的嵌入大小)，并使用对数尺度的幅度。

输入序列被限制在512个位置(511个谱图帧加上一个可学习的EOS嵌入)，输出被限制在1024个位置(1023个符号序列加上一个可学习的EOS嵌入)。这对应的最大片段长度为4.088秒。我们使用512个输入位置来匹配T5的序列长度，但未来的工作可以探索其他序列长度是否会带来更好的性能。之所以使用1024个输出位置，是因为我们发现512个输出位置并不总是足以象征性地描述输入音频。

我们在32个TPUv3内核上训练了所有模型，结果每核批处理大小为8。为了提高训练速度，我们使用了这种配置，但是模型足够小，可以在单个TPUv2实例(8核)上进行训练。基于验证集的结果，过度拟合似乎并不是一个问题，所以我们允许训练进行400K步，这对于我们的基线模型来说大约需要2.5天。

4.1 数据集

为了评估我们的模型在钢琴转录任务上的性能，我们使用了MAESTRO数据集[2]，该数据集包含了大约200小时的精湛钢琴演奏，并在音频和地面真实音符注释之间进行了精细校准。为了与之前的转录工作进行比较，我们在MAESTRO V1.0.0上进行训练，但对于其他研究，我们使用MAESTRO V3.0.0，因为它包含包含26小时数据的额外92场表演。MAESTRO V3.0.0*还包含低音*和*低音*踏板事件，尽管我们的模型(和评估)没有使用这些。我们也没有像Kong等人[19]那样直接对维持踏板事件进行建模，而是像Hawthorne等人[2]一样，在按下维持踏板时延长音符持续时间。

4.2 评价

在评估钢琴转录系统的性能时，我们使用了Note F1评分指标:检测单个音符的精度和召回率的谐波平均值。这涉及到将每个预测音符与基于起始时间、音高和可选偏移时间的唯一的基音真音匹配。此外，起始速度可用于丢弃具有显著不同速度的匹配。我们主要使用考虑起跳、偏移和速度的F1分数。我们还包括仅考虑起病或起病和偏移的F1分数的结果。我们根据*mir_eval*[37]库对我们使用的(标准)转录指标进行了精确定义。

由于钢琴是一种打击乐器，与偏移量相比，准确识别音符开始通常更容易(在感知上也更重要)[38]。我们使用*mir_eval*的默认匹配容差为50 ms来识别起跳，50ms或音符持续时间的20%来识别偏移。

4.3 与之前工作的比较

我们将我们的序列到序列方法与表1中MAESTRO数据集V1.0.0上以前钢琴转录论文的报告分数进行比较。与现有的最佳方法相比，我们的方法能够获得具有竞争力的F1分数，同时在概念上非常简单，使用通用架构和解码算法以及标准表示。

4.4 烧蚀研究

我们使用表1中MAESTRO数据集的V3.0.0对模型的一些组件执行消融研究。首先，我们验证了这种架构的灵活性，以使用一组不同的特征来描述输入音频。我们修改符号数据，仅通过使用Note、Time和EOS事件来描述发作。模型训练成功，并在这个修改后的仅发作任务上获得了很高的F1分数。

接下来，我们研究不同的输入表示。对于“STFT”，我们在FFT计算后去除对数尺度。这样得到的输入帧大小为1025，被密集层投影到模型嵌入大小为512。对于“原始样本”，我们简单地根据谱图使用的跳宽(128个样本)将au-dio样本分成几段，并直接使用这些样本作为输入，再次由密集层投影到嵌入大小。这两种配置都可以成功训练，但表现不如log mel输入。我们怀疑这是因为mel缩放产生了有用的特征，否则模型将不得不使用它的一些能力来提取这些特征。

我们还通过训练具有相对时移的模型来验证绝对时移更适合这种架构。正如预期的那样，它并没有表现得那么好。进一步，我们注意到验证集上基于笔记的评估指标在训练期间变化很大，相邻验证步骤之间的起始F1分数有时相差多达15分。我们假设这是因为相对时移预测的微小变化在整个序列中累积以确定度量计算所需的绝对时间时会被放大;也就是说，相对时移会导致最终的转录结果与音频偏离一致。

