基于歌词的节奏建模：用于流行歌曲旋律生成

Daiyu Zhang Ju-Chiang Wang KaterinaKosta Jordan B. L. Smith Shicen Zhou字节跳动
{daiyu.zhang, ju-chiang.wang, katerina.kosta, jordan.smith, zhoushicen}@字节跳动.com

arXiv:2301.01361v1 [eess.AS] 3 Jan 2023

摘要

根据预先写好的歌词创作流行歌曲旋律，是作曲家的典型做法。建立歌词如何谱成旋律的计算模型，对于自动作曲系统至关重要；然而，端到端的歌词到旋律模型需要海量的配对训练数据。为缓解数据限制，我们采用两阶段方法，将任务分为歌词到节奏和节奏到旋律两个模块。然而，由于其多模态性，歌词到节奏任务仍颇具挑战。本文提出一种新颖的歌词到节奏框架，其中引入词性标签以实现更好的歌词谱曲效果，并采用专为建模长期音节与音符关联而设计的Transformer架构。对于节奏到旋律任务，我们采用一种经过验证的和弦条件旋律Transformer，该模型已取得最先进的成果。中文歌词到旋律生成实验表明，所提出的框架能够对数据集中节奏和音高分布的关键特征进行建模；在主观评估中，本系统生成的旋律被评为与最先进的替代方案相似或更优。

1. 引言

为歌词谱写旋律，是作曲家常见却复杂的任务。歌词的形式、发音、韵律格律以及表达上的对称性，既能激发旋律编排的灵感，也可能对其形成限制。鉴于旋律的重要性，音乐元创作系统数十年的发展历程中涌现了无数旋律创作系统（综述见 [1]）。然而，歌词转旋律生成任务（即为给定的输入歌词生成旋律）受到的关注较少。该任务面临诸多挑战，包括但不限于：需要正确处理文本韵律（例如，应避免将“the”这样的非重读词谱在旋律的重拍音符上）；需要在旋律中体现歌词的结构；以及首先要创作出优美的旋律。

Zhang、J.-C. Wang、K. Kosta、J. B. L. Smith 和 S. Zhou。根据知识共享署名 4.0 国际许可协议（CC BY 4.0）授权。
署名： D. Zhang、J.-C. Wang、K. Kosta、J. B. L. Smith 和 S. Zhou，“从歌词建模节奏以生成流行歌曲旋律”，载于 第 *23* 届国际音乐信息检索学会会议论文集，印度班加罗尔，2022年。

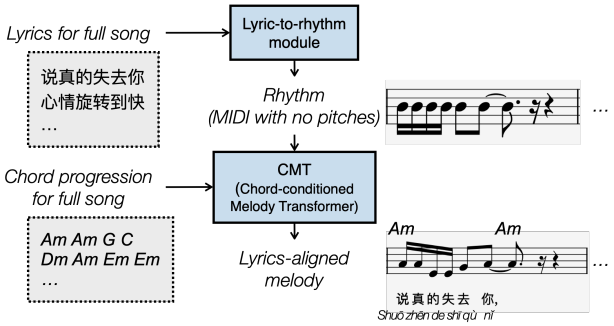


图 1：所提出系统的示意图。

随着深度学习工具的迅速发展，这项任务受到了更多关注，近期也出现了许多歌词到旋律创作系统的实例，其中大多数采用端到端方法 [2–5]。对歌词音节与音符之间关系的建模是一项复杂的跨模态任务，但人们希望借助大量配对示例（即与相应旋律对齐的歌词）取得成功。然而，获取此类数据成本高昂，而无监督学习带来的性能提升有限 [3]。这里提到的所有系统所使用的歌曲歌词训练样本都少于 200,000 条；相比之下，文本到图像系统 DALL-E 拥有 120 亿个参数，训练时使用了数亿组配对的文本—图像数据 [6]。

另一种方案（ [7], 中提出）是选取一种中间表示，并采用两阶段方法：一个模型将歌词转换为所选表示，另一个模型再将其转换为旋律。这样做的动机在于，有足够的数据分别训练这两个模型，而无需采用端到端方法所要求的歌词—旋律配对数据。

我们选择“节奏”作为中间步骤，因为如果不考虑花腔和富有表现力的演唱技巧，就可以假设音节与起音之间、起音与旋律音高之间存在一一对应关系。此外，每个步骤都有充足的数据可用于建模：首先，我们可以从卡拉 OK 式滚动歌词数据中获得歌词音节与音乐中音符起音之间的对齐关系，从而也能得到音符时值和节拍位置。其次，还有多个公开数据集可供学习如何根据音符时长为每个音符分配音高。因此，我们的目标是解决两个子任务，即歌词到节奏和节奏到旋律，并假设节奏生成过程独立于音高生成过程{v1}。

假设节奏生成过程独立于音高生成过程 [7]。

近年来出现了许多旋律生成模型 [8–11], ，但鲜少有人尝试进行歌词到节奏建模。本文提出了一种新颖的框架，采用编码器-解码器 Transformer 架构将歌词转换为节奏 [12]。所提出的系统如图1所示：给定一组输入歌词，歌词到节奏模块会为每个音节分配起音时间和时值。随后，将该节奏与用户提供的和弦进行一同输入和弦条件旋律 Transformer（CMT） [13], ——一种先进的旋律生成系统，以预测每个音符的音高。歌词到节奏模块和 CMT 的详细信息分别见第 3.3 节和第 3.2 节。

