

DOI: 10.11992/tis.202001006

基于序列模型的音乐词曲匹配度智能评估算法

陈壮豪¹, 张茂清¹, 郭为安², 康琦¹, 汪镭¹

(1. 同济大学 电子与信息工程学院, 上海 201804; 2. 同济大学 中德工程学院, 上海 201804)

摘要: 情感匹配模型是一种常用于评价词曲匹配程度的方法; 然而, 单纯地依靠情感匹配模型无法对评价词曲匹配度进行准确的评价。为改善此问题, 提出了基于序列模型的音乐词曲匹配度智能评估算法, 其综合考虑词曲情感和词曲间节奏关系以给出一个更加准确的词曲评估方法。基于公开词曲同步数据集, 构建了音乐情感和节奏正反例模型, 并基于此模型将音乐切分成片段; 进一步, 将歌词和旋律片段分别通过歌词编码器和旋律编码器进行编码, 并将编码后具有上下语境的歌词特征和旋律特征输入词曲匹配解码器, 解析词曲间特征关系, 判断词曲片段匹配程度。仿真结果表明: 基于序列模型的音乐词曲匹配度智能评估算法, 相对于单纯的情感匹配模型, 能够更精确地评价词曲匹配程度, 验证了本文提出算法的有效性。

关键词: 音乐词曲; 情感; 节奏; 序列模型; 歌词编码器; 旋律解码器; 词曲匹配解码器; 词曲匹配度;

中图分类号: TP393.04 **文献标志码:** A **文章编号:** 1673-4785(2020)01-0067-07

中文引用格式: 陈壮豪, 张茂清, 郭为安, 等. 基于序列模型的音乐词曲匹配度智能评估算法 [J]. 智能系统学报, 2020, 15(1): 67-73.

英文引用格式: CHEN Zhuanghao, ZHANG Maoqing, GUO Weian, et al. Music lyrics-melody intelligent evaluation algorithm based on sequence model[J]. CAAI transactions on intelligent systems, 2020, 15(1): 67-73.

Music lyrics-melody intelligent evaluation algorithm based on sequence model

CHEN Zhuanghao¹, ZHANG Maoqing¹, GUO Weian², KANG Qi¹, WANG Lei¹

(1. College of Electronics and Information Engineering, Tongji University, Shanghai 201804, China; 2. Sino-German College of Applied Sciences, Tongji University, Shanghai 201804, China)

Abstract: Emotional matching model is a method often used to evaluate the degree of lyrics and melody matching. However, it cannot be accurately evaluated based on the emotion matching model. In order to improve it, this paper proposes an intelligent evaluation algorithm of lyrics-melody matching based on a sequence model, which comprehensively considers the emotion and the rhythm relationship between lyrics and melody to give an evaluation method for more accurate evaluation. Firstly, this paper researches and builds music positive and negative samples considering music emotion and phrase based on the public lyrics-melody paired dataset and divide songs to music pieces. Further, the lyrics and melody fragments are encoded by the lyrics-encoder and the melody-encoder, respectively. And take the encoded lyrics feature and melody feature that are contextualized as the input of the lyrics-melody matching decoder to analyze the characteristic relationship between the lyrics and melody, and then determine the matching degree of the lyrics-melody segment. The experimental results show that the music lyrics-melody matching intelligent evaluation algorithm model based on sequence model can more accurately judge the matching degree of lyrics-melody matching than simple music emotion matching, which verifies the effectiveness of the proposed algorithm.

Keywords: music lyrics-melody; emotion; rhythm; sequence model; lyrics encoder; melody encoder; matching decoder; lyrics-melody matching degree; music lyrics-melody matching

收稿日期: 2020-01-06.

基金项目: 国家自然科学基金面上项目 (51775385, 71371142);
国家自然科学基金项目 (71771176, 61503287).

通信作者: 汪镭. E-mail: wanglei@tongji.edu.cn.