最后，我们研究了更大的模型尺寸是否会提高性能。我们根据T5的“基础”配置扩大了模型大小。具体来说,我们

	Model	Onset, Offset, & Velocity F1	Onset & Offset F1	Onset F1
MAESTRO V1.0.0	Transformer (ours)	**82.18**	**83.46**	95.95
	Kong et al. 2020 [19]	80.92	82.47	**96.72**
	Kwon et al. 2020 [21]	–	79.36	94.67
	Kim & Bello 2019 [20]	80.20	81.30	95.60
	Hawthorne et al. 2019 [2]	77.54	80.50	95.32
MAESTRO V3.0.0	Transformer	**82.75**	**83.94**	96.01
	Onsets only vocabulary	–	–	**96.13**
	STFT input	81.81	82.92	95.44
	Raw samples input	74.79	77.26	92.35
	Relative time shifts output	66.25	67.35	80.02
	“Base” model size (100K steps)	81.41	82.78	95.60

表1。MAESTRO测试集结果。使用V1.0.0与以前的工作进行比较，使用V3.0.0与不同的模型配置进行比较，因为它的尺寸更大。所有Transformer模型都被训练到400K步，除了“Base”配置被训练到100K步。

修改了以下超参数:d_{model}= 768，d_{ff}= 2048,12个头部用于注意，编码器和解码器各12层。这些改变导致了一个有213M参数的模型，而不是我们的“小”配置，只有54M。这个模型快速过拟合训练数据集，在100K步之后，验证集上的分数开始下降，所以我们在那一点上停止训练。即使提前停止，这个模型的表现也不如我们的“小”配置好，这清楚地表明，尽管钢琴抄写是一项相当复杂的任务，但它并不需要特别大的Transformer。

5. 结论和未来的工作

我们已经证明，经过训练的通用Transformer架构可以将谱图映射到类似midi的输出事件，而无需预训练，可以在自动钢琴转录上实现最先进的性能。我们认为这是对简单性的呼吁;我们尽可能地使用标准格式和ar架构，并且能够达到与钢琴转录定制模型相当的结果。在我们的设置中，复杂性的主要来源可能是将示例划分为片段;未来的工作可能包括对稀疏注意力机制的研究，以实现在单个编码和解码通道中转录整首音乐。同样值得探索的是使用蒸馏[39]或相关技术，使这样的模型能够在移动设备或网络上实时运行。

我们的研究结果表明，具有transformer的通用序列到序列框架也可能有利于其他MIR任务，例如拍跟踪，基频估计，和弦估计等。自然语言处理领域已经看到，一个单一的大型语言模型，如GPT-3或T5，已经能够通过利用任务之间的共性来解决多个任务。我们对MIR任务可能出现类似现象的可能性感到兴奋，我们希望这些结果指向通过专注于数据集创建和标记而不是自定义模型设计来创建新的MIR模型的可能性。

6. 引用

[1] V. Emiya, R. Badeau, and B. David, “Multipitch esti-mation of piano sounds using a new probabilistic spec-tral smoothness principle,” *IEEE Transactions on Au-dio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1643–1654, 2009.

[2] C. Hawthorne, A. Stasyuk, A. Roberts, I. Si-mon, C.-Z. A. Huang, S. Dieleman, E. Elsen, J. Engel, and D. Eck, “Enabling factorized piano music modeling and generation with the MAE-STRO dataset,” in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview. net/forum?id= r1lYRjC9F7

[3] C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. Engel, S. Oore, and D. Eck, “Onsets and Frames: Dual-objective piano transcription,” in *ISMIR*, 2018.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin, “At-tention is all you need,” in *NeurIPS*, 2017.

[5] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weis-senborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszko-reit, and N. Houlsby, “An image is worth 16x16 words: Transformers for image recog-nition at scale,” in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy

[6] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, “Exploring the limits of transfer learning with a unified text-to-text

Transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020.

[7] A. Jaegle, F. Gimeno, A. Brock, A. Zisserman, O. Vinyals, and J. Carreira, "Perceiver: General perception with iterative attention," *arXiv preprint arXiv:2103.03206*, 2021.

[8] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kir-illov, and S. Zagoruyko, "End-to-end object detection with Transformers," in *ECCV*, 2020.

[9] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-shot text-to-image generation," *arXiv:2102. 12092*, 2021.

[10] K. Lin, L. Wang, and Z. Liu, "End-to-end human pose and mesh reconstruction with Transformer s," in *CVPR*, 2021.

[11] L. Dong, S. Xu, and B. Xu, "Speech-Transformer: a no-recurrence sequence-to-sequence model for speech recognition," in *ICASSP*, 2018.

