

分类号: 按中国图书分类法, 学位办网上可查

单位代码: 10335

密 级: 注明密级与保密期限

学 号: _____

浙 江 大 学

硕士学位论文



中文论文题目: 流行歌曲自动翻译与歌声合成研究

英文论文题目: Automatic Song Translation and

Singing Voice Synthesis on Pop Songs

申请人姓名： 李承曦

指导教师： 卜佳俊教授于智研究员

合作导师：

学科(专业)： 计算机科学与技术

研究方向： 研究方向

所在学院： 计算机科学与技术

论文递交日期 递交日期

流行歌曲自动翻译与歌声合成研究



论文作者签名: _____

指导教师签名: _____

论文评阅人 1: _____ 姓名

评阅人 2: _____ 姓名

评阅人 3: _____ 姓名

评阅人 4: _____ 姓名

评阅人 5: _____ 姓名

答辩委员会主席: _____

委员 1: _____

委员 2: _____

委员 3: _____

委员 4: _____

委员 5: _____

答辩日期 _____

Automatic Song Translation and
Singing Voice Synthesis on Pop Songs



Author's signature: _____

Supervisor's signature: _____

External reviewers: _____ Name
_____ Name
_____ Name
_____ Name
_____ Name

Examining Committee Chairperson:

Examining Committee Members:

Date of oral defence: _____

浙江大学研究生学位论文独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得 浙江大学 或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

学位论文作者签名:

签字日期:

年

月

日

学位论文版权使用授权书

本学位论文作者完全了解 浙江大学 有权保留并向国家有关部门或机构送交本论文的复印件和磁盘，允许论文被查阅和借阅。本人授权 浙江大学 可以将学位论文的全部或部分内容编入有关数据库进行检索和传播，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

(保密的学位论文在解密后适用本授权书)

学位论文作者签名:

导师签名:

签字日期:

年

月

日

签字日期:

年

月

日

勘误表

致谢

序言

摘要

Abstract

缩略词表

英文缩写	英文全称	中文全称
NLP	Natural Language Processing	自然语言处理
AI	Artificial Intelligence	人工智能
MT	Machine Translation	机器翻译
SVS	Singing Voice Synthesis	歌声合成
NMT	Neural Machine Translation	神经机器翻译

目录

勘误表.....	I
致谢	III
序言	V
摘要	VII
Abstract	IX
缩略词表	XI
目录	XIII
图目录	XV
表目录	XVII
1 绪论	1
1.1 引言	1
1.2 国内外研究现状	1
1.2.1 自动歌曲翻译国内外研究现状	1
1.2.2 歌声合成国内外研究现状	1
1.3 研究意义及内容	2
1.4 章节安排	3
1.5 章节小结	4
2 相关研究介绍	5
2.1 歌曲歌词生成及限制性翻译研究介绍	5
2.2 扩散模型相关研究介绍	5
3 自动歌曲翻译	7
3.1 歌曲翻译数据集	7
3.2 歌曲翻译数据的收集和预处理	8
3.3 自动歌曲翻译模型结构	8
3.3.1 音符表示池化嵌入层	9
3.3.2 对齐解码器的结构	11
3.4 基于 Back Translation 的数据增强	14

3.5	损失函数对于自动歌曲翻译模型的优化.....	15
3.6	歌曲翻译的评价指标.....	15
3.7	实验设计.....	16
3.8	实验结果与分析.....	17
3.8.1	主要实验结果.....	17
3.8.2	消融实验分析.....	17
3.9	本章小结.....	18
4	基于扩散模型的歌声合成.....	21
4.1	歌声合成数据集.....	21
4.2	歌声合成数据的收集和预处理.....	21
4.3	歌声合成的参数前端.....	21
4.4	扩散模型.....	21
4.4.1	正向扩散过程.....	21
4.4.2	反向去噪过程.....	21
4.4.3	训练和采样生成.....	21
4.5	浅扩散机制.....	21
4.5.1	浅扩散机制的原理.....	21
4.5.2	浅扩散点的选取.....	21
4.6	歌声合成的评价指标.....	21
4.7	实验结果与分析.....	21
4.8	本章小结.....	21
5	歌曲到歌声翻译系统.....	23
5.1	歌曲翻译结果后处理.....	23
6	总结和展望.....	25
6.1	总结.....	25
6.2	挑战与未来展望.....	25
	参考文献.....	27
	附录.....	29
	作者简历.....	31

图目录

图 1.1	以 <i>Rolling In the Deep</i> 一曲中 “But you play it to the beat” 一句的完整歌曲翻译为例。	2
图 3.1	本文提出的歌曲翻译数据标注流程概览。	8
图 3.2	本章提出的模型架构概览，图示以第 j 个解码时间步进行了说明。Transformer 解码器将输出目标语言的词语，对齐解码器会输出对齐音符的数量。	9
图 3.3	(a) 音符表示池化嵌入层会根据音符序列的表示和对齐信息进行编码。(b) (c) 对齐解码器会根据停止概率的分布计算对齐音符的数量。	9
图 3.4	如何在歌词和歌曲-旋律对齐的共同翻译训练中使用回译数据和带注释的数据的图示说明。“bt” 表示来自 back-translation 回译数据增强得到的数据，“at” 表示来自人工标注的准确歌曲数据。	15
图 3.5	源歌词、参考翻译结果和本章中进行比较的三个歌曲翻译模型的翻译结果的歌谱示例。两个例子分别为《 <i>Will You Love Me Tomorrow</i> 》一曲中的 “Is love I can be sure of” 和《小幸运》一曲中的 “她会有多幸运”。	19
图 3.6	En→Zh 方向测试集中的真实歌曲翻译结果对齐情况和模型预测的对齐结果重叠展示的频率分布直方图。	19

表目录

表 3.1	本文中所涉及的数据集的数据统计情况。	7
表 3.2	对于歌曲翻译的可理解度 (MOS-T)、翻译后歌曲的自然度、可唱性 (MOS-S) 和整体质量评估 (MOS-T) 的平均意见得分和 95% 置信区间。带有 † 标记的翻译方向表示其评测时提供的翻译歌曲片段的音频样本是用未针对该目标语言训练的语音合成模型生成的。因此，这些结果仅供参考。	17
表 3.3	两个翻译方向上的 sacreBLEU 分数和对齐分数结果。(* 表示行内第二优的结果。)	18

‘

1 绪论

1.1 引言

文字和语音都是自然语言处理领域的研究对象。无论是比较早期的句法、词法分析，音素研究，还是诞生稍晚的针对语音内容的识别、拼接合成，对于文字和语音内容的研究可称得上互为表里。语音是文字可感知的信号载体，文字是语音内容的本质表示。机器翻译一直以来都是自然语言处理领域针对不同语言文字的一项重要技术。对于机器翻译的研究最早始于 20 世纪 30 年代，它的发展一直与计算机技术、语言学和信息论等学科密切相关。从早期的字典匹配、结合专家知识的规则翻译等朴素方法，到结合概率统计学和语言学的统计机器翻译，再到近年来随着运算硬件性能提高和深度神经网络发展而兴起的神经机器翻译，机器翻译技术取得了世人瞩目的成功，这项技术本身也逐渐从学界中的理论发展走向工业界的落地实践。