2. 背景

歌词与旋律并非随意组合；常识以及先前的分析 [14] 表明，歌词和旋律中的模式彼此相关，且在一定程度上可以通过旋律特征（例如音符时长）和歌词特征（例如音节重音）进行建模。最早一批歌词到旋律系统中，有一种专为处理日语韵律而设计 [15]：首先，将输入文本切分为短语；接着，从一组预先创作的节奏中搜索出符合音节数且与文本重音模式相匹配的节奏；最后，使用动态规划分配音高，使音程走向与词语的自然韵律相协调。另一个更早的歌词到节奏系统也利用了一个由预先创作的节奏构成的数据集，并根据这些节奏与输入文本的音节重音和词语稀有度模式的匹配程度进行评分 [16]。尽管我们的系统与这些研究几乎没有共同之处，但都采用了节奏作为中间表示。

自动音乐生成算法是音乐元创作系统的一个子集 [1], ，这类系统以多种形式存在于西方音乐中，包括用于创作独立作品，以及作为人工作曲过程的一部分离线或实时使用。借助机器学习和深度学习架构，许多此类系统已展现出生成合理结果的能力，这些结果符合给定数据集的音乐特征。有监督生成模型旨在学习训练集分布的内在特征表示。根据模型的不同，这种表示既可以显式呈现，也可以隐式用于从所学分布中生成样本 [17]。

有些系统旨在借助另一段给定内容来生成音乐作品的一部分（包括从旋律创作歌词 [18], ——这正是我们所考虑任务的反向形式）。根据其他模态的数据（如低音线或结构）来设定生成模型的参数，可以实现可控生成系统 [19][第82页-83]。对于利用和弦来引导旋律生成的情况，近期已有一套调整通用对抗网络架构的系统问世 [20], 可选择在给定的

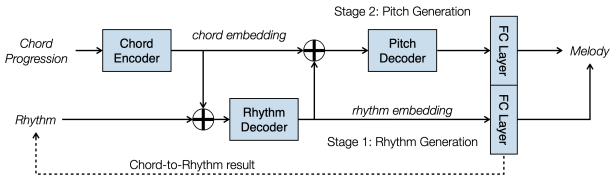


图 2：CMT 的两阶段结构，其中 ⊕ 表示串接。阶段1：和弦到节奏。阶段 2：基于阶段1的结果，由节奏+和弦到音高。

伴奏上生成旋律线。和弦条件旋律变换器（CMT） [13] 是该领域最新的研究成果；我们借鉴了该系统的大部分设计，并将其扩展为能够同时接收歌词和和弦作为输入。下文将在第3节介绍该系统的细节以及我们对其所做的改进。

3. 方法

3.1 系统概述

我们的系统设计受到和弦条件旋律 Transformer（CMT） [13]的启发。CMT 的作者提出了一个两阶段系统，假设生成旋律的过程分为两个阶段，如图2所示：阶段*1*，根据和弦进行生成音符的节奏；阶段*2*，根据和弦进行和生成的节奏为每个音符生成音高。我们提出的系统以一个新颖的歌词到节奏模块替代 CMT 中的和弦到节奏模块（即阶段1），从而对 CMT 进行了扩展。因此，用户可以在我们的系统中输入完整歌曲的歌词和和弦进行（见图1）。随后，歌词到节奏模块生成 MIDI（其中音高为空）。第二步，CMT 处理 MIDI 和和弦进行，生成旋律。这样，节奏是在对歌词的全局视角下生成的，而旋律则是在对节奏和和弦的因果视角下生成的。

在以下各小节中，我们将首先回顾 CMT，并说明在第 3.2 节中修改它以处理歌词到旋律任务所面临的困难。随后，我们将在第3.3节详细介绍我们的解决方案。

3.2 和弦条件旋律变换器 (CMT)

CMT 采用类似钢琴卷帘的表示 [13, 21]，其中包含和弦、节奏和音高 (CRP) 信息。它将时间线划分为十六分音符长度的帧（即一拍的 1/4），每帧由三个向量描述：一个 12 维二进制和弦 向量（和弦中的音级取值为 1）；一个 3 维独热节奏 向量（起始时间、延音状态、休止状态）；以及一个 22 维独热旋律音高 向量（在此部分，我们将 MIDI 音高限制在 48 至 67 之间，另加一个延音状态和一个休止状态，总维度为 22）。示意图请参见 [13][图 1]。

CMT 包含三个主要模块：和弦编码器（*CE*），一个双向长短期记忆网络 [22]；节奏解码器（*RD*），由一系列自注意力模块组成；以及音高解码器（*PD*），同样由一系列自注意力模块组成。在阶段 1 中，给定输入的和弦进行，由 CE 编码得到的和弦嵌入会以自回归方式输入 RD，以输出节奏嵌入，

