**PaddleSpeech: An Easy-to-Use All-in-One Speech Toolkit**

Hui Zhang1 , Tian Yuan1 , Junkun Chen3 , Xintong Li2 , Renjie Zheng2 , Yuxin Huang1 , Xiaojie Chen1 , Enlei Gong1 , Zeyu Chen1 , Xiaoguang Hu1 , Dianhai Yu1 , Yanjun Ma1 , Liang Huang2,3

1Baidu Inc., Beijing, China

2Baidu Research, Sunnyvale, CA, USA

3Oregon State University, Corvallis, OR, USA

{zhanghui41, yuantian01, xintongli, renjiezheng, huangyuxin, chenxiaojie06, gongenlei, chenzeyu01}baidu.com

摘要

PadderSpeech是一个开源的多功能语音工具包。它旨在通过提供易于使用的命令行界面和简单的代码结构，促进语音处理技术的开发和研究。本文描述了PadleSpeech的设计理念和核心架构，以支持几个基本的语音到文本和文本到语音任务。PadleSpeech在各种语音数据集上实现了极具竞争力或最先进的性能，并实现了最流行的方法。它还提供了配方和预训练模型，以快速再现本文中的实验结果。

1 介绍

语音处理技术使人类能够 直接与计算机通信，这是 这是巨大应用程序的重要组成部分 作为智能家居设备(Hoy,2018)，自主以及同声翻译(Zheng et al., 2020)。开源工具包降低了语音处理技术的应用和研究障碍，从而促进了语音处理技术的发展 面积(Young et al.,2002;Lee et al.，2001;HugginsDaines et al.,2006;Rybach et al.,2011;Povey et al.,2011;Watanabe et al.,2018;

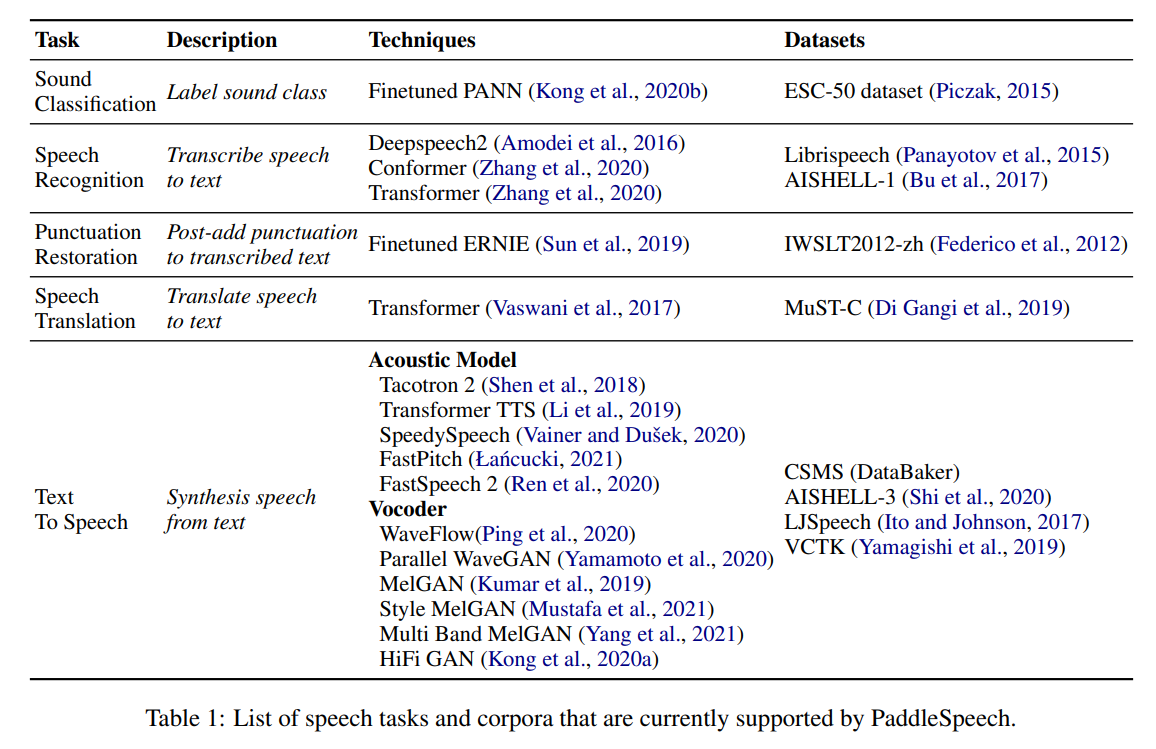
Han et al.,2019;Wang et al.,2020;Ravanelli et al.,2021;Zhao et al.,2021)。