自动歌曲翻译是神经机器翻译在这一基础上针对非常规语体的拓展性研究，这一任务旨在。

歌声合成则是由语音合成任务衍生而来。语音合成是仅以文本为输入、以梅尔特征图或声波波形为输出的生成任务，歌声合成和语音合成任务不同的是，其文本发声时所对应的音高和时长都为设定好的乐谱所限制。歌声合成使得歌曲翻译结果的直观评测成为可能。同时，由于歌声合成可以承接歌曲翻译的输出结果而连接成为完整的、级联式的歌曲到歌声的翻译系统，这为歌曲翻译的实际应用打下了坚实的基础。

1.2 国内外研究现状

1.2.1 自动歌曲翻译国内外研究现状


1.2.2 歌声合成国内外研究现状

最初的歌声合成工作始于连接音频片段式^[1-2]或基于隐马尔可夫统计模型的参数化^[3-4]方法。这些方法由于早期的硬件运算能力、算法研究等局限性，和现在的方法相比过程非常繁琐，而且合成的结果缺乏人声的灵感和和谐感。近年来，深度学习蓬勃发

展,在过去的几年中,基于深度神经网络的歌声合成系统逐渐成为研究主流。利用神经网络将上下文特征映射到声学特征。Ren et al.^[5]使用从音乐网站挖掘的演唱数据,成功构建具有实际落地潜力的歌声合成系统。Blaauw et al.^[6]提出了一种基于 Transformer 的非自回归序列生成模型,相比自回归模型,可以更快速地进行模型推理,而且能避免自回归模型引起的暴露偏差问题。Besides, with the help of adversarial training, Lee et al.^[7] propose an end-to-end framework which directly generates linear-spectrograms. Wu et al.^[8] present a multi-singer SVS system with limited available recordings and improve the voice quality by adding multiple random window discriminators. Chen et al.^[9] introduce multi-scale adversarial training to synthesize singing with a high sampling rate (48kHz). 总之,歌声合成系统相关研究在近年来取得了较大进展,能够合成的歌声语音的自然度、和谐感和多样性不断提高。


1.3 研究意义及内容

歌曲翻译技术是人类为了攀登更高层次的跨文化交流之巴别塔而做出的很有意义的技术努力。然而,尽管机器翻译(Machine Translation, MT)技术,尤其是神经机器



But you play — it to the bea — t

But	you	play	it	to	the	beat					
G4	G4	A4	A4	A4	G4	G4	A4	A4	G4	F4	E4
$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{2}$
但	你	没	有	好	好	珍					惜



但 你 没 — 有 好 好 珍 — 惜

图 1.1 以 *Rolling In the Deep* 一曲中 “But you play it to the beat” 一句的完整歌曲翻译为例。

翻译^[10-12] (Neural Machine Translation, NMT) 的进步,自动歌曲翻译在自然语言处理学界中并未得到充分的研究探索。这其中客观存在的一些挑战包括缺乏收集平行歌词和

对齐数据的高效方式、难以对文本和旋律之间的复杂交互进行建模以及没有对乐谱规定的演唱方式进行直观评估的方式。歌曲翻译虽然与文本翻译密切相关，但本质上是一项更复杂的任务。除了在翻译中如用词和词序这样的考虑之外，歌曲的人工翻译者还需要具有目标语言的背景，能理解源语言并作出目标语言中诗意化的表达。此外，如图1.1所示，翻译的歌词需要与旋律合理地对齐来保持歌曲的美感，这是歌曲翻译中不可缺少的要素^[13]。

此前，学界也探索过歌声合成 (Singing Voice Synthesis, SVS) 这一技术来自动化地合成歌曲的人声演唱，并提出了一些在给定歌词和歌谱的情况下产生具有真实人声音色的、自然的、准确的歌声的方法。这样的方法不但使得对歌曲翻译结果方便而直观的评估成为可能，而且也为自动写歌谱曲、自动歌曲翻译这样的研究任务的实际落地奠定了基础。然而，自动歌曲翻译方向上的研究和歌声合成相比很少。作为目前为数不多的工作之一，Guo et al.^[14]专注于通过在神经机器翻译的推理过程中施加特定约束来匹配有声调语言的翻译目标词语和旋律的音调、节奏等来得到更加合适、不易造成误解的翻译歌词。然而，Guo et al.^[14]直接使用文本翻译模型并对音符和字符之间对齐的严格规定一对一的匹配，无法捕捉到歌曲翻译更复杂的本质——即歌词和歌词-旋律对齐之间的关系。虽然音符的数量可以当作是翻译长度的一个简单上限，但正如 Haapaniemi et al.^[15]一文中所观察到的现象，歌词和旋律之间的微妙对齐不应仅为简单而严格的规则所决定。

为了解决上述技术挑战，我们提出了带有自适应分组的歌词-旋律共同翻译模型，这是自动歌词翻译问题的第一个完整的技术解决方案，通过在基于 Transformer 的编码器-解码器框架内对歌词翻译和歌词-旋律对齐进行联合建模，我们提出的模型翻译出的歌曲既忠实于原歌词，又符合旋律，无论是客观指标还是主观评测都显示出模型翻译表现的优越性。

1.4 章节安排

本文各章节组织如下：

第一章：绪论。第一章节主要介绍了歌曲翻译技术和歌声合成技术的定义、应用和发展情况、自动歌曲翻译和歌声合成的国内外研究现状、以及自动歌曲翻译和歌声合成

的研究背景及意义。另外，第一章节还阐述了本文将基于 Transformer Encoder-Decoder 模型和歌词和歌词-旋律对齐的关系来研究自动歌曲翻译任务、基于扩散模型来研究歌声合成任务，并探究 xxx 对于自动歌曲翻译和 xxx 对于歌声合成效果的影响。

第二章：相关研究介绍。

第三章：自动歌曲翻译研究

第四章：基于扩散模型的歌声合成研究。

第五章：歌曲到歌声翻译系统实践。

第六章：总结和展望。

1.5 章节小结

2 相关研究介绍

本文的研究对象涉及自然语言处理的多个子领域，本章将分别介绍 xxxxxx 的研究现状和相关研究进展。以下分别对 xxxxxx 进行阐述。

2.1 歌曲歌词生成及限制性翻译研究介绍

2.2 扩散模型相关研究介绍

3 自动歌曲翻译

本章将详细介绍本文在自动歌曲翻译研究中使用的数据、提出的模型和实验结果。本章首先说明了使用的单语言数据集来源、收集双语平行数据集的方法，并提出一种基于神经机器翻译中常用的 Transformer Encoder-Decoder 结构的歌词和歌词-旋律对齐共同翻译框架，并在此基础上设计了一系列实验来检验模型框架的表现。实验结果显示，本章提出的框架相比其他自动歌曲翻译算法，在歌词文本翻译质量和歌曲翻译整体演唱效果上都取得了更好的效果。