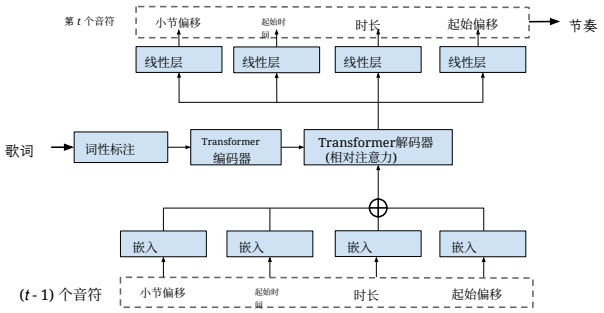


图 3：提出的歌词到节奏框架。

随后通过一个全连接层（图2中的“FC Layer”）预测整首歌曲的节奏向量序列。在阶段 2 中，和弦嵌入与节奏嵌入的拼接结果以自回归方式输入 PD，随后通过一个全连接层预测音高向量序列。最后，将节奏向量和音高向量组合并转换为旋律。

然而，要将 CMT 用于歌词到旋律任务，我们面临三个问题。(1) 多模态：CMT 的设计初衷是以和弦进行为输入来生成旋律。然而，直接添加歌词编码器来处理歌词输入并非易事，因为歌词比和弦更为复杂，是序列数据。(2)表示：CMT 使用类似钢琴卷帘的表示方式（即 CRP）对旋律进行编码，其中时间轴按固定间隔均匀划分（例如 1/16 拍），因此一个音符可能需要多个词元来表示其时长。这使得难以建立一一对应关系，将一个音节（或字符）与单个音符词元关联起来。(3) 长度限制：在 CMT 中，和弦编码器以分段生成方式生成旋律（例如每次生成 8 小节），而不利用整首歌曲和弦进行的全局上下文。然而，我们认为，输入歌词所包含的结构信息对于确定输出旋律的重复模式至关重要。下一小节将详细说明我们如何解决这些问题：(1) 通过词性标注器紧凑地编码有用的歌词信息；(2) 和 (3) 则通过采用复合词表示来解决。

3.3 歌词到节奏框架

图 3 展示了我们的歌词到节奏框架，它类似于一项语言翻译任务：即将输入的歌词序列转换为输出的音符序列。为简化处理，我们假设每个音节（或字符）都映射到一个音符；如何处理一字多音仍是未来需要解决的挑战。为此，我们采用了编码器-解码器 Transformer 架构 [12]。为增强对音符序列中重复连贯性的建模，我们引入了相对自注意力 [23, 24]。

为了提取歌词的特征，我们使用 Transformer 编码器进行词性（*POS*）标注。遵循先前的研究 [25, 26], ，我们用一个元组（小节_偏移、位置、时长、起始时间_偏移）来表示音符的节奏特征，并对这些元组的序列进行建模，

词元名称	词汇量。	描述
bar_移位	0, 1, 2	当前小节的时间偏移
起始时间	0 – 15	当前小节内以四分之一拍为单位的起始时间
时长	0 – 31	以四分之一拍为单位的时长
起始时间_偏移	0 – 15	相对于前一个音符起始时间、以四分之一拍为单位的时间偏移
		音符的起始时间

表 1：音符的节奏特征。

使用复合词（*CP*）Transformer解码器[27]。歌词到节奏模块根据完整的歌词上下文以及此前已生成的音符（从第一个到第（t-1）个），以自回归方式生成第 t 个音符，并屏蔽未来音符。我们将在接下来的小节中介绍词性标签表示和 CP Transformer。

3.3.1 词性标注

在自然语言处理中，词性标注是指为文本中的每个词标注其词性的过程。词性标签的分类因语言而异，但通常包括“名词”“动词”“形容词”“副词”等。词性标签可以通过指示句子结构来补充文本信息 [28], , 因此在传统文本转语音（TTS）系统中对输入词语进行切分时发挥着重要作用 [29, 30]。

词性是词级描述符，但为了将歌词与节奏对齐，我们需要音节级描述符。（处理中文歌词时，也可以称为“字符级”，因为每个汉字都是一个音节。）因此，我们将每个词性标签与音节索引结合，创建一个词性“标记”：例如，输入英文句子“Why not tell someone”将得到：[“副词-0”、“副词-0”、“动词-0”、“名词-0”、“名词-1”],, 其中“someone”中的两个音节分别表示为[“名词-0”、“名词-1”]。

3.3.2 复合词 *Transformer*

与 [27], 提出的 CP 不同，我们不区分与音符和节拍相关的事件。相反，我们在每个复合词中都纳入四个词元（见表 1），以便为每个音节设置一组词元。根据卡拉 OK 滚动歌词数据，我们可以获取每个音节的起始时间和时长；通过追踪强拍，我们可以获得其在小节中的节拍位置。按照 [27], 的方法，四个词元中的每一个都会被转换为嵌入，随后将这些嵌入拼接起来，再输入 Transformer解码器。四个输出嵌入中的每一个都会经过线性投影，以预测与第 t个音符对应的词元值。

所有音符准备就绪后，我们按照以下步骤将其转换为 MIDI（音高未指定）：

1. 在第 0 小节、位置 0 处放置一个空音符。2. 从前一个起始时间开始，移动（小节_偏移×16+起始时间_偏移）个单位来确定起始时间。3. 通过 min(时长，下一个音符的起始时间_偏移) 设置时长。4. 重复步骤 2 和 3，直到处理完所有音符。