音乐词曲匹配度评估是针对音乐的字曲匹配程度给出的一个客观评估。随着智能音乐研究发展, 越来越多智能音乐作品产生, 随之而来的便

是对生成的音乐作品评价问题。

目前对音乐作品的评价方式主要包括以下几种方式: 1) 基于互动的评价方式, 这种评价方式一般依赖于听者对一般音乐的反馈, 通过收集听者对生成音乐作品的反应, 如心跳、脉搏和皮肤电导, 如评估分数、偏好或生理信号等; 然后使用收集到的信息来评估生成的音乐作品。由于音乐作品的主观性, 这种交互式评价方法被广泛使用。文献 [1] 根据用户对音乐流派的选择, 使用马尔可夫链和歌词数据库创建歌词, 而音乐则由旋律和该流派的 3 个伴奏之一组成。而词曲的组合则通过人工的方式由用户来决定。2) 在基于交互评价的基础之上, 通过与音乐家合作获得专业反馈。文献 [2] 通过选择专业人士和非专业人士若干, 综合评价为两方评价均分的加权求和。3) 基于规则的评价方法, 使用显式的评价规则对生成的音乐作品进行评价。评价规则是根据个人的创作经验或音乐理论, 综合考虑节奏、词组、音阶、和弦等音乐要素而制定的。如文献 [3] 中采用了音乐理论中考虑跳跃和间隔的一些评价规则对音乐作品进行评价。这种方式对于创作特定音乐流派的音乐作品是有效果的。4) 基于学习的评价方法, 文献 [4] 利用神经网络构建了音乐情感分类及评估模型。

本文采用的是基于学习的评价方式, 随着自然语言处理技术的发展和运用, 通过算法模型对音乐词曲进行情感分析的准确性逐渐增加。然而对旋律或歌词进行单独的情感分析是不准确的, 主要包含两点: 1) 缺少旋律衬托的歌词无法确定作曲者要突出的内容, 从而对情感的分析是不准确的; 2) 在没有旋律背景下的歌词, 容易忽略歌词中的节奏感, 而节奏感在音乐中, 对情感表达是非常重要的。而现有的评价方式却未充分考虑上述问题。

因此本文尝试综合考虑词曲的情感和词曲间的节奏关系来给出一个尽量准确且客观的智能评估。而要学习到词曲间的情感和节奏特征, 则需要研究设计合理的音乐表达形式, 才能从中学习到情感及节奏上的特征, 因此本文将主要从音乐数据表达形式研究及算法模型设计两个方面进行阐述。

1 音乐数据表达形式

要从音乐数据中能学习到词曲之间的情感关系和节奏关系, 则需要研究并构建情感、节奏正反例音乐样本, 音乐反例构建的好坏决定了模型

是否可以通过正反例之间的差异挖掘到词曲之间匹配的特征关系。若反例无法充分将不匹配的关键因素表达出来, 则难以通过词曲匹配算法区分当前音乐词曲是否匹配。

1.1 音乐情感反例构建

由于本文是基于序列模型进行词曲匹配度评估, 是基于音乐序列的分析, 所以在音乐情感模型上, 采用离散的音乐情感模型, 提出了简化的 Hevner^[5] 情感分类模型, 并结合国人的审美标准, 对关键词进行适当修改。如图 1 所示, 建立了音乐情感模型, 从情感角度将音乐分成四大类。相邻类别具有一定的情感相似性, 情感可以在相邻类别上转换。相对两个情感类别表达情感内涵相反。

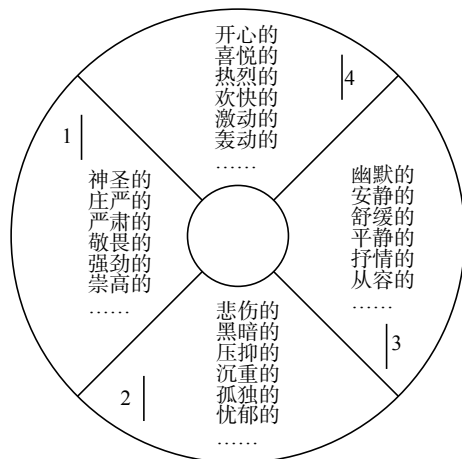


图 1 音乐情感模型

Fig. 1 Music emotion model

基于提出的 CZH 音乐情感模型。对公开的词曲同步音乐数据集^[6] 根据情感进行划分。在情感标定的过程中, 尽可能使得每一类情感音乐都具有鲜明的情感表达, 即当听欢快的音乐时, 能很容易区分该音乐为欢快的音乐, 而不会模棱两可。该做法的目的旨在为后续构建音乐情感反例打下基础。基于此基准, 一共标定 1 275 首乐曲。如表 1 所示。