然而，目前流行的语音处理工具包会认定他们的用户是有经验的从业者或研究人员，因此初学者在开发他们令人兴奋的应用程序时可能会感到困惑。例如，使用Kaldi (Povey et al.，2011)构建新的语音应用程序原型，用户必须能够轻松地阅读和修改使用Bash、Perl和Python脚本编写的决策，并精通c++以破解其实现。最近的工具包，例如Fairseq S2T (Wang et al.，2020)和NeurST(Zhao et al.，2021)，在建立在通用深度学习库上变得更加灵活。但它们复杂的代码风格也使学习费时，而且很难从一个迁移到另一个。因此，我们开发了PaddleSpeech，提供命令行界面和便携式功能，使每个人都能开发与语音相关的应用程序。

值得注意的是，中文社区有许多开发者渴望为社区做出贡献。然而几乎所有的深度学习库，例如 Pytorch(Paszke et al., 2019) 和Tensorflow(Abadi et al., 2016)，主要都是针对英文社区，大大增加了中文开发者的难度。PaddlePaddle作为唯一面向中英文社区的全功能开源深度学习平台，累计超过500k提交，476k模型，被157k企业使用。 因此，我们期待与PaddlePaddle一起开发的PaddleSpeech能够消除英语和中文社区之间的障碍，从而促进语音技术和应用的发展。

为行业开发语音应用程序与在学术界进行研究不是同一种情况。研究论文主要集中在开发新的模型，以更好地执行特定的数据集。但是，在应用语音产品时通常不存在干净的数据集。因此，PaddleLanguage为原始音频提供了即时预处理，使PaddleLanguage可以直接用于面向产品的应用程序。值得注意的是一些预处理方法在 PaddleLanguage 中是独占的，例如基于规则的中文文本到语音前端，能够显著提高合成语音的性能。

性能是所有应用程序的基石。PaddleSpeech达到了在各种常用基准上最先进的亦或是具有竞争力的表现，如表1所示。



我们在本文中的主要贡献是两方面的

我们将介绍如何设计PaddleSpeech以及它支持哪些功能。

我们提供实现和可重现的实验细节，产生在各种任务中最先进的亦或有竞争力的表现。

**2 Design of PaddleSpeech**

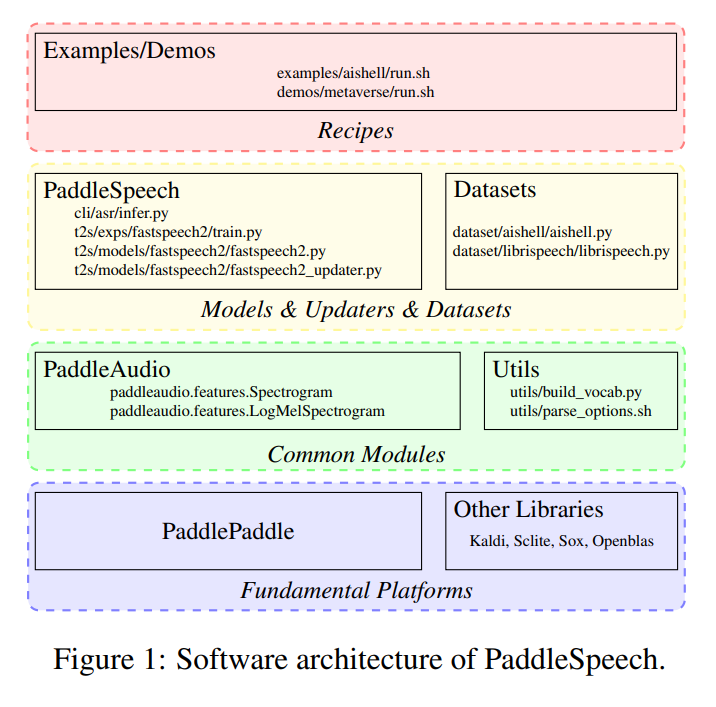


图1显示了PaddleSpeech的软件架构。作为一个易于使用的语音处理工具包，PaddleSpeech提供了许多完整的决策来执行各种与语音相关的任务，并演示命令行界面的使用。熟悉顶层就足够构建与语音相关的应用程序。