需要注意的是，生成 MIDI 文件时不会使用起始时间。相反，我们使用偏移后的位置来确定起始时间，从而按递增顺序放置音符。尽管如此，我们认为起始时间有助于训练中的正则化。使用 CP 表示可以解决第 3.2 节中提到的“长度约束”问题，因为它能以更紧凑的词元序列对更长的时长进行建模，例如完整歌曲。

4. 系统配置

本节介绍我们如何训练歌词到节奏模型和节奏到旋律模型。对于每个模型，我们都会说明所用数据及其收集方式。我们以中国流行歌曲验证系统，但由于词性和音节是广泛适用的概念，该框架也可适用于其他语言。

4.1 歌词到节奏模型

我们采用与 [31] 和 [7][附录 A]类似的流程，收集了 4.5 万首中国流行歌曲的数据。具体而言，我们从网上爬取配对的歌词和音频，其中的时间戳标示了每行歌词的起始时间。然后，对每首歌曲执行以下步骤：(1) 使用声源分离提取人声音频；(2) 将歌词转换为音素序列；(3) 使用强制音素对齐估算音素起始时间戳；(4) 估算拍号、拍和强拍时间。根据音素和节拍数据，我们可以推导出音节起音，从而得到模型所需的小节偏移、起始时间、时长和起始时间偏移属性。这些步骤分别使用了与 [7]中所用工具相当的内部工具：Spleeter [32], Phonemizer [33], 蒙特利尔强制对齐器 [34], 和 Madmom [35],。

我们保留了检测到拍号为 4/4 的歌曲（约占 90%），并以四分拍为单位量化每个音节的时间戳。自动歌词对齐、节拍跟踪以及拍号检测中的误差都会降低模型质量，因此我们选取了 330 首歌曲，手动调整其时间戳。该子集用于微调模型。

在进行词性标注时，我们采用了结巴分词[1]，这一支持 56 种中文常用标签的开源工具。不使用词性标签时，我们的数据集词汇量为 5,368 个不同的字符。将字符简化为词性标签，再加入音节索引后，得到包含 123 个不同词性标记的词汇表。在第 5.2 节中，我们将比较使用这个 123 维词性向量的模型与系统的消融版本；后者用 5368 维向量对原始字符索引进行编码。

在中国流行歌曲中，文本结构的对称表达通常体现为旋律重复。图 4 展示了一个例子：张学友的《吻别》[2] 副歌旋律。实线框标出的两个短语旋律相同，文本也几乎完全相同；但即使歌词不同（虚线框内），

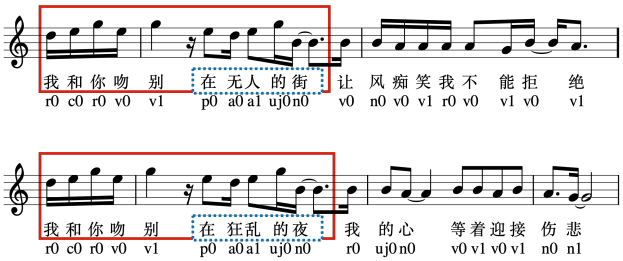


图4：《Goodbye Kiss》副歌部分的旋律-歌词-词性示例。词性缩写对照：{‘r’：代词，‘c’：连词，‘v’：动词，‘p’：介词，‘a’：形容词，‘n’：名词，‘uj’：助词}。

它们的词性标签也相同。借助词性标签表示，我们认为，如果两个文本短语具有共同的结构，歌词到节奏模型就能学会为它们生成相似的节奏。

我们为编码器-解码器 Transformer 使用以下参数：输入长度为 1000；注意力头数、编码器层数和解码器层数分别为 8、6 和 6；歌词、小节_-移位、起始时间、时长和起始时间_移位的嵌入维度分别为 512、32、128、128 和 128；dropout 为 0.1；批量大小为 16；使用 Adam 优化器，学习率为 1e-5。使用单块 Tesla-V100-SXM2-32GB GPU，在自动对齐的数据集上训练出令人满意的模型需要 ∼10 小时，在人工标注的子集上则需要 1.5 小时。两种情况下，我们都使用数据集的 10% 进行验证。

4.2 节奏到旋律模型

为训练节奏到旋律模型，我们使用了 POP909[36] 和 Lead-Sheet-Dataset [37]。POP909 包含 909 首中国流行歌曲的数据，包括和弦、MIDI 格式的旋律，以及本文未使用的其他信息。Lead-Sheet-Dataset（LSD）是一个从 HookTheory[3] 收集符号内容的数据集，包含 16K 个歌曲片段的主旋律谱（即旋律和和弦），格式为 MIDI。我们使用了四四拍歌曲（舍弃了约 10% 的数据），并将所有乐曲转调至 C大调或 A小调。最终，POP909 提供了约 30K 小节的音乐，LSD 则提供了约 40K 小节。

我们遵循 [13] 训练模型，将输入长度设为 8 小节，并将音高范围从 48 缩小至 20；超出范围的音高均移八度至范围内。从歌词到节奏模块获取完整歌曲的节奏 MIDI 后，我们以自回归方式生成 8 小节的音高，并使用 4 小节的滑动窗口，即模型根据已创作的前 4 小节来创作接下来的 4 小节。

5. 评估

我们希望回答两个问题：首先，我们的系统能否成功模拟训练数据的基本音乐特征？其次，它能否创作出悦耳、可行的

[1] https://github.com/fxsjy/jieba[2] https://www.youtube.com/watch?v=bJRkEmrkIO4