3.1 歌曲翻译数据集

目前，在歌曲翻译研究领域并没有高质量的平行歌词翻译和歌词-旋律对齐的公共数据集可用，所以本文收集并标注了一个数据集 PopCV (Pop songs with Cover Version)，该数据集包含若干中文歌曲的英文翻唱版本和英文歌曲翻唱版本。除此以外，本文还使用了一些单语言歌曲语料，包括一个英文歌词和歌词-旋律对齐数据集 LMD¹[16]，以及一个从唱吧 App 上爬取的一些中文歌曲语料。两组单语言歌曲数据仅被用于训练模型，测试数据是在经专业标注人员标注的真实数据上进行的。数据集概述见表3.1。

	语种	歌曲数 (首)	歌词数 (句)	数据来源和实验用途
LMD	英文	—	152,991	回译
唱吧	中文	—	542,034	回译
PopCV	中文、英文	79	2,959	标注
测试集	中文、英文	25	629	标注

表 3.1 本文中所涉及的数据集的数据统计情况。

¹<https://github.com/yy11lab/Lyrics-Conditioned-Neural-Melody-Generation>

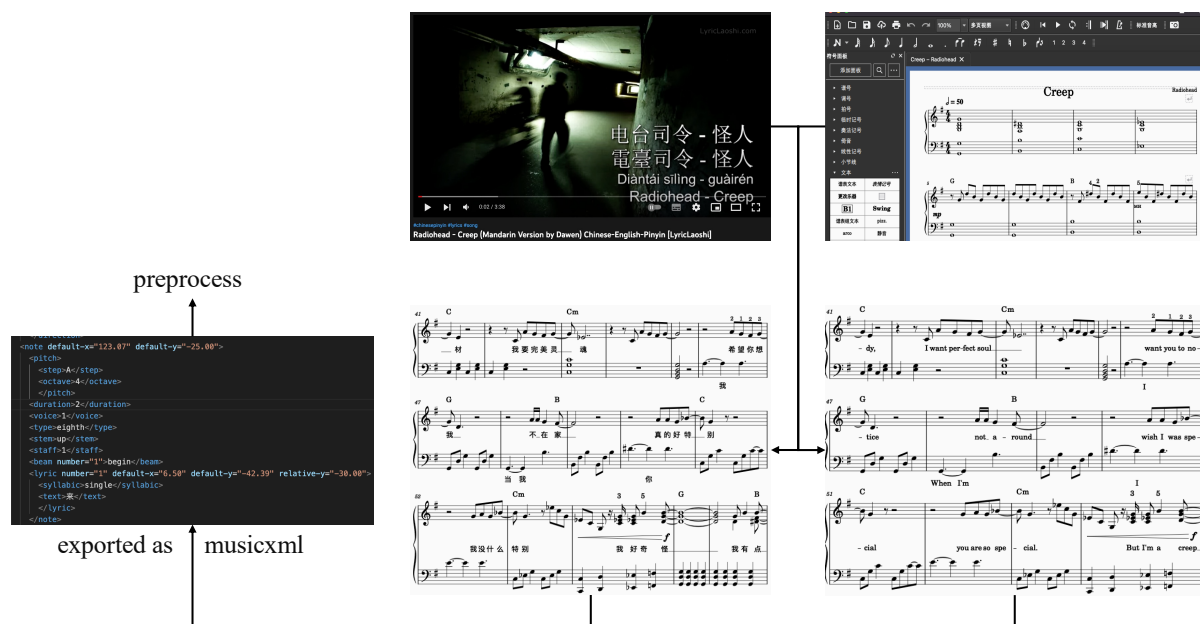


图 3.1 本文提出的歌曲翻译数据标注流程概览。

3.2 歌曲翻译数据的收集和预处理

由于此类歌曲翻译的数据集并没有行业标准或其他公开发表的先例，于是，本文首先设计了一个相对省时且对于标注员来说，比较容易执行的标注过程。首先，从一些公开的乐谱网站收集歌曲的乐谱文件²。然后，专业标注人员会根据歌曲在原版和翻唱版本中的演唱方式，按照一般的歌谱编纂规则的指示³将歌词添加到乐谱的音符上。然后，标注好的乐谱文件会被以 .musicxml 的格式导出，然后自动提取出歌词及其对齐的音符并整理成数据集。

3.3 自动歌曲翻译模型结构

本章设计的模型属于神经机器翻译中常用的自回归翻译架构，但与一般的翻译模型不同的是，它能同时进行自回归的歌词文本翻译和歌词文本与旋律的对齐预测。如图3.2所示，它由用于歌词翻译的基于 Transformer Encoder-Decoder 的子结构、两个音符表示池化嵌入层和一个对齐解码器组成。Transformer Encoder-Decoder 部分参考了 Guo et al.^[14]中的做法，使用去噪自编码器^[17]和翻译作为预训练任务。在预训练期间，由于混