第二层面向的是语音和语言处理方面的研究人员。PaddleSpeech的设计理念以模型为中心，简化语音处理方法的学习和开发。对于特定的方法，特定模型的所有计算都包含PaddleSpeech/<task>/models/<model>下的两个文件中。

PaddleSpeech已经实现了大多数常用的和性能良好的模型。模型体系结构在一个独立的文件中实现，该文件由方法命名。相应的训练步骤和评估步骤在另一个更新程序文件中实现。通常，阅读或洞悉这两个文件就足以理解或设计模型。更高级的洞悉关于更精细的数据处理或更复杂的训练/评估循环，也可在 PaddleSpeech/任务></经验/><模型>。原始数据集可通过对应dataset/<dataset>/中的脚本获取。PaddleSpeech支持分布式多gpu高效率训练。

标准模块例如音频和文本特性转换和实用程序脚本在第三层中作为库实现。PaddleSpeech的后端主要是PaddlePaddle，部分函数来自第三方库，如第四层所示。PaddleSpeech提供了多种方法，使用PaddleAudio和Kaldi从原始音频中提取多种类型的语音特征，例如频谱图和滤波器组，可以根据任务的需求进行更改。

**3 实验**

在这部分，我们将PaddleSpeech中的模型与其他流行实现在五个与语音相关的任务中的性能进行比较，包括声音分类、语音识别、发音、语音翻译和语音合成。工具箱可以在大多数任务上达到SOTA。本节中的所有实验都包括数据准备、评估指标和提高可重复性的实施细节。

3.1声音分类

声音分类是一项识别特定声音的任务，包括语音命令(Warden,2018)，环境声音(Piczak, 2015)，识别乐器(Engel et al.,2017)，寻找鸟鸣(Stowell et al.,2018)，情感识别(Xu et al.,2019)和说话人验证(Liu et al.,2018)。

数据集

在这部分中，我们分析了在ESC-50数据集(Piczak, 2015)上的声音分类中PaddleSpeech的性能。ESC-50数据集是2000个5秒环境音频记录的标记集合，包括50个声音事件，如“狗”、“猫”、“呼吸”和“烟花”，每个事件40个录音。

数据处理

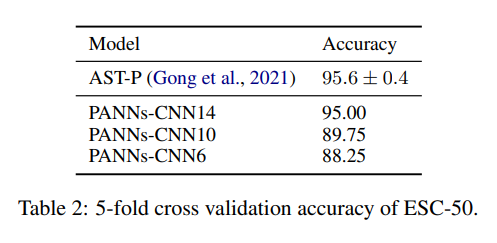
首先，我们将所有音频录音重新采样到32kHz，并将其转换为单声道以与AudioSet上训练的PANNs一致(Kong et al.,2020b)。然后，我们通过对汉明窗大小为1024，跳跃步长为320样本的波形进行短时傅里叶变换，将录音转换为log mel频谱图。这种配置导致每秒100帧。继Kong et al.(2019)之后，我们使用64mel滤波器组来计算log mel谱图。

实现

PANNs (Kong et al.,2020b)是用于音频相关任务的预训练CNN模型之一，其特征在于使用AudioSet (Gemmeke et al.,2017)。PANNs对于只提供有限数量的训练片段的任务很有帮助。在这种情况下，我们为环境声音分类任务微调PANN的所有参数。所有参数都从PANN初始化，除了最后的全连接层是随机初始化的。具体来说，我们分别用6层、10层和14层实现了CNNs(Kong et al.， 2020b)。

结果

我们报告了在ESC-50数据集上的5倍交叉验证精度值。如表2所示，PANNs-CNN14达到0.9500 5倍交叉验证精度，与目前最先进的方法相当(Gong et al.,2021)。