[3] https://www.hooktheory.com/

歌词谱曲方案？为回答第一个问题，我们将模型生成的旋律（记作“流行旋律”）与留出的训练数据进行比较，并探讨二者的相似之处。为回答第二个问题，我们进行了聆听测试，请参与者评估我们模型的歌词谱曲质量，以及当前最先进的替代方案 TeleMelody [7] 的歌词谱曲质量。

5.1 客观结果

我们从两个客观评估方面分析系统创作的旋律。首先，展示模型生成的节奏与原始数据有多相似（见图5）；其次，探究并描述两个模型生成的旋律之间的差异（见图6）。我们比较了数据集中的乐曲以及两个系统生成的乐谱在多种音乐量上的统计数据。为进行比较，我们分别用每个系统生成了400份乐谱，并从数据集中取用了同等数量的乐谱。

我们计算的大多数音乐量都改编自 [38]和 [39]。事实证明，将这些符号描述符融入认知上合理的音乐预测系统中，有助于增强旋律预期。研究表明，预期和记忆性是识别合理旋律的重要特征，而惊奇度和重复则是与这些特征相关的可测量要素。（有关这类描述符，以及音乐愉悦感中可预测性与不确定性概念的更多背景信息，请参见 [40,41]。）

我们展示两组描述符，分别对应两条评估线。第一组包括：旋律音符的时长；它们的起音间隔（IOI，即某个音符的起始时间与前一个音符起始时间之间的间隔）；以及它们在小节中的节拍位置。图 5 展示了数据集以及我们系统（标记为“流行旋律”）在微调前后（见第 4.1 节）输出的这些描述符的分布。请注意，我们将 TeleMelody 排除在此比较之外，因为它是在不同的数据集（约 11 万个样本）上训练的，因此将其与我们的训练数据进行比较并无意义。

从这些分布来看，两种模型的输出总体上与数据集中的旋律相似。不过，生成的旋律中显然短音符（0.25 个四分音符，即十六分音符）过多，因而使起音间隔的分布出现偏差。此外，就小节中的音符位置而言，数据集中可能起音位置的分布存在细微变化，但生成的数据并未体现这一点；微调前，生成数据中的该分布几乎是均匀的。

另一组描述符包括：音高轮廓，它表示旋律中的下一个音符较低（下降）、较高（上升）或保持不变的可能性；音符稀疏度，即旋律中没有音符的时间线占比（值为 0 表示旋律中没有休止符）；以及音高与和弦三和弦匹配比例，这是一种“协和度”指标，计算方法为旋律中属于伴奏和弦三和弦的音符所占比例。

这些描述符见图6，其中比较了

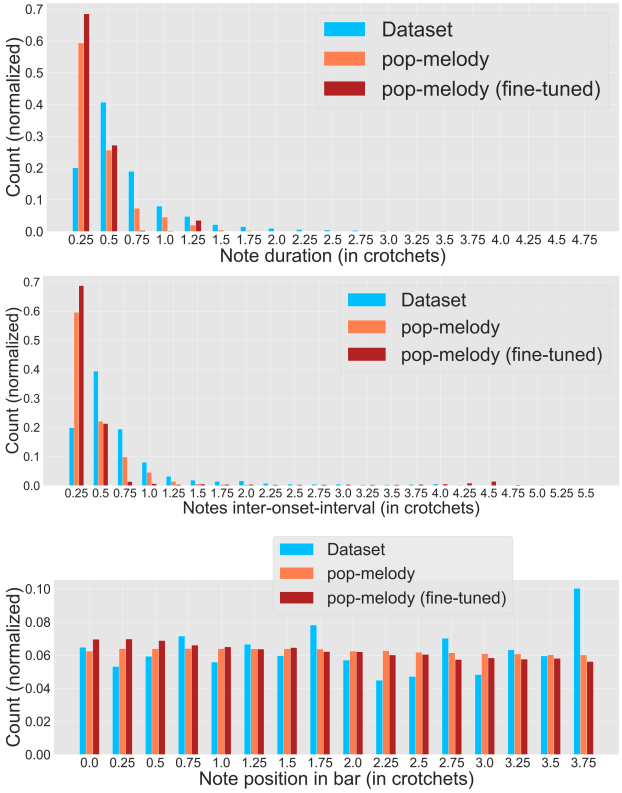


图5：数据集中的旋律与所提出的流行旋律系统生成的旋律所对应的描述符值分布。

我们微调系统生成的旋律（“流行旋律”）与 TeleMelody。这里的目的并非将系统与数据进行比较——它们是在不同数据上训练的，且各自都可能很好地反映其训练集的特点——而是评估这些系统生成的旋律有何不同。从轮廓描述符可以清楚地看出，TeleMelody 更有可能生成许多连续且音高相同的音符，而所提出系统生成的旋律则变化更多。我们的系统生成的旋律也往往休止符更少，并且包含更多出现在底层和弦中的音符。后者可以理解为一种倾向，即保持协和，并限制“不协和”或“出乎意料”时刻的空间。