表 1 音乐情感分类数目统计
Table 1 Music emotion classification statistic

类别	类别1	类别2	类别3	类别4
数目	299	306	345	325

进一步, 本文在歌词数相当的情况, 将情感表达相反的两种情感音乐, 进行词曲交换, 作为音乐情感反例。如图 2 为音乐情感反例。图中两个片段分别来自音乐剧《悲惨世界》中的《On my own》以及欢快的音乐《Boogie Down》构建的反例。



图 2 情感反例展示

Fig. 2 Negative music example

可以发现, 其实在进行情感音乐词曲交换后, 反例音乐一方面是由于情感不一致, 而导致不匹配。但更具体的体现为歌词和旋律的整体节奏产生了一致。这种不一致, 使得词曲原先相互呼应的情感关系被破坏, 从而导致词曲不匹配。换句话说, 情感很大程度上还是通过节奏表现出来的。比如存在一些音乐, 歌词在没有旋律的情况下, 仅理解歌词含义, 可能表达的是悲伤的情感, 但是结合旋律以后, 会发现其实这首音乐听起来是欢快的。仔细探究, 便会发现其实歌词中, 通过大量的有节奏性的重复操作, 使得整首歌的节奏感很强, 因此不会透露出悲伤的情绪。

1.2 音乐节奏反例构建

对于一首音乐, 节奏至关重要。其中歌词和旋律常在节奏上, 如发音的音调上; 或者是对于短且快的旋律片段常对应发音可以较短的歌词, 而不是复杂的歌词; 在相同的旋律片段下, 对应的歌词也具有足够的相似性, 即在节奏上与歌词保持一致等。本文通过歌词乱序的方法将原有的节奏联系打破, 从而实现节奏反例构建。

而要对歌词进行乱序操作, 就必须确定乱序的粒度, 即以歌词片段、以单词或者以词块为最小粒度进行乱序。以《on my own》歌曲为例, 如图 3~5 所示为 3 种乱序方式。

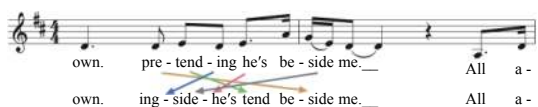


图 3 以词块为最小粒度的乱序

Fig. 3 Shuffle in word-piece level



图 4 以单词为最小粒度的乱序

Fig. 4 Shuffle in word-level



图 5 以歌词片段为最小粒度的乱序

Fig. 5 Shuffle in sentence

首先对于以词块为最小粒度的乱序会存在破坏基本单词结构, 比如连续唱出词块 *ing*。这样的

歌词序列是不合理的, 因此不应通过该方式进行乱序操作。其次对于以单词为最小粒度的乱序, 这种方式相对词块方式更加合理, 这种方式会破坏句子语意。同时表达了节奏上的不匹配和语意上不匹配。所以可以作为打乱的最小粒度。最后对于以歌词片段为最小粒度的乱序, 片段长度设定为尽可能包含一个句子的长度, 因此该乱序方式, 最大程度地保留了语意上的特征。同时表达了节奏上的不匹配。其构建算法流程如下:

算法 1 音乐歌词乱序算法

输入 m 首输入音乐的输入歌词 $L^{(i)}, i \in (0, m), L^{(i)} = \{L_0^{(i)}, L_1^{(i)}, L_2^{(i)}, \dots, L_n^{(i)}\}$, 乱序方式 $\text{shuffle_level} \in \{\text{WORD}, \text{SENTENCE}\}$ 。

参数 输入音乐歌词长度 $n = \text{len}(L)$, 歌曲片段长度 $l = \text{len}(L^{(i)})$, 随机种子 seed 。

输出 乱序后的音乐歌词。

流程 对整体的迭代次数 $i = 0, 1, 2, \dots, m$:

1) 生成随机数 seed , 作为当前歌曲乱序的随机种子。

2) 根据乱序标志符 shuffle_level 决定对当前音乐样本进行词级别还是片段级别乱序。

3) 对于单首音乐片段个数, 循环次数 $j = 0, 1, 2, \dots, \text{length}$:

① if $\text{shuffle_level} = \text{SENTENCE}$:

a. 要交换音乐歌词片段的索引:

$k = \text{int}(\text{random}(\text{seed}) * (j + 1))$

b. 交换两个音乐片段:

$\text{swap}(L_j^{(i)}, L_k^{(i)})$

② if $\text{shuffle_level} = \text{WORD}$:

for v in range(l):

a. 此片段要交换单词的索引:

$k = \text{int}(\text{random}(\text{seed}) * (v + 1))$

b. 交换此片段中的两个单词:

$\text{swap}(L_j^{(i)}(v), L_j^{(i)}(k))$

1.3 音乐反例构建结果

基于已情感标定的 1 275 首正例音乐, 并利用上述两种音乐反例构建方法, 随机选择其中一部分的音乐用于音乐反例构建。随机选择主要是为了避免模型在特定样本中学到过多额外的特征, 使得泛化性能降低。如表 2 所示, 其中音乐情感反例为 1 184 首, 音乐节奏反例为 1 175 首, 共计 2 359 首反例音乐。

表 2 音乐反例数目统计

Table 2 Negative music statistics

反例类别	音乐情感反例	音乐节奏反例	反例总和
数目/首	1 184	1 175	2 359

1.4 音乐歌词及旋律表达形式

由于本文使用的是英文的音乐数据集,而英文是表音的,在音乐中,单词的后缀发音等对音乐本身的节奏是有影响的。因此对于音乐中的歌词,为了更好地捕捉歌词的含义及发音。采用分布式形式表示^[7],同时将单词拆分为更小的词块(word piece),考虑单词编码和词块编码,充分提取英文歌词中发音的特征,即:

$$\mathbf{v}_{\text{lyrics}} = \mathbf{v}_{\text{word}} \parallel \mathbf{v}_{\text{piece}}$$

式中: \mathbf{v}_{word} 为词向量, $\mathbf{v}_{\text{piece}}$ 为词块向量。拼接得到歌词向量。如对于单词 womon, 由词块 wom, on 组成, 则: $\mathbf{v}_{\text{lyrics}}^{(j)} = \mathbf{v}_{\text{wom}} + \mathbf{v}_{\text{womon}}$, 其中 $\mathbf{v}_{\text{lyrics}}^{(j)}$ 表示音乐片段中第 j 个词块; 对于音乐旋律, 将音符转换为音高、音长、休止时长来进行表征, 则在第 j 个词块对应的音符编码向量表示为 $\mathbf{v}_{\text{melody}}^{(j)} = \{\text{pitch}, \text{duration}, \text{rest}\}$ 。

如表3所示, 为旋律编码结果。

表3 旋律编码结果

Table 3 Melody encode result

	American				womon		stay	away		from	me
	A	mer	i	can	wom	on	stay	a	way	from	me
音符	B4	A5	A5	A5	A5	G5	B4	B4	D5	E5	E5
音长	0.5	0.5	0.5	0.5	1.5	4.0	0.75	0.5	1.5	1.0	2.0
休止	0	0	0	0	0	0	1	0	0	0	1

1.5 音乐切分方法

本文将用于训练的单曲划分为多个片段, 采用滑动窗口的方法对音乐进行切分, 假设音乐长度 $N=10$, 每个窗口长短为 $l=4$, 窗口移动步长 $s=l/2=2$, 则切分效果如图6。

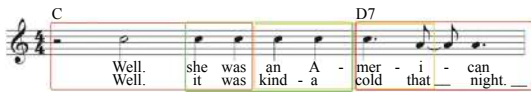


图6 输入音乐切分效果示意

Fig. 6 music segmentation diagram

通过设置合理的窗口大小, 能尽可能保证每个片段的语意完整, 且每个音乐片段长度固定。本文将窗口大小 l 设定为 20。算法流程如下:

算法2 音乐切分算法

输入 音乐 $S = M \cup L$, 且 $\text{len}(S) = N$ 。

参数 窗口大小 l , 移动步长 $s=l/2$ 。

输出 切分后的词曲片段数组 M, L 。

流程如下:

1) 计算循环次数 (乐曲片段数) n :

$$n = \begin{cases} \left\lfloor \frac{N-l-1}{s} \right\rfloor, & (N-l-1) \bmod s = 0 \\ \left\lfloor \frac{N-l-1}{s} \right\rfloor + 1, & (N-l-1) \bmod s \neq 0 \end{cases}$$

式中: $\lfloor \cdot \rfloor$ 表向下取整, MOD 表取余。

2) 对于整体循环次数 $i = 0, 1, 2, \dots, n$:

① 计算单个片段范围:

start = $i \cdot s$, end = start + l ;

② 构造旋律片段样本:

$M_i = M[\text{start}, \text{end}]$;

③ 构造旋律片段样本:

$L_i = L[\text{start}, \text{end}]$;

3) 重新构建 M, L :

$$M = \{M_0, M_1, \dots, M_n\} \quad M_i \in M$$

$$L = \{L_0, L_1, \dots, L_n\} \quad L_i \in L$$

2 词曲匹配度评估算法

通过数据处理, 得到了用于训练词曲匹配算法模型的样本数据。本节综合考虑音乐词曲情感及节奏特征, 提出基于序列模型^[8]的词曲匹配度智能评估算法。

2.1 算法原理及框图

第一章中提到, 对旋律或歌词进行单独的情感分析是不准确的, 主要包含两点: 1) 缺少旋律衬托的歌词无法确定作曲者要突出的内容, 从而对情感的分析是不准确的; 2) 在没有旋律背景下的歌词, 容易忽略歌词中的节奏感, 而节奏感在音乐中, 对情感表达是非常重要的, 正如本文提出的观点: 情感很大程度上还是通过音乐节奏表现出来的。因此本节设计的算法, 将情感和节奏两者综合考虑, 设计得到词曲匹配算法模型如图7。结合模型框图, 给出算法定义如下:

给定输入音乐 $S = M \cup L$, 其中输入旋律 $M = \{M_0, M_1, \dots, M_n\}$ 包含 n 个旋律片段, 每个输入旋律片段包含 l 个音符, 即 $M_i = \{M_i^{(0)}, M_i^{(1)}, \dots, M_i^{(l)}\}$, 同理对于输入歌词 $L = \{L_0, L_1, \dots, L_n\}$ 包含 n 个歌词片段, 每个歌词片段包含 l 个词块, 即 $L_i = \{L_i^{(0)}, L_i^{(1)}, \dots, L_i^{(l)}\}$ 。音乐词曲匹配算法通过歌词编码器和旋律编码器, 对输入旋律片段 M_i 、输入歌词片段 L_i 中

对应的词块 $L_i^{(j)}$ 和音符 $M_i^{(j)}$ 进行同步编码, 并将编码结果通过词曲匹配解码器解析、判断当前词曲匹配度, 并在解过程中, 保存更新词曲特征匹