3.2 自动语音识别

自动语音识别(ASR)是一种将音频内容转录为相同语言的文本的任务。

数据集

我们在Librispeech(Panayotovet al.,2015)和Aisell -15(Bu et al.,2017)两个主要数据集上进行ASR实验。Librispeech包含1000小时语音数据。整个数据集分为3个训练集(100h清洗，360h清洗，500h其他)，2个验证集(清洗，其他)和2个测试集(清洗，其他)。Aishell包含178小时语音数据。来自中国不同口音地区的400名演讲者参与了录音。数据集分为训练集(340人)、验证集(40人)和测试集(20人)。

数据预处理

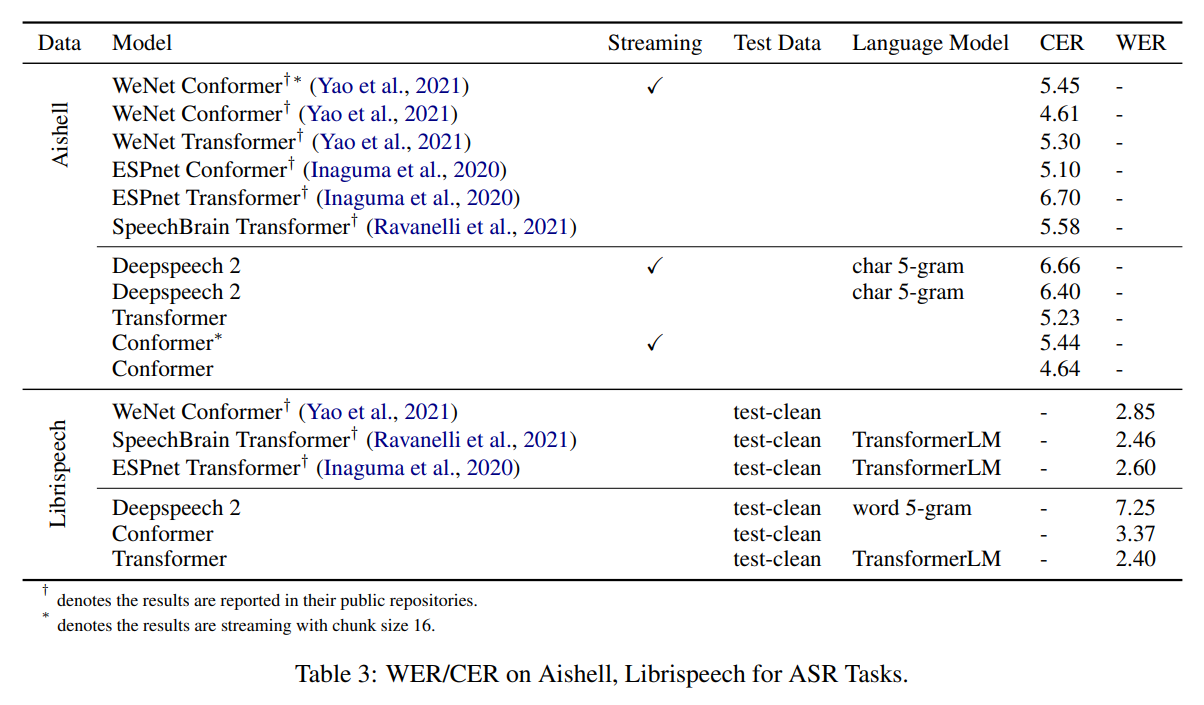
Deepspeech 2 采用两种英语的字符级词汇和普通话任务。对于其他模型，我们使用汉字级词汇。并用SentencePiece(Kudo And Richardson, 2018)对英语文本进行预处理。这两种数据集都增加了四个额外的字符，分别是<’>, <space>, <blank> and <eos>。对于倒谱均值和方差归一化(CMVN)，选择训练集的一个子集或全部子集，用于计算特征均值和标准误差。对于特征提取，我们实现了几种方法，如线性频谱图，滤波器组和梅尔频率倒谱系数。目前，Deepspeech 2 模型使用线性谱图或滤波器组，但Transformer和Conformer模型使用滤波器组。为了公平的比较，我们在Transformer中加入了额外的三维音高特征，以与ESPnet保持一致。

实现

我们实现了流媒体和非流媒体Deepspeech 2 (Amodei et al., 2016)。非流模型有2个卷积层和3个LSTM层。流媒体 该模型有2个卷积层和5个LSTM层。Conformer和Transformer模型是根据Zhang et al(2020)实现的，具有12个编码器层和6个解码器层。