²<https://www.musescore.com> 和 <https://www.midishow.com>

³<https://lilypond.org> and <https://musescore.org/howto>

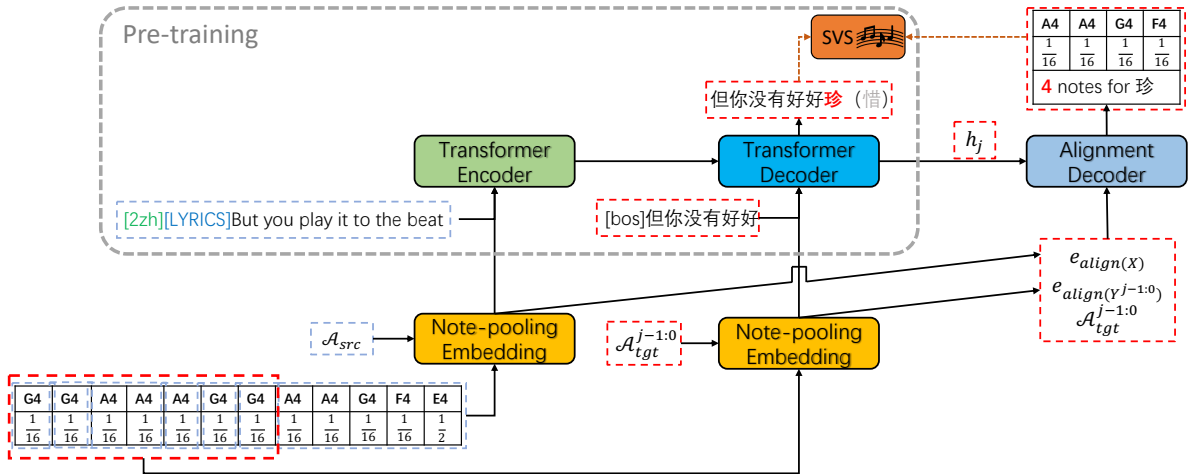


图 3.2 本章提出的模型架构概览，图示以第 j 个解码时间步进行了说明。Transformer 解码器将输出目标语言的词语，对齐解码器会输出对齐音符的数量。

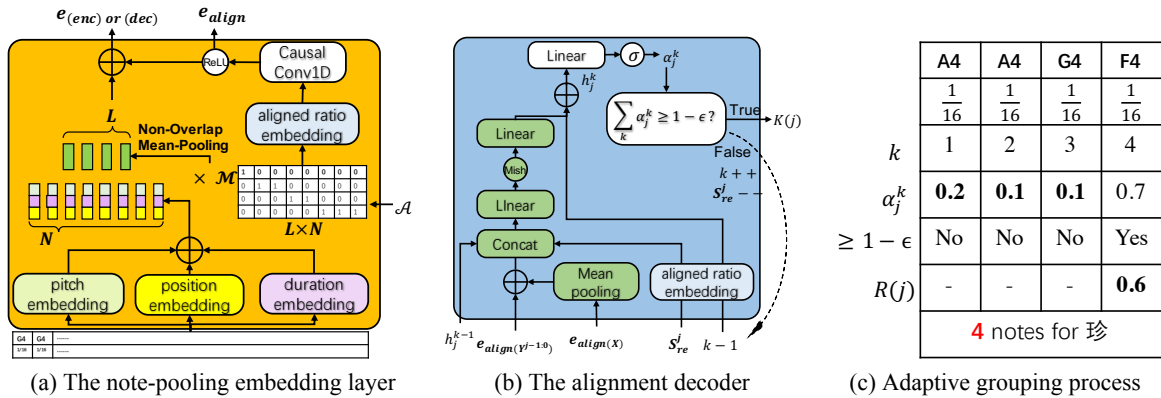


图 3.3 (a) 音符表示池化嵌入层会根据音符序列的表示和对齐信息进行编码。(b) (c) 对齐解码器会根据停止概率的分布计算对齐音符的数量。

合使用了单语言和双语翻译数据，且数据文本来自新闻、书本和歌词等多个领域，两个分别表示翻译方向和文本域的前缀词语会被添加到源语言的输入句子里以建立模型区分翻译方向和目标文本所属文本域的能力。音符表示池化嵌入层的结构如图 3.3a 所示，这是一个用于处理歌曲旋律信息的模块。图 3.3b 中的对齐解码器则是基于本章后提出的节的自适应音符分组方法构建，该方法在自回归解码期间能动态预测与当前解码时间步预测出的的词语相对齐的音符数量。

3.3.1 音符表示池化嵌入层

音符表示池化嵌入层将音符和音符与歌词词语的对齐信息作为输入，并输出池化后的音符嵌入表示和对齐嵌入表示。输入的旋律音符序列由 MIDI 格式的音高和每个音

符的持续时间两部分组成。每一个音符的持续时间都是离散的谱面时长的一类：四分音符、半音符或八分音符等。音符的 MIDI 音高和持续时间可以分别表示为嵌入表示 e_{midi} 和 e_{dur} 。定义第 i 个音符的嵌入表示为：

$$\mathbf{e}_{note}^i = \mathbf{e}_{midi}^i + \mathbf{e}_{dur}^i + \mathbf{e}_p^i \quad (3-1)$$

其中， \mathbf{e}_p^i 是位置嵌入表示。池化嵌入层会根据对齐信息，对音符的嵌入表示序列进行互不重叠的平均池化操作。具体而言，就是将对齐到同一词语的连续的一段音符序列的嵌入表示进行平均。下面给出公式化的表达，对齐信息 \mathcal{A} 会被表示为一个 01 矩阵 $\mathbf{M} \in \{0, 1\}^{L \times N}$ ，其中 L 和 N 分别表示文本序列和音符序列的序列长度。如果第 i 个音符对齐到了第 j 个词语，那么 $\mathbf{M}_{ji} = 1$ ，否则 $\mathbf{M}_{ji} = 0$ 。这样， \mathbf{M} 就能直接通过矩阵乘法有效地进行互不重叠的平均池化操作，其结果记为为旋律嵌入表示 \mathbf{e}_{md} 。

$$\mathbf{e}_{md} = \text{Non-Overlap-Mean-Pool}(\mathbf{e}_{note}, \mathbf{M}) \quad (3-2)$$

有音符嵌入表示 $\mathbf{e}_{note} \in \mathbb{R}^{N \times d}$ (d 是嵌入表示张量的维度) 和对齐矩阵 $\mathbf{M} \in \{0, 1\}^{L \times N}$ 。互不重叠的平均池化操作可以按照如下计算进行：

$$\mathbf{W} = \mathbf{M} / \text{sum}(\mathbf{M}, \text{dim} = -1, \text{keepdim} = \text{True})$$

$$\mathbf{e}_{md} = \mathbf{W} * \mathbf{e}_{note}$$

其中， $/$ 代表矩阵按元素相除， $*$ 代表矩阵相乘。通过 pytorch 支持的 `gather` 和 `scatter` 张量操作，上述互不重叠的平均池化操作就可以在训练时的小批次数据中进行了。由上述说明易知，此池化操作的核的尺寸大小不是固定的，而是随着 01 矩阵 \mathbf{M} 的行的和变化而变化。

进一步说，由于歌词-旋律的对齐是单调的，即歌谱中每个音符只能对应一个词语，通过计算对齐音符的数量的累积和就可以更简洁地编码对齐情况：

$$\mathbf{s} = \text{CumSum}(\text{RowSum}(\mathbf{M})) \quad (3-3)$$

其中， \mathbf{s} 是一个长度为 L 的整数向量。那么 s^j/N 就表示每个对齐音符的**对齐比率**。接下来，通过将累积对齐比率分组为 $(0, 1]$ 范围内的大小相等的区间就可以将比率离散化，并引入一组嵌入表示张量 \mathbf{E}_{ratio} 来表示每个区间。划分成的区间数是一个可调节的超参

数。所以，对齐比率嵌入表示的计算方法如下：

$$\mathbf{e}_{align}^j = f(\mathbf{E}_{ratio}(s^j/N)) \quad (3-4)$$

其中， $f(\cdot)$ 是一个简单的非线性神经网络层，由一维因果卷积层和 ReLU 激活函数组成。最后，将旋律嵌入表示和对齐比率嵌入求和，结果被输入到基于 Transformer 的子结构的编码器或解码器，和其中原有的嵌入表示相加：

$$\mathbf{e}_{enc(dec)} = \mathbf{e}_{token} + \mathbf{e}_p + (\mathbf{e}_{md} + \mathbf{e}_{align}) \quad (3-5)$$