5.2 主观结果

我们采用与 [3, 7, 42] 类似的设计进行了主观聆听测试：从包含 4.5 万首歌曲的数据集测试部分中随机选取十首歌曲的歌词，并将其输入三个系统以生成旋律：“TeleMelody”；“流行旋律”，即所提出的系统；以及“基线”，即我们系统的一个消融版本，它不使用词性标记（见第4.1节）。最终生成了 30 首完整歌曲：十组三首歌曲，每组三首使用相同的歌词、和弦和速度。我们使用内部开发的、可与小冰媲美的歌声合成技术 [42]将歌词和旋律渲染为音频，并根据和弦渲染了简单的伴奏。

我们共有 20 名参与者，他们都有一定的音乐背景，能够阅读乐谱并演奏乐器。



图 6：由 TeleMelody [7] 和所提出的流行旋律系统生成的旋律，其描述符值分布。

参与者每次聆听一组三首歌曲，系统身份会被隐藏，播放顺序随机。随后，他们按照李克特量表，从差(1)到优秀 (5)，根据四项标准为每首歌曲评分：

1. **节奏**：音符的时序是否适合歌词？ 2. **和声**：音高是否符合和弦与调性？ 3. **旋律**：旋律线与歌词搭配起来听起来自然吗？ 4. **整体**：旋律的整体质量如何?

为每组三项内容评分后，听众还使用 5 分制李克特量表，对自己对输入歌词所对应原歌曲的熟悉程度进行评分。此处的平均评分为 1.5，介于“1. 从未听过歌名或旋律”和“2. 听过歌名，但没听过旋律”之间。

研究结果如表2所示。总体而言，听众对三个系统的平均评分相近：均在 3.6 ± 0.25 范围内。不过，Wilcoxon 符号秩检验显示，系统之间存在细微但一致的差异；所有比较的 p 值见表3。首先，拟议系统的表现始终优于基线，这表明基于词性的分词方法是有效的。其次，我们发现拟议系统的表现也与 TeleMelody 相当或优于它；差异在节奏质量方面最为显著。尽管总体评分偏正面，大多介于一般（3）和良好（4）之间，参与者的评论大多指出了输出的不足。TeleMelody 和流行旋律都收到了一些评论，称“旋律有点奇怪”，有时还“过于重复”；但只有 TeleMelody 的输出收到评论称“节奏有点奇怪”且“支离破碎”。

系统	节奏	和声	旋律	总体
基线	3.42(.78)	3.67(.75)	3.46(.81)	3.42(.67)
TeleMelody	3.58(.82)	3.69(.83)	3.38(.72)	3.57(.58)
流行旋律	3.84(.72)	3.87(.69)	3.64(.70)	3.68(.57)

表2：主观结果与比较。

比较	节奏	和声	旋律	总体
流行音乐 vs Tele	0.0006	0.007	0.001	0.1
流行音乐 vs 基线	1.1e-08	0.002	0.006	1.8e-05
Tele vs 基线	0.01	0.89	0.22	0.02

表 3：主观结果比较的 P值。

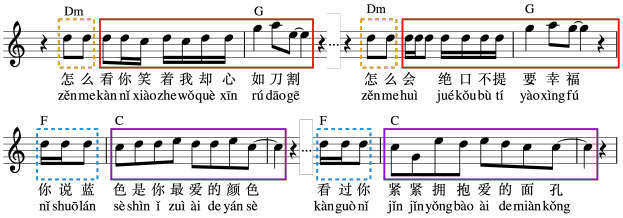


图 7：输出示例，上方为 (a)，下方为 (b)。

图7(a)和图7(b)展示了我们系统生成的两个输出示例。在这两种情况下，旋律都遵循输入和弦进行；当歌词结构重复出现时，我们可以发现旋律中的平行性和变化。例如，在图7(a)中，相似的歌词以相同的两个旋律音符开头（虚线框），其余部分则具有相似的节奏和轮廓（实线框）。同样，在图7(b)中，相似的歌词有相同的开头（虚线框），后续部分的节奏也完全一致（实线框）。

6. 结论与未来工作

本文提出了一种新方法，通过结合歌词到节奏和节奏到旋律模块，为给定歌词生成旋律。我们发现，听众认为我们的系统生成的长篇歌词谱曲效果可以接受，且至少不逊于另一款竞争系统。

为了实现从音节到起音再到旋律音符的跨模态映射，我们简化地假设每个音节都由一个音符演唱。改进歌词到节奏模块的一种显而易见的方法，是在复合词中添加音节状态标记，以指示当前处于音节的起始时间还是延续部分。不过，允许音节到音符的一对多映射也会使自动音节对齐步骤更加复杂，从而让人工校正数据变得更加珍贵。

我们还发现，词性标签是有价值的文本标记；使用它们提升了歌词谱曲的质量。鉴于这一成果，我们应当利用更多语言信息，例如音节重音和词频，如 [15, 16]所示。音乐结构标签（如主歌和副歌）也可能很有价值。这是一个尚未得到充分探索的领域，但随着更多数据集的引入，或借助自动标注系统 [43]进一步扩充现有数据，这一领域或许会变得切实可行。

7. 参考文献

[1] P. Pasquier、A. Eigenfeldt、O. Bown 和 S. Dubnov，“音乐元创作简介”，《*Computers in Entertainment (CIE)*》，第 14 卷第 2 期，第 1–14 页，2017 年。

[2] Y. Yu, A. Srivastava 和 S. Canales，《基于歌词的条件 LSTM-GAN 旋律生成》，《*ACM* 多媒体计算、通信与应用汇刊》*(TOMM)*，第 17 卷，第 1 期，第 1–20 页，2021 年。