结果

我们分别报告了Librispeech(英语)和Aishell(普通话)语音识别的单词错误率(WER)和字符错误率(CER)。如表3所示，Conformer和Transformer优于Deepspeech 2。与相关工作相比，我们的最佳模型在两个数据集上都实现了相当的性能。



3.3 标点符号恢复

标点恢复是ASR系统的后处理问题。提高转录文本对人类读者的可读性和促进下行NLP任务至关重要。

数据集

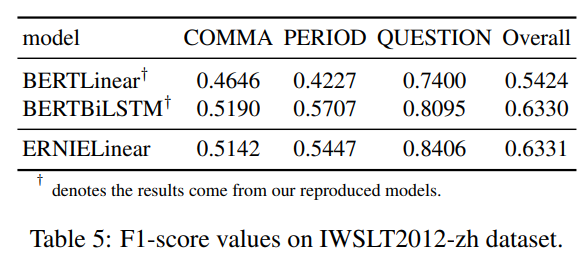
我们在包含150k汉语标点句子的IWSLT2012-zh6数据集上进行了实验。在这个任务中，我们选择逗号、句号和问号作为恢复目标，因此在训练模型之前，我们用这三个标记替换其他标点符号。我们将数据分为训练集、验证集和测试集，分别使用147k、2k和1k样本。

实现

我们将标点恢复问题定义为具有四个目标类的序列标记任务，包括EMPTY、COMMA、PERIOD和QUESTION (Nagy et al., 2021b)。ERNIE (Sun et al., 2019)作为一种预训练的语言模型，在五个中文自然语言处理任务上取得了最新的结果，包括:自然语言推理、语义相似性、命名实体识别、情感分析和问题回答。因此，我们为这项任务调整了ERNIE模型。更具体地说，所有参数都从ERNIE预训练的模型初始化，除了最后的共享全连接层是随机初始化的。

结果

我们在IWSLT2012-zh数据集上报告F1-score值。如表5所示，我们的ERNIELinear模型总体f1得分为0.6331，与之前的工作(Nagy et al., 2021a)相当。



3.4 语音翻译

语音翻译，即将源语言的语音翻译成另一种语言的文本，有利于人类交流。

数据集

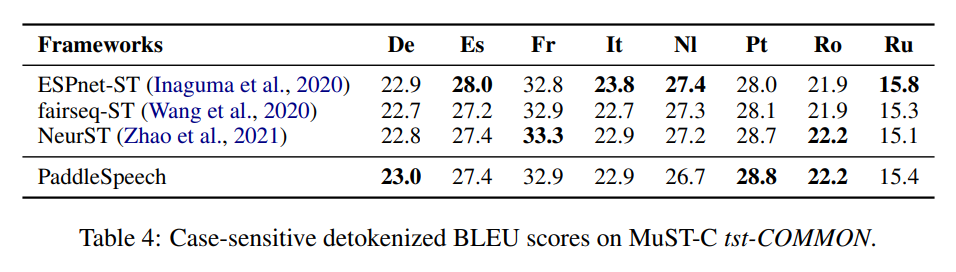
在这部分，我们分析了使用PaddleSpeech在MuST-C数据集(Di Gangi et al., 2019)上使用8种不同的语言翻译对进行语音到文本翻译的性能，这些语言翻译对采用英语语音作为源输入。

实现

我们使用Kaldi (Povey et al., 2011)处理原始音频，并使用25ms的窗口大小和10ms的步长提取80维的log-mel滤波器组。文本首先使用Moses tokenizer7进行标记，然后由SentencePiece (Kudo and Richardson, 2018)使用每个语言对大小为8K的联合词汇表进行处理。我们使用Transformer (Vaswani et al., 2017)作为语音翻译实验的基础架构。具体来说，Transformer模型有12层编码器层和6层解码器层，编码器层遵循2层二维卷积，核大小为3，步幅大小为2。每一层包含4个注意头，大小为256。编码器由预训练的ASR模型初始化。