如公式 (3-2) 所示，每个旋律嵌入表示都会对应于该段旋律对齐到的词语。此外，使用因果卷积层意味着音符对齐比率的嵌入张量也具有与文本序列相同的长度，并且能够保证每个对齐比率的嵌入表示仅以自回归的方式与先前的比率嵌入表示相关。上述性质就保证了该层在解码器中可以完美地适应自回归方式的翻译需要进行的 Teacher-forcing 式训练。来自源端歌词的对齐嵌入表示由于在解码时并没有目标端的词语可以对去，所以这部分会整体经过池化层处理以形成全局的对齐参考表示，并输入到对齐解码器中。本章提出的这种设计的动机在于用对齐的音符数来隐式地对歌词的翻译过程进行限制。

3.3.2 对齐解码器的结构

受自适应计算时间方法 (Adaptive Computation Time, ACT)^[18] 的启发，本章提出了 **自适应分组** 模块来对歌词和音符的对齐情况进行建模。如图 3.3b, 3.3c 所示，此模块能够预测出应将多少个连续的音符分配给当前解码时间步正在处理的词语。

3.3.2.1 自适应音符分组预测

一般地，有 $1 \leq j \leq L_Y$ ，设 y_j 为第 j 个目标端文本词语， \mathbf{h}_j 为基于 Transformer 的解码器的最后一层相对应的隐层表示。为了说明之便而又不失一般性，假设之前的文本序列 $y_{j-1:0}$ 已经与前 $n-1$ 个音符完成对齐，下文将通过遍历一个索引变量 k (k 从 1 开

始) 来定义自适应音符分组预测与 y_j 对齐的音符数量的过程。

$$\mathcal{S}_{re}^j = N - s_{tgt}^{j-1} \quad (3-6)$$

$$\mathbf{h}_j^0 = \mathbf{h}_j \quad (3-7)$$

$$\mathbf{h}_j^k = g(\mathbf{h}_j^{k-1}, \mathbf{e}_{align(X)}, \mathbf{h}_{align(y_{j-1:0})}, \mathcal{S}_{re}^j, k-1) \quad (3-8)$$

$$\alpha_j^k = \sigma(\text{Linear}(\mathbf{h}_j^k)) \quad (3-9)$$

其中, $\mathbf{e}_{align(X)}$ 是完整的来自源语言端的对齐嵌入表示, $\mathbf{e}_{align(y_{j-1:0})}$ 则是已经过解码的先前部分的对齐嵌入表示, s_{tgt}^{j-1} 是 \mathbf{s} 向量中的第 j 元素 (\mathbf{s} 来自公式 (3-3))。现在模型既有每句歌词对应的完整旋律中的音符数量信息, 又有已经对齐到目标端的音符数量, 那么可以计算当前解码步骤中第 j 时间步时, 剩余未对齐音符的数量为 \mathcal{S}_{re}^j 。如上文所述, $\mathbf{e}_{align(X)}$ 经过一个平均池化层以获得单个向量表示作为全局参考, 从而使得来自源语言端的对齐情况始终可以与可变长度的目标端的 $\mathbf{e}_{align(y_{j-1:0})}$ 进行加和。一个多层神经网络 $g(\cdot)$ 会对所有的输入进行处理, 具体网络结构如图3.3b中绿色部分所示。最后, 经 Sigmoid 函数 $\sigma(\cdot)$ 处理, 模块会这一步处理中间态的自适应分组停止概率 α_j^k 。所有中间态停止概率的总和表示当前 k 个音符与目标端词语 y_j 对齐的可能性。

给定一个超参数 ϵ , 通常为一个很小的浮点数 (例如, 0.01), 如果此时的 k 满足 $\sum_k \alpha_j^k < 1 - \epsilon$, 即累计概率未超过阈值, 那么自适应分组过程会继续进行并将 k 递增至 $k = k + 1$; 相应递减 $\mathcal{S}_{re}^j = \mathcal{S}_{re}^j - 1$, 然后进行上述计算。否则, 累计概率已经超过了既定阈值, 对齐预测的分组过程停止, 对齐解码器输出当前对齐的音符数 $K(j)$ 。

$$K(j) = \underset{K}{\operatorname{argmin}} \left\{ \sum_{k=1}^K \alpha_j^k \geq 1 - \epsilon \right\} \quad (3-10)$$

在 ϵ 是正值 $\epsilon > 0$ 的情况下, 这个预测过程能确保 $K(j) \geq 1$, 也就是说对于每个词语, 至少有一个音符会被对齐到该词语上。为了清晰地定义对齐 $K(j)$ 个音符到当前词语的概率, 引入一个余项 $R(j) = 1 - \sum_{k=1}^{K(j)-1} \alpha_j^k$ 。这样, α_j^k 和 $R(j)$ 就都可以是有效的概率分布了。图3.3c是本节提出的自适应分组方法在歌词音符对齐预测上运行的一个示例。

$$\begin{aligned} L_G &= \left| \sum_j K(j) - N \right| + \sum_j |K(j) - \Delta_j| \\ &\approx \left| \sum_j (K(j) - (1 - R(j))) - N \right| \\ &\quad + \sum_j |K(j) - (1 - R(j)) - \Delta_j| \end{aligned} \quad (3-11)$$

3.3.2.2 自适应分组模块的梯度计算

本小节将详细阐述3.3.2.1小节中介绍的自适应分组算法的优化过程，由于算法运行过程是求满足 $\sum_{k=1}^K \alpha_j^k \geq 1 - \epsilon$ 的 k 的最小值，该过程本身不可导，所以对算法中使用的神经网络参数的优化和梯度计算需要一些技巧。在人工标注的歌词-旋律对齐数据中，每个目标端文本序列的词语所对齐的音符数的真实值都被标注出来了，这里先将序列中第 j 个词语对齐的此值记为 Δ_j 。与自适应计算时间方法^[18]原文中最小化“思考成本” $\sum_j K(j) + R(j)$ 不同的是，本节提出的自适应分组算法会通过最小化下面这项自适应分组损失 L_G 来优化参数。这种做法会通过引入 Δ_j 来自然地限制目标端文本序列中每个词语的“思考成本”的上界。

$$\begin{aligned} L_G &= \left| \sum_j K(j) - N \right| + \sum_j |K(j) - \Delta_j| \\ &\approx \left| \sum_j (K(j) - (1 - R(j))) - N \right| \\ &\quad + \sum_j |K(j) - (1 - R(j)) - \Delta_j| \end{aligned} \quad (3-12)$$