配关系的状态信息, 实现音乐片段的词曲匹配判断, 综合输入音乐的 n 个词曲片段匹配程度, 判断当前输入音乐词曲匹配程度。

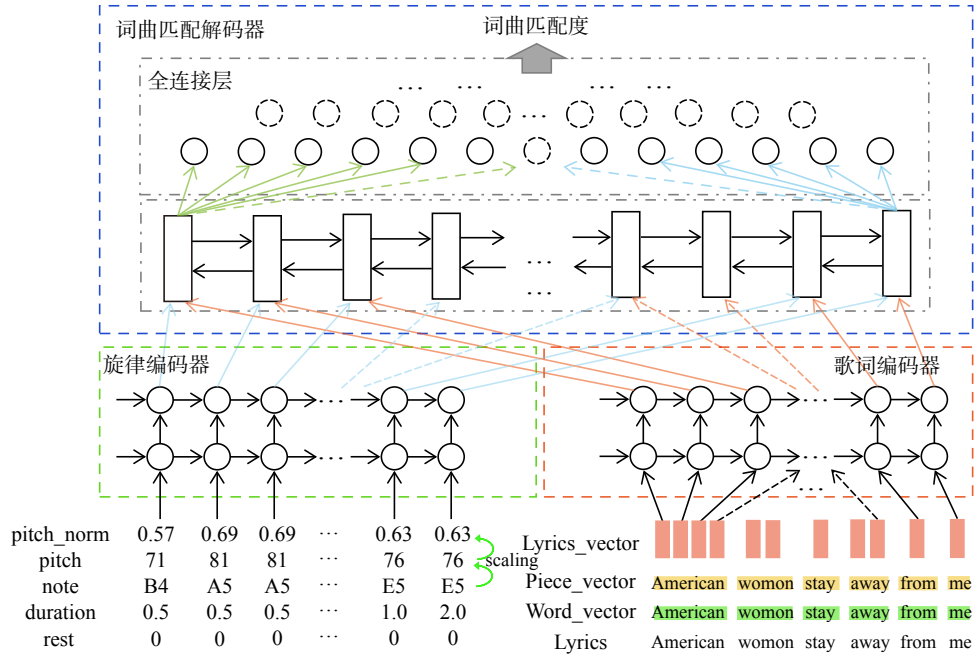


图 7 词曲匹配算法框架

Fig. 7 Music lyrics-melody matching algorithm

基于词曲匹配算法框架, 算法计算流程如下:

算法 3 词曲匹配算法

输入 旋律 M 包含 n 个旋律片段, 每个片段包含 l 个音符, $M_i^{(j)}$ 表示第 i 个输入旋律片段中的第 j 个音符的向量表示, $M_i^{(j)} = \{\text{norm}(\text{pitch}), \text{duration},$

$\text{rest}\}$; 歌词 L 包含 n 个歌词片段, 每个片段包含 l 个词块, $L_i^{(j)}$ 表示第 i 个输入歌词片段中的第 j 个歌词的向量表示为: $v_{\text{lyrics}} = v_{\text{word}} + v_{\text{piece}}$, 其中 v_{word} 为歌词词向量, v_{piece} 为歌词词块向量;

模型参数如表 4 所示。

表 4 模型参数

Table 4 Configuration of matching model

层	旋律编辑器	歌词编码器	词曲匹配度
Input	3(Pitch, Duration, Rest)	20(v_{lyrics})	128(code)
Layer1	64, LSTM	64, LSTM	128, LSTM
Layer2	64, LSTM	64, LSTM	128, LSTM, fully-connected, sigmoid
Output	64($\text{code}_{\text{note}}$)	64($\text{code}_{\text{lyrics}}$)	2 (real or fake), sigmoid

输出 输入词曲的匹配程度

流程如下:

1) 令 $X_i^{(j)} = M_i^{(j)}$, 通过循环神经网络, 即通过式 (1)~(6), 对输入旋律进行编码, 同理, 令 $X_i^{(j)} = L_i^{(j)}$, 对输入歌词进行编码。

$$\mathcal{T}_f^{(i)} = \sigma(W_f[a^{i-1}, X_i^{(j)}] + b_f) \quad (1)$$

$$\mathcal{T}_u^{(i)} = \sigma(W_u[a^{i-1}, X_i^{(j)}] + b_u) \quad (2)$$

$$\hat{c}^{(i)} = \tanh(W_c[a^{i-1}, X_i^{(j)}] + b_c) \quad (3)$$

$$c^{(i)} = \mathcal{T}_f^{(i)} \circ c^{(i-1)} + \mathcal{T}_u^{(i)} \circ \hat{c}^{(i)} \quad (4)$$

$$\mathcal{T}_o^{(i)} = \sigma(W_o[a^{i-1}, X_i^{(j)}] + b_o) \quad (5)$$

$$\text{code}^{(j)} = a^{(i)} = \mathcal{T}_o^{(i)} \circ \tanh(c^{(i)}) \quad (6)$$

式中“ \circ ”表示前后量之间对应元素相乘。

对于输入旋律片段 M_i 的第 j 个时间步 $M_i^{(j)}$ 以及输入歌词片段 L_i 的第 j 个时间步 $L_i^{(j)}$ 。可计算得到第 j 个时间步的旋律编码向量和歌词编码向量。

$$\text{code}_{\text{note}}^{(j)} = f_{\text{LSTM}}(M_i^{(j)}) \quad (7)$$

$$\text{code}_{\text{lyrics}}^{(j)} = f_{\text{LSTM}}(L_i^{(j)}) \quad (8)$$

式中: $\text{code}_{\text{note}}^{(j)}$ 表示第 j 个音符的编码向量, 为旋律编码器输出; $\text{code}_{\text{lyrics}}^{(j)}$ 表示第 j 个歌词的编码向量,