[3] Z. Sheng、K. Song、X. Tan、Y. Ren、W. Ye、S. Zhang 和 T. Qin，《Songmass：通过预训练和对齐约束自动创作歌曲》，发表于*AAAI* 人工智能会议论文集，第 35 卷，第 15 期，2021 年，第 13 798–13 805 页。

[4] H. Bao、S. Huang、F. Wei、L. Cui、Y. Wu、C. Tan、S. Piao 和 M. Zhou，《基于歌词的神经旋律创作》，发表于*CCF* 自然语言处理与中文计算国际会议。Springer，2019 年，第 499–511 页。

[5] H.-P. Lee、J.-S. Fang 和 W.-Y. Ma，《iComposer：一种用于中文流行音乐的自动歌曲创作系统》，载于北美计算语言学协会年会论文集，2019 年，第 84–88 页。

[6] A. Ramesh、M. Pavlov、G. Goh、S. Gray、C. Voss、A. Radford、M. Chen 和 I. Sutskever，《零样本文本到图像生成》，载于第 *38* 届国际机器学习会议论文集，《机器学习研究论文集》，M. Meila 和 T. Zhang 编，第 139 卷。PMLR，2021 年 7 月 18–24 日，第 8821–8831 页。

[7] Z. Ju、P. Lu、X. Tan、R. Wang、C. Zhang、S. Wu、K. Zhang、X. Li、T. Qin 和 T.-Y. Liu，《TeleMelody：采用基于模板的两阶段方法进行歌词转旋律生成》，*arXiv* 预印本 *arXiv:2109.09617*，2021 年。

[8] K. Chen、C.-i. Wang、T. Berg-Kirkpatrick 和 S. Dubnov，“MusicSketchNet：通过音高和节奏的因子化表示实现可控音乐生成”，载于*ISMIR* 论文集，2020 年，第 77–84 页。

[9] S. Dai、Z. Jin、C. Gomes 和 R. B. Dannenberg，“通过分层音乐结构表示实现可控深度旋律生成”，载于*ISMIR* 论文集，2021 年，第 143–150 页。

[10] K. Chen、G. Xia 和 S. Dubnov，“通过解耦短期表示和结构条件实现连续旋律生成”，载于*2020* 年第 *14* 届 *IEEE* 国际语义计算会议（*ICSC*）。IEEE，2020 年，第 128–135 页。

[11] H. H. Tan，《ChordAL：一种使用双向长短期记忆网络的基于和弦的音乐生成方法》，发表于计算创造力国际会议论文集（*ICCC*），2019年，第364–365页。

[12]A. Vaswan ¡，N. S hazeer，N. Parmar，J. Usz korei t、L. Jones、A. N. Gomez、Ł. Kaiser 和 I. Polosukhin，“Attention Is All You Need”，神经信息处理系统进展，第 30 卷，2017 年。[13] K. Choi、J. Park、W. Heo、S. Jeon 和 J. Park，“基于 Transformer 解码器的和弦条件旋律生成”，*IEEE Access*，第 9 卷，第 42 071–42 080 页，2021 年。

[14] E. Nichols、D. Morris、S. Basu 和 C. Raphael，《流行音乐中歌词与旋律的关系》，载于*ISMIR*论文集，2009年，第471–476页。

[15] S. Fukayama、K. Nakatsuma、S. Sako、T. Nishimoto 和 S. Sagayama，《利用日语韵律自动创作歌曲》，载于第7届声音与音乐计算会议（*SMC*）论文集，2010年，第299–302页。

[16] E. Nichols，《基于歌词的节奏建议》，载于国际计算机音乐会议论文集，2009年。

[17] I. J. Goodfellow，《NIPS 2016教程：生成对抗网络》，*arXiv*预印本 *arXiv:1701.00160*，2016年。

[18] X. Ma、Y. Wang、M.-Y. Kan 和 W. S. Lee，“AI-Lyricist：生成音乐和词汇受限的歌词”，载于第 *29* 届 *ACM* 多媒体国际会议论文集，2021 年，第 1002–1011 页。

[19] J.-P. Briot、G. Hadjeres 和 F. D. Pachet，音乐生成的深度学习技术。Springer Cham，2020 年。

[20] L.-C. Yang、S.-Y. Chou 和 Y.-H. Yang，“MidiNet：一种用于符号域音乐生成的卷积生成对抗网络”，载于*ISMIR* 论文集，2017 年，第 324–331 页。

[21] R. Yang、D. Wang、Z. Wang、T. Chen、J. Jiang 和 G. Xia，“通过潜在表示解耦实现深度音乐类比”，载于 *ISMIR* 论文集，2019 年，第 596–603 页。

[22] A. Graves、S. Fernández 和 J. Schmidhuber，《用于改进音素分类与识别的双向 LSTM 网络》，载于国际人工神经网络会议。Springer，2005 年，第 799–804 页。[23] P. Shaw、J. Uszkoreit 和 A. Vaswani，《具有相对位置表示的自注意力》，载于计算语言学协会北美分会会议论文集，2018 年，第 464–468 页。[24] C.-Z. A. Huang、A. Vaswani、J. Uszkoreit、N. Shazeer、I. Simon、C. Hawthorne、A. M. Dai、M. D. Hoffman、M. Dinculescu 和 D. Eck，《音乐Transformer》，*arXiv* 预印本 *arXiv:1809.04281*，2018 年。[25] M. Zeng、X. Tan、R. Wang、Z. Ju、T. Qin 和 T.-Y. Liu，《MusicBERT：通过大规模预训练实现符号音乐理解》，*arXiv* 预印本 *arXiv:2106.05630*，2021 年。