显然，变量 $K(j)$ 的计算方式对于停止概率是不连续的，因此模型在优化时近似使用了 $1 - R(j)$ 项以使自适应分组损失可微。根据上述自适应分组损失的定义，其实只需要分析以下项对预测对齐音符数的作用：

$$|K(j) - (1 - R(j)) - \Delta_j| \quad (3-13)$$

1) 如果在正向传播过程中， $K(j) > \Delta_j$ ，则由于本节提出方法的设定， $K(j)$ 和 Δ_j 都是正数，所以有 $K(j) - \Delta_j \geq 1$ 。那么，为了最小化自适应分组损失， $1 - R(j) = \sum_{k=1}^{K(j)-1} \alpha_j^k$ 就应该被调整得更大。也就是说，此时，优化过程会将 $\sum_{k=1}^{K(j)-1} \alpha_j^k$ 这一项逐渐推向 $K(j) - \Delta_j$ 。Note that the theoretical upper bound of $\sum_{k=1}^{K(j)-1} \alpha_j^k$ is $K(j) - 1$, which is larger or equal to $K(j) - \Delta_j$. 因此，这种情况下对参数的优化是有效的，并且在优化期间将满足以下条件：

$$\sum_{k=1}^{K(j)-1} \alpha_j^k \geq 1 - \epsilon \quad (3-14)$$

根据前文中对 $K(j)$ 的定义，可以得出以下结论：

$$K(j)^{\text{new}} = \arg \min_K \left\{ \sum_{k=1}^K \alpha_j^k \geq 1 - \epsilon \right\} \leq K(j) - 1 \quad (3-15)$$

2) 如果有 $K(j) < \Delta_j$, 相应过程的推导分析也是类似的。这种情况下, 要将 $\sum_{k=1}^{K(j)-1} \alpha_j^k \rightarrow 0$ 往减小自适应损失的方向推, 也就意味着如果第 $K(j)$ 个计算得出的停止概率仍不满足 $\alpha_j^{K(j)} \geq 1 - \epsilon$ 的条件, 那么 $K(j)^{\text{new}}$ 就会在优化过程中呈增大趋势。然而, 如果有 $\alpha_j^{K(j)} \geq 1 - \epsilon$, 优化就会停止。所以可以优化 $\left| K(j) - \left(1 - R(j) + \alpha_j^{K(j)} \right) - \Delta_j \right|$, 即, $\left| K(j) - \sum_{k=1}^{K(j)} \alpha_j^k - \Delta_j \right|$ 。虽然有以上分析结果, 但是在实际训练中的观察发现, 这种情况其实非常罕见, 几乎不会出现。就算出现, 在训练仅进行几个周期后, 这种现象就会完全消失。因此, 在训练中可以仍然采用了上述 1) 情况中同样形式的自适应分组损失。

3) 如果有 $K(j) = \Delta_j$, 那么此词语对应的损失项在自适应分组损失中就可以完全不起作用, 无需其他讨论。

此外, 由于与多个音符对齐的词语在歌曲中属于少数, 因此在实际训练时, 来自于这些词语的对齐损失会被加上更大的权重来让优化更多地关注这些重点部分的对齐预测情况。

3.4 基于 Back Translation 的数据增强

虽然本章中收集的规模在上千句左右的歌曲数据集能够初步满足训练模型的需求, 其中包含的来自人工翻译平行双语歌词和歌词-旋律对齐信息的标注数量是非常有限的, 而且这对标注人员素质提也提出了较高要求, 因此对于实际应用会有收集耗时较长, 代价昂贵的问题。

所以, 本章还改造了近年来神经机器翻译研究中广泛使用的基于回译的数据增强方法^[19]来生成更多的双语歌曲训练数据。相比于本章提出的人工标注的平行双语歌曲数据集, 在公开网络上显然可以搜集到数量更为庞大的单语歌曲数据。通过构建一个可以进行长度控制的预训练歌词翻译模型, 目标语言中的单语数据就能被回译到源语言中。长度控制可以确保翻译出的结果的词语的数量与所对应的旋律的音符数量相同, 这样就能在源语言端制造歌词和音符的简单一对一对齐。经过如此改造后的回译方法就能够制造出相对较大的双语歌曲数据集。显然, 这样增强出的数据在源端有信息噪声, 但在目标端仍然保留了非常准确的信息。由于回译出的数据比人工标注的数规模大得多, 本章在实际训练中采用了类似课程学习的方式来进行训练时的数据采样。即在训练初期, 来自回译增强的数据将与来自人工标注的数据真实数据混合。人工标注的数据被上采样到

与回译数据差不多的数量级，对回译数据的降采样率则会随训练进行不断减小，这样，每个小批次训练数据中人工标注数据的占比就会随着训练的进行不断提高。具体来说，回译产生的数据的下采样率从 1.00 开始，在总训练周期的一半时下降到 0.01。人工标注的数据的采样率从 20.00 开始进行上采样，并在总训练周期结束时降至 5.00。

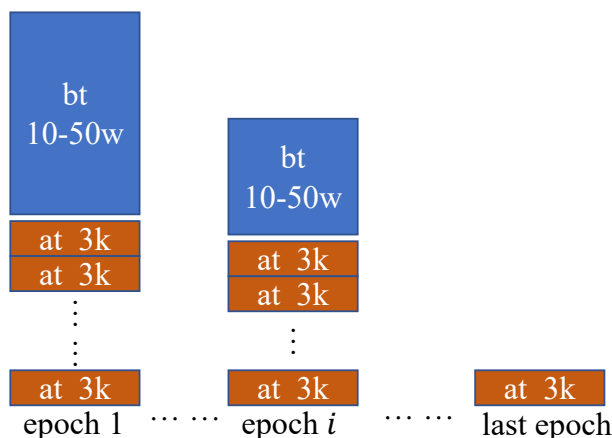


图 3.4 如何在歌词和歌曲-旋律对齐的共同翻译训练中使用回译数据和带注释的数据的图示说明。“bt”表示来自 **back-translation** 回译数据增强得到的数据，“at”表示来自人工标注的准确歌曲数据。

3.5 损失函数对于自动歌曲翻译模型的优化

3.6 歌曲翻译的评价指标

对本章提出的模型所针对的自动歌曲翻译任务的表现的评价，最有说服力的指标就是翻译后的歌曲结果是否能被正常演唱、歌词文本易于理解，以及，最核心的指标，是否仍和原歌曲一样是人类乐于欣赏的歌曲作品。因此，本章的实验评测参考了 Sheng et al.^[20] 中的做法：在进行人工评测时，评测人员会根据展示出的模型翻译结果——歌词和歌词-旋律对齐情况制成的歌谱，给出评测打分。但是由于本章所针对的自动歌曲翻译任务的特殊性，为了以一种接近实际中端到端的方式验证翻译结果的可唱性，本章的实验评测还使用了一个开源的歌唱语音合成模型^[21]为评测人员提供翻译后歌曲的演唱音频，期望以此让标注人员进行更直观的演唱评测。

进行人工测试的歌谱和声音是由测试集中随机得到的 20 个片段，并展示乐谱和合成歌声对于翻译质量的自动评测，本章采用了 sacreBLEU⁴分数。对于歌曲翻译的可理解