结果

我们报告解标记区分大小写的BLEU。如表4所示，与其他框架相比，PaddleSpeech可以获得具有竞争力的结果。



3.5 文本转语音

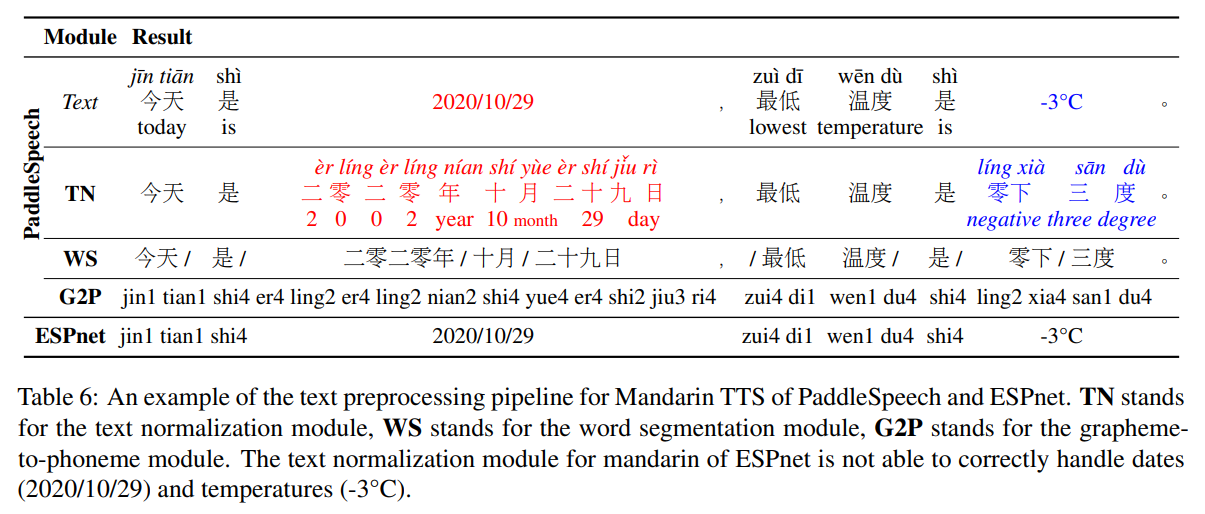
文本到语音(TTS)系统将给定的语言文本转换为语音。PaddleSpeech的TTS管道包括三个步骤。我们首先通过文本前端模块将原始文本转换为字符/音素。然后，通过声学模型，我们将字符或音素转换为声学特征，如mel谱图，最后，我们通过声码器从声学特征生成波形。在PaddleSpeech中，文本前端是一个受专家知识启发的基于规则的模型。声学模型和声码器是可训练的。

数据集

在PaddleSpeech中，我们主要关注普通话和英语语音合成。我们有CSMSC, AISHELL-3, LJSpeech, VCTK的基准测试。由于篇幅所限，我们只在CSMSC上列出了实验结果，其中包含了12小时的演讲音频，对应10k个句子。

文本前端

文本前端模块用于从给定文本中提取语言特征、字符和音素。主要包括:文本分割、文本归一化(TN)、单词分割(WS)、词性标注、韵律预测和字母到音素(G2P)(见表6)。



对于普通话，我们的G2P系统包括一个复音模块，它使用pypinyin和g2pM，以及一个变调模块，它使用基于中文分词的规则。据我们所知，与其他公开发布的作品相比，我们的中文文本前端系统是最完整的。

数据预处理

PaddleSpeech TTS使用以下模块进行数据预处理:首先，我们使用蒙特利尔强制对齐器来获取对应音素的持续时间。其次，提取mel谱图作为特征(Fastspeech 2的额外音调和能量特征)。最后，对每个特征进行统计归一化。

声学模型

声学模型主要分为自回归模型和非自回归模型。自回归模型的解码依赖于每一步之前的预测，这导致更长的推理时间，但相对更好的质量。而非自回归模型是并行生成输出，因此推理速度较快，但生成的结果质量相对较差。

如表1所示，PaddleSpeech实现了以下常用的自回归声学模型:Tacotron 2和Transformer TTS，以及非自回归声学模型:SpeedySpeech, FastPitch和FastSpeech 2。

语音编码器

如表1所示，PaddleSpeech实现了以下声码器:WaveFlow, Parallel WaveGAN, MelGAN, Style MelGAN, Multi - Band MelGAN和HiFi GAN。

实现

结果