为歌词编码器输出。

2) 如式 (7)~(8), 将第 j 个时间步的旋律编码向量 $\text{code}_{\text{note}}^{(j)}$ 和歌词编码向量 $\text{code}_{\text{lyrics}}^{(j)}$ 进行拼接:

$$\text{code}^{(j)} = \begin{bmatrix} \text{code}_{\text{note}}^{(j)} \\ \text{code}_{\text{lyrics}}^{(j)} \end{bmatrix}$$

3) 将当前输入歌词及旋律片段中 l 个同步音符及词块编码完成后, 作为词曲匹配解码器输入, 通过双向 LSTM 计算得到两个隐藏状态编码, 该状态保存了词曲间的情感关系和节奏关系摘要:

$$c_{\text{head}}, c_{\text{tail}} = f_{\text{Bi-LSTM}}(\text{code}^{(0)}, \text{code}^{(1)}, \dots, \text{code}^{(l)})$$

4) 将正反向状态编码拼接, 作为全连接层输入。

$$c = \begin{bmatrix} c_{\text{head}} \\ c_{\text{tail}} \end{bmatrix} \quad (9)$$

解码器通过全连接层对词曲情感特征和节奏特征输入, 判断词曲是否匹配, 即求解:

$$p_i = P(y_i | c) = \text{sigmoid}(W_m c + b_m)$$

因此给出目标函数:

$$\mathcal{L} = -\frac{1}{m} \sum_{i=1}^m [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)]$$

式中 m 表示音乐片段个数。

5) 重复步骤 1) 至步骤 4), 直至当前音乐 m 个词曲片段判断完成。

2.2 实验过程及结果

本节主要对算法模型实现及结果进行阐述, 模型参数同表 4。其中两个编码器隐藏单元个数均设置为 64, 因此解码器输入为 128 维向量, 解码器隐藏单元个数与输入保持一致, 经过全连接层及 sigmoid 得到匹配概率值。算法模型使用零值初始化旋律编码器和歌词编码器的隐藏状态。

训练过程使用学习率为 0.01 的 Adam^[10] 优化器计算并更新反向传播梯度, 并基于式 (9) 损失函数完成迭代训练。迭代过程损失变化如图 8 所示。

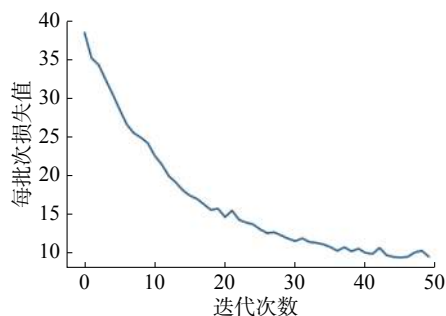


图 8 模型损失曲线

Fig. 8 Model loss curve

图 8 中, 每批次损失值为批量样本损失之和, 每个批次样本片段数 64; 迭代过程精确度变化如图 9 所示。

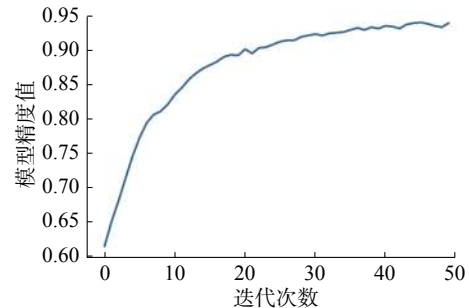


图 9 模型精度曲线

Fig. 9 Model accuracy curve

可以发现, 模型在迭代过程中是收敛的, 在训练集上的精度达到了 94.0%, 通过实验, 在测试集上的精度达到 85.9%。

表 5 为词曲匹配结果, 包含了当前词曲名称、最终匹配程度, 以及当前音乐在算法模型中被切分为多少个片段和匹配了多少个片段。同时, 展示了每个片段的匹配结果。图中展示的音乐为歌曲《I feel lonely》, 在单看旋律时, 旋律体现较为欢快的情感; 单看歌词时, 歌词则体现悲伤的情感, 但是歌词结合旋律可以发现, 该歌曲的节奏感很强, 因此整体节奏是吻合的, 是一首词曲匹配的音乐。若不将情感和节奏同时考虑, 单纯通过情感维度, 会认为该乐曲的词曲不匹配。利用本文提出的词曲匹配模型对该首音乐进行词曲评价, 可以发现当前音乐 18 个词曲片段中, 有效匹配的片段数为 16 个, 最终词曲匹配得分为 0.89, 说明算法模型可以综合考虑情感和节奏两个因素, 给出一个合理的评价。