[26] Y.-H. Chou、I. Chen、C.-J. Chang、J. Ching、Y.-H. Yang等，《MidiBERT-Piano：用于符号音乐理解的大规模预训练》，*arXiv* 预印本 *arXiv:2107.05223*，2021 年。[27] W.-Y. Hsiao、J.-Y. Liu、Y.-C. Yeh 和 Y.-H. Yang，《复合词 Transformer：在动态有向超图上学习创作整首歌曲音乐》，载于*AAAI* 人工智能会议论文集，第 35 卷，第 1 期，2021 年，第 178–186 页。[28] M. Sun 和 J. R. Bellegarda，《改进文本转语音合成的词性标注》，载于*IEEE* 声学、语音与信号处理国际会议论文集（*ICASSP*）。IEEE，2011 年，第 5384–5387 页。[29] D. Jurafsky 和 J. H. Martin，《语音与语言处理：自然语言处理、计算语言学与语音识别导论》，第 2 版。Prentice Hall，2008 年。[30] H. Zen、T. Nose、J. Yamagishi、S. Sako、T. Masuko、A. W. Black 和 K. Tokuda，《基于隐马尔可夫模型的语音合成系统（HTS）2.0 版》，载于*SSW*，2007 年，第 294–299 页。[31] L. Xue、K. Song、D. Wu、X. Tan、N. L. Zhang、T. Qin、W.-Q. Zhang 和 T.-Y. Liu，《DeepRapper：基于押韵与节奏建模的神经网络说唱生成》，载于计算语言学协会与自然语言处理国际联合会议论文集，2021 年，第 69–81 页。[32] R. Hennequin、A. Khlif、F. Voituret 和 M. Moussal-lam，《Spleeter：一种使用预训练模型的快速高效音乐源分离工具》，开放源代码软件期刊，第 5 卷，第 50 期，第 2154 页，2020 年。[33]M. Bernard 和 H. Titeux，《Phonemizer：使用 Python 将文本转写为多种语言的音素》，开放源代码软件期刊，第 6 卷，第 68 期，第 3958 页，2021 年。[34] M. McAuliffe、M. Socolof、S. Mihuc、M. Wagner 和 M. Sonderegger，《蒙特利尔强制对齐器：使用 Kaldi 进行可训练的文本-语音对齐》，载于 *Interspeech* 会议论文集，2017 年，第 498–502 页。[35] S. Böck、F. Korzeniowski、J. Schlüter、F. Krebs 和 G. Widmer，《madmom：一个新的 Python 音频与音乐信号处理库》，载于第 *24* 届 *ACM* 多媒体国际会议论文集，2016 年 10 月，第 1174–1178 页。[36] Z. Wang、K. Chen、J. Jiang、Y. Zhang、M. Xu、S. Dai、X. Gu 和 G. Xia，《Pop909：用于音乐编配生成的流行歌曲数据集》，载于*ISMIR* 会议论文集，2020 年，第 38–45 页。[37] Y.-C. Yeh、W.-Y. Hsiao、S. Fukayama、T. Kita-hara、B. Genchel、H.-M. Liu、H.-W. Dong、Y. Chen，T. Leong 和 Y.-H. Yang，“基于三和弦的自动旋律和声编配：一项比较研究”，《新音乐研究杂志》（*JNMR*），第50卷第1期，第37–51页，2021年。[38] D. Conklin 和 I. H. Witten，“用于音乐预测的多视角系统”，《新音乐研究杂志》，第24卷第1期，第51–73页，1995年。[39] M. T. Pearce，“音乐感知与作曲中旋律结构统计模型的构建与评估”，博士学位论文，伦敦城市大学，2005年。[40] B. P. Gold、M. T. Pearce、E. Mas-Herrero、A. Dagher 和 R. J. Zatorre，“音乐愉悦感中的可预测性与不确定性：学习的奖励？”，《神经科学杂志》，第39卷第47期，第9397–9409页，2019年。[41]D. Müllensiefen，“Fantastic：特征分析技术访问统计数据（语料库中）：技术报告v1”，英国伦敦：伦敦大学金匠学院。取自 *http://www.doc.gold.ac.uk/isms/m4s*，2009年。[42] H. Zhu、Q. Liu、N. J. Yuan、C. Qin、J. Li、K. Zhang、G. Zhou、F. Wei、Y. Xu 和 E. Chen，“小冰乐队：流行音乐旋律与编曲生成框架”，载于第*24*届 *ACM SIGKDD*知识发现与数据挖掘国际会议论文集，2018年，第2837–2846页。[43] J.-C. Wang、Y.-N. Hung 和 J. B. L. Smith，“捕捉副歌、主歌、前奏或其他部分：运用结构功能分析歌曲”，载于*IEEE*国际声学、语音与信号处理会议（*ICASSP*）论文集，2022年，第416–420页。