[12] N.-Q. Pham, T.-S. Nguyen, J. Niehues, M. Müller, S. Stüker, and A. Waibel, "Very deep self-attention networks for end-to-end speech recognition," in *Inter-speech*, 2019.

[13] N. Li, S. Liu, Y. Liu, S. Zhao, and M. Liu, "Neural speech synthesis with Transformer network," in *AAAI*, 2019.

[14] Y. Gong, Y.-A. Chung, and J. Glass, "AST: Audio spectrogram Transformer," *arXiv:2104. 01778*, 2021.

[15] MIDI Manufacturers Association and others, "The complete midi 1.0 detailed specification," *Los Angeles, CA, The MIDI Manufacturers Association*, 1996.

[16] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, "Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription," in *ICML*, 2012.

[17] J. Nam, J. Ngiam, H. Lee, and M. Slaney, "A classification-based polyphonic piano transcription ap-proach using learned feature representations," in *IS-MIR*, 2011.

[18] S. Böck and M. Schedl, "Polyphonic piano note tran-scription with recurrent neural networks," in *ICASSP*, 2012.

[19] Q. Kong, B. Li, X. Song, Y. Wan, and Y. Wang, "High-resolution piano transcription with pedals by regress-ing onsets and offsets times," *arXiv: 2010.01815*, 2020.

[20] J. W. Kim and J. P. Bello, "Adversarial learning for im-proved onsets and frames music transcriptio n," in *IS-MIR*, 2019.

[21] T. Kwon, D. Jeong, and J. Nam, "Polyphonic pi-ano transcription using autoregressive multi-state note model," in *ISMIR*, 2020.

[22] R. Kelz, S. Böck, and G. Widmer, "Deep polyphonic ADSR piano note transcription," in *ICASSP*, 2019.

[23] A. Elowsson, "Polyphonic pitch tracking with deep layered learning," *Journal of the Acoustical Society of America*, vol. 148, no. 1, pp. 446–468, 2020.

[24] J. Engel, R. Swavely, L. H. Hantrakul, A. Roberts, and C. Hawthorne, "Self-supervised pitch detection by inverse audio synthesis," in *ICML Workshop on Self-Supervision in Audio and Speech*, 2020.

[25] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Ka-plan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learn-ers," in *NeurIPS*, 2020.

[26] M. Awiszus, "Automatic music transcription using se-quence to sequence learning," Master's thesis, Karl-sruhe Institute of Technology, 2019.

[27] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735– 1780, 1997.

[28] K. Ullrich and E. van der Wel, "Music transcription with convolutional sequence-to-sequence models," in *ISMIR*, 2017.

[29] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603. 04467*, 2016.

[30] J. Heek, A. Levskaya, A. Oliver, M. Ritter, B. Rondepierre, A. Steiner, and M. van Zee, "Flax: A neural network library and ecosystem for JAX," 2020. [Online]. Available: http://github. com/google/flax

[31] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, "JAX: composable transformations of Python+ NumPy programs," 2018. [Online]. Available: http://github.com/google/jax

[32] S. Oore, I. Simon, S. Dieleman, D. Eck, and K. Simonyan, "This time with feeling: Learning expressive musical performance," *Neural Computing and Appli-cations*, vol. 32, no. 4, pp. 955–967, 2020.

[33] A. Friberg and J. Sundberg, "Perception of just-noticeable time displacement of a tone presented in a metrical sequence at different tempos," *Journal of The Acoustical Society of America*, vol. 94, no. 3, pp. 1859– 1859, 1993.

[34] S. Handel, *Listening: An introduction to the perception of auditory events.* MIT Press, 1993.

[35] N. Shazeer and M. Stern, "Adafactor: Adaptive learn-ing rates with sublinear memory cost," in *International Conference on Machine Learning*. PMLR, 2018, pp. 4596–4604.

[36] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. [Online]. Available: https://www.tensorflow.org/

[37] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, D. P. Ellis, and C. C. Raffel, "mir_eval: A transparent implementation of common MIR metrics," in *ISMIR*, 2014.

[38] A. Ycart, L. Liu, E. Benetos, and M. Pearce, "Inves-tigating the perceptual validity of evaluation metrics for automatic piano music transcription," *Transactions of the International Society for Music Information Re-trieval*, 2020.

[39] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[40] N. Shazeer, "Glu variants improve transformer," *arXiv preprint arXiv:2002.05202*, 2020.