表 5 模型结果展示
Table 5 Result of model

音乐类别	乐曲名	词曲匹配度	词曲总片段数	匹配片段数	片段匹配结果
正例	《I feel lonely》	0.888 9	18	16	[1, 1, 1, 0, 1, ..., 1, 1, 0]
乱序反例	《Travis》	0.062 5	16	1	[0, 0, 0, 0, 0, ..., 0, 1, 0]

进一步的, 随机选择一首音乐, 当对音乐打乱时, 由于词曲间节奏不一致和歌词押韵被打破, 因此词曲匹配度将降低。可以发现通过模型计算后, 在当前音乐 16 个片段中, 有效匹配的片段数仅为 1 个, 且最终匹配得分仅为 0.062 5。

3 结束语

针对情感匹配模型在评价词曲匹配度时存在评价准确度低的缺陷, 本文提出综合考虑词曲情感和词曲间节奏关系, 进一步提出了基于序列模型的词曲匹配度智能评估算法。仿真结果表明, 综合考虑音乐词曲间情感和节奏特征的音乐词曲匹配算法模型, 能给音乐作品做出更加准确的智能评估。

参考文献:

- [1] BILES J A. GenJam: a genetic Algorithm for generating jazz solos[C]//Proceedings of 1994 International Computer Music Conference. Aarhus, Denmark, 1994: 131–137.
- [2] PHON-AMNUAISUK S, TUSON A, WIGGINS G. Evolving musical harmonisation[C]//Proceedings of International Conference on Artificial Neural Nets and Genetic Algorithms. Portorož, Slovenia, 1999: 229–234.
- [3] 曹西征, 毛文涛, 乔锟, 等. 基于音高旋律元的柔和乐曲的自动作曲算法 [J]. *自动化学报*, 2012, 38(10): 1627–1638.
CAO Xizheng, MAO Wentao, QIAO Kun, et al. Automated composition algorithm for gentle music based on pitch melody unit[J]. *Acta automatica sinica*, 2012, 38(10): 1627–1638.
- [4] GIBSON P M, BYRNE J A. NEUROGEN, musical composition using genetic algorithms and cooperating neural networks[C]//Proceedings of the 1991 2nd International Conference on Artificial Neural Networks. Bournemouth, UK, 1991: 309–313.
- [5] HEVNER K. Experimental studies of the elements of expression in music[J]. *American journal of psychology*, 1936, 48(2): 246–268.
- [6] YU Yi, CANALES S. Conditional LSTM-GAN for melody generation from lyrics [J]. arXiv: 1908.05551, 2019.
- [7] HARRIS Z S. Distributional structure[J]. *Word*, 1954, 10(2/3): 146–162.
- [8] SUTSKEVER I, VINYALS O, LE Q V. Sequence to sequence learning with neural networks[C]//Proceedings of the 27th International Conference on Neural Information Processing Systems. Cambridge, USA, 2014.
- [9] HOCHREITER S, SCHMIDHUBER J. Long short-term memory[J]. *Neural computation*, 1997, 9(8): 1735–1780.
- [10] KINGMA D P, BA J. Adam: a method for stochastic optimization [J]. arXiv: 1412.6980, 2014.

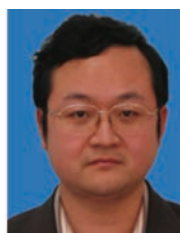
作者简介:



陈壮豪, 硕士研究生, 主要研究方向为音乐评价的自动化。



张茂清, 博士研究生, 主要研究方向为进化计算及其应用研究。已发表学术论文 10 余篇。



汪镭, 教授, 博士生导师, 主要研究方向为群体智能、并行实现技术。出版专著 4 部, 发表学术论文 90 余篇。