⁴<https://github.com/mjpost/sacrebleu>

度、翻译后歌曲的自然度、可唱性和整体质量评估,本章采用了人工评估,使用平均意见得分 (Mean Opinion Score, MOS), 三项分别对应分数 MOS-T、MOS-S 和 MOS-Q。在自动评估对齐情况时,机器翻译领域中常用的对齐错误率 (Alignment Error Rate, AER) 却并不适用于此,因为除了模型生成的需要评测的歌曲-旋律对齐之外,目标端的歌词翻译结果也是机器生成的。于是,本节提出了一种对齐分数 (Alignment Score, AS), 用于计算预测的歌词-旋律对齐和真实情况之间的在测试集上的统计频率的加权交 (Intersection over Groundtruth, IOG):

$$AS = \frac{\sum_k \min(\text{freq}_{pred}^k / F_{pred}, \text{freq}_{gt}^k / F_{gt}) * k}{\sum_k (\text{freq}_{gt}^k / F_{gt}) * k} \quad (3-16)$$

式中 k 即为前文中的对齐音符数, 而 $F = \sum_k \text{freq}^k$ 是频率的总和。

3.7 实验设计

其中之一是在上文中提到过的 **GagaST** 系统^[14], 该文章的主要关注点是根据如中文这样的带音调语言的歌词音调限制翻译结果, 并搭建了一个歌词翻译系统, 在对解码过程的限制进行调整后也能适应无音调语言的歌词翻译。另一个是本章提出的模型的变体。此变体使用的是基于 **Transformer Encoder-Decoder** 子结构输出的隐层表示的分类器对每个翻译出的目标端的词语的对齐音符进行预测。可能的对齐音符的最大数目类别设置为 30, 这个量与上节阐述的对齐解码器中模型允许的出现最大 $K(j)$ 相同。

为了生成符合音乐五线谱规则的乐谱和歌声, 我们在对齐预测中添加了一些基于规则的后处理, 以获得对模型预测结果更大的容忍度。对于序列整体对齐音符预测总和大于旋律中音符的数量的情况, 预测的对齐音符数量会被从最后一个词语到第一个词语 (或从第一个词语至最后一个词语) 一一减小直到预测总数和旋律的音符总数相符。对于预测音符数量较少的情况, 所有相差的对齐音符数量都会被添加到最后一个词语处。这样的规则是统一、简单又比较符合直觉的, 不会对预测结果产生本质性影响。

模型名称	MOS-T		MOS-S		MOS-Q	
	En→Zh	Zh→En	En→Zh	Zh→En [†]	En→Zh	Zh→En [†]
GagaST	3.66 ± 0.06	3.72 ± 0.05	3.49 ± 0.10	\	3.65 ± 0.05	\
LTAG-cls	3.66 ± 0.05	3.79 ± 0.05	3.58 ± 0.07		3.62 ± 0.05	
only bt	3.69 ± 0.05	3.80 ± 0.04	3.53 ± 0.09		3.63 ± 0.05	
w/o bt	3.64 ± 0.05	3.30 ± 0.05	2.16 ± 0.05		3.14 ± 0.04	
LTAG	3.71 ± 0.05	3.85 ± 0.05	3.68 ± 0.05		3.69 ± 0.04	
only bt	3.71 ± 0.05	3.80 ± 0.05	3.58 ± 0.07		3.65 ± 0.04	
w/o bt	3.69 ± 0.05	3.28 ± 0.04	3.63 ± 0.07		3.67 ± 0.04	

表 3.2 对于歌曲翻译的可理解度 (MOS-T)、翻译后歌曲的自然度、可唱性 (MOS-S) 和整体质量评估 (MOS-Q) 的平均意见得分和 95% 置信区间。带有[†] 标记的翻译方向表示其评测时提供的翻译歌曲片段的音频样本是用未针对该目标语言训练的语音合成模型生成的。因此，这些结果仅供参考。

3.8 实验结果与分析

3.8.1 主要实验结果

本节对 LTAG 和上节中阐述的其他两个基线歌曲翻译模型进行比较，并报告实验结果。首先，表3.2中罗列了中文到英文 (Zh→En) 和英文到中文 (En→Zh) 两个翻译语向的歌曲翻译质量的人工评价指标 (MOS-T) 的结果。总体上来说，本章提出的 LTAG 与两个基线模型相比都得到了改进，但人工评测的实验结果也显示，模型间和不同实验设置之间的差距并不明显。这样的实验结果部分是由于专业人士翻译歌词的结果通常是意译较多而非直译较多。那么在不同片段中，一个单词缺失可能会对人工评价的结果造成负面、中性甚至正面的影响，只有较明显的语义偏差或语法错误会导致分数在一定程度上下降。正如在节3.6中所讨论的，自动评测指标 sacreBLEU 应该不是比较歌词的机器翻译质量和自由翻译质量的合适的衡量标准。

3.8.2 消融实验分析

首先，本节分析了一些消融实验以研究不同设置下通过回译增强产生的数据的影响。在表3.2中，对 LTAG 和 LTAG-cls 的分析可以得到有以下几点发现：(1) 由于回译增强产生的歌曲翻译数据量明显大于真实人工标注的数据量，如果仅使用回译增强产生

模型名称	BLEU \uparrow		AS. \uparrow	
	En \rightarrow Zh	Zh \rightarrow En	En \rightarrow Zh	Zh \rightarrow En
GagaST	11.87	5.67	0.701	0.468
LTAG-cls	14.21	10.01	0.827	0.555
only bt	15.54	10.21	0.709	0.667
w/o bt	13.73	8.26	0.704	0.490
LTAG	16.02*	10.68	0.923	0.781
only bt	16.27	10.26*	0.880*	0.718*
w/o bt	14.12	7.86	0.845	0.710
w/o e_{align}	15.16	9.24	0.852	0.703

表 3.3 两个翻译方向上的 sacreBLEU 分数和对齐分数结果。（* 表示行内第二优的结果。）

的翻译数据进行训练，则模型在翻译质量上的表现几乎没有差异。这使得在平行数据非常匮乏和昂贵的情况下，无监督训练方式可以很好地满足训练模型的需要。（2）如果只使用数量有限的人工标注监督数据，模型性能会显著下降。（3）LTAG 在这一系列消融实验中的表现始终要比 LTAG-cls 更好。此外，从音符表示池化嵌入层和对齐解码器中移除本章提出的对齐嵌入结构 e_{align} 的消融实验验证了这以结构的重要性。从相关实验结果的对比中可以观察到，不使用 e_{align} 提供的信息，模型翻译的歌曲的 BLEU 和对齐分数都出现了不可忽略的下降。

3.9 本章小结

本章主要针对自动歌曲翻译任务提出了一个可以进行同时考虑歌词和歌词-旋律对齐情况的翻译模型。本章提出的 LTAG，本章细致阐述了，介绍了本章提出的数据收集方法、实验中使用的数据集情况、。最后，通过对比实验和翻译结果的歌谱展示说明了 LTAG 相比两个很强的基线模型取得了更好的文本语义翻译表现和更为合理、更有表现力的歌词-旋律对齐，从而获得了整体上更好的歌曲翻译表现。

(a) 源歌谱和参考翻译结果左图：英 → 中。右图：中 → 英。

(b) GagaST

(c) LTAG-cls

(d) LTAG

图 3.5 源歌词、参考翻译结果和本章中进行比较的三个歌曲翻译模型的翻译结果的歌谱示例。两个例子分别为《Will You Love Me Tomorrow》一曲中的“Is love I can be sure of”和《小幸运》一曲中的“她会有多幸运”。

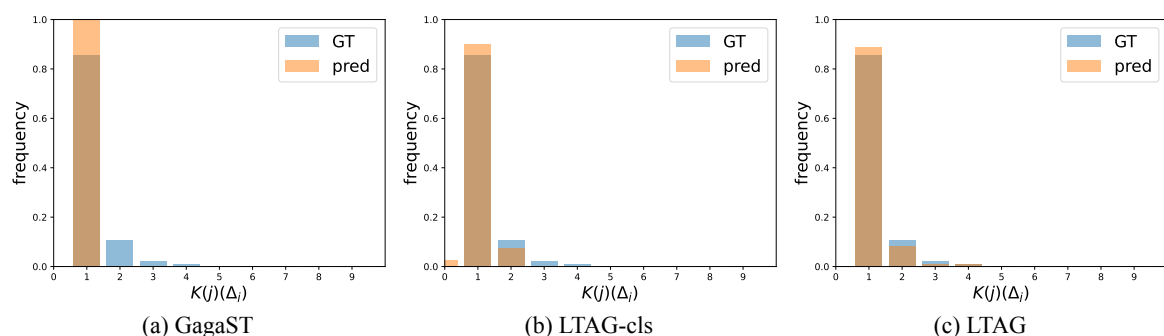


图 3.6 En→Zh 方向测试集中的真实歌曲翻译结果对齐情况和模型预测的对齐结果重叠展示的频率分布直方图。

4 基于扩散模型的歌声合成

4.1 歌声合成数据集

PopCS

4.2 歌声合成数据的收集和预处理

4.3 歌声合成的参数前端

4.4 扩散模型

4.4.1 正向扩散过程

4.4.2 反向去噪过程

4.4.3 训练和采样生成

4.5 浅扩散机制

4.5.1 浅扩散机制的原理

4.5.2 浅扩散点的选取

4.6 歌声合成的评价指标

质量 MOS, 速度 RTF

4.7 实验结果与分析

4.8 本章小结

5 歌曲到歌声翻译系统

5.1 歌曲翻译结果后处理

6 总结和展望

6.1 总结

Guo et al.^[14]

6.2 挑战与未来展望

参考文献

- [1] MACON M, JENSEN-LINK L, GEORGE E B, et al. Concatenation-based MIDI-to-singing voice synthesis[C]//Audio Engineering Society Convention 103. 1997.
- [2] KENMOCHI H, OHSHITA H. Vocaloid-commercial singing synthesizer based on sample concatenation[C]//Eighth Annual Conference of the International Speech Communication Association. 2007.
- [3] SAINO K, ZEN H, NANKAKU Y, et al. An HMM-based singing voice synthesis system[C]//Ninth International Conference on Spoken Language Processing. 2006.
- [4] OURA K, MASE A, YAMADA T, et al. Recent development of the HMM-based singing voice synthesis system—Sinsy[C]//Seventh ISCA Workshop on Speech Synthesis. 2010.
- [5] REN Y, TAN X, QIN T, et al. Deepsinger: Singing voice synthesis with data mined from the web[C]//Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2020: 1979-1989.
- [6] BLAAUW M, BONADA J. Sequence-to-sequence singing synthesis using the feed-forward transformer[C]//ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2020: 7229-7233.
- [7] LEE J, CHOI H S, JEON C B, et al. Adversarially Trained End-to-End Korean Singing Voice Synthesis System[J]. Proc. Interspeech 2019, 2019: 2588-2592.
- [8] WU J, LUAN J. Adversarially Trained Multi-Singer Sequence-to-Sequence Singing Synthesizer[J]. Proc. Interspeech 2020, 2020: 1296-1300.
- [9] CHEN J, TAN X, LUAN J, et al. HiFiSinger: Towards High-Fidelity Neural Singing Voice Synthesis [J]. arXiv preprint arXiv:2009.01776, 2020.
- [10] BAHDANAU D, CHO K, BENGIO Y. Neural machine translation by jointly learning to align and translate[J]. In Proceedings of the International Conference on Learning Representations, 2015.
- [11] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.
- [12] HASSAN H, AUE A, CHEN C, et al. Achieving human parity on automatic chinese to english news translation[J]. arXiv preprint arXiv:1803.05567, 2018.
- [13] FRANZON J. Three dimensions of singability. An approach to subtitled and sung translations[J]. Text and Tune. On the Association of Music and Lyrics in Sung Verse. Bern: Peter Lang, 2015: 333-346.
- [14] GUO F, ZHANG C, ZHANG Z, et al. Automatic Song Translation for Tonal Languages[C/OL]//Findings of the Association for Computational Linguistics: ACL 2022. Dublin, Ireland: Association for Computational Linguistics, 2022: 729-743. <https://aclanthology.org/2022.findings-acl.60>. DOI: 10.18653/v1/2022.findings-acl.60.
- [15] HAAPANIEMI R, LAAKKONEN E. The materiality of music: interplay of lyrics and melody in song translation[J]. Translation Matters, 2019.
- [16] YU Y, SRIVASTAVA A, CANALES S. Conditional LSTM-GAN for Melody Generation from Lyrics [J/OL]. ACM Trans. Multimedia Comput. Commun. Appl., 2021, 17(1). <https://doi.org/10.1145/3424116>. DOI: 10.1145/3424116.
- [17] LEWIS M, LIU Y, GOYAL N, et al. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension[C/OL]//Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Online: Association for Computational Linguistics, 2020: 7871-7880. <https://aclanthology.org/2020.acl-main.703>. DOI: 10.18653/v1/2020.acl-main.703.
- [18] GRAVES A. Adaptive computation time for recurrent neural networks[J]. arXiv preprint arXiv:1603.08983, 2016.
- [19] SENNRICH R, HADDOW B, BIRCH A. Improving Neural Machine Translation Models with Monolingual Data[C/OL]//Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Berlin, Germany: Association for Computational Linguistics, 2016: 86-96. <https://aclanthology.org/P16-1009>. DOI: 10.18653/v1/P16-1009.
- [20] SHENG Z, SONG K, TAN X, et al. Songmass: Automatic song writing with pre-training and alignment

- constraint[C]//Proceedings of the AAAI Conference on Artificial Intelligence: vol. 35: 15. 2021: 13798-13805.
- [21] LIU J, LI C, REN Y, et al. Diffsinger: Singing voice synthesis via shallow diffusion mechanism[C]// Proceedings of the AAAI Conference on Artificial Intelligence: vol. 36: 10. 2022: 11020-11028.

附录

作者简介

已发表的会议论文

已授权的发明专利