结合 CTC 损失与自监督预训练声学编码器的端到端口语理解

Jixuan Wang Martin Radfar KaiWei Clement Chung

Amazon Alexa AI

arXiv:2305.02937v2 [cs.CL] 2 Jun 2023

摘要

由于缺乏文本信息，口语理解（SLU）难以直接从音频信号中提取语义。常见的端到端（E2E）SLU 模型采用序列到序列自动语音识别（ASR）模型提取文本嵌入，并将其作为输入来推断语义，但这需要计算开销巨大的自回归解码。本文利用经连接时序分类（CTC）微调的自监督声学编码器提取文本嵌入，并将 CTC 损失与 SLU 损失联合用于话语级口语理解任务。实验表明，在 DSTC2数据集上，我们的模型相比最先进水平的对话行为分类模型绝对提升了 4%；在 SLURP数据集上，相比最先进水平的 SLU 模型绝对提升了 1.3%。

索引词— 口语理解、意图分类、对话行为分类

1. 引言

口语理解（SLU）模型从口语话语中推断语义 [1–3]。常见的口语理解任务包括意图分类、槽位填充和对话行为分类。传统上，口语理解模型由一个将音频信号转写为文本的自动语音识别模型（ASR）和一个从文本中提取语义的自然语言理解（NLU）模型组成 [4]。由于自动语音识别模型和自然语言理解模型是独立优化的，自动语音识别组件产生的错误会传播到自然语言理解组件。例如，如果自动语音识别模型将“打开电视”识别为“关闭电视”，自然语言理解模型就很难预测出正确的意图。

近年来，人们对构建端到端口语理解模型的兴趣日益浓厚。这类模型联合优化声学模型和文本模型，从而使口语理解模型更加稳健 [5–16]。早期工作 [5, 6] 直接从音频信号中学习话语级语义表示，而不进行语音识别。由于缺少文本信息时语义信息可能会丢失，因此这项任务颇具挑战性。为引导从音频信号中学习语义表示，研究人员采用多模态损失，将预训练BERT模型和声学编码器生成的话语级嵌入联系起来 [7]。词元级跨模态对齐和联合空间学习在 [8]中进行了研究。

上述方法的局限性在于，它们无法用于序列标注任务，例如槽位填充。为了解决这一问题，另一类研究构建了统一模型，这些模型可以进行端到端训练，并同时用于意图分类和槽位填充。实现端到端训练的一种方法是将口语理解重新定义为序列到序列任务，其中，语义标签被视为除转录文本 [9–12] 之外的另一种输出标签序列。另一种方法是统一自动语音识别和自然语言理解模型，并通过可微分神经接口对其进行联合训练 [13–16]。一种常用的神经接口是将来自自动语音识别的词元级隐藏表示作为自然语言理解模型的输入 [13–16]。[14, 15] 采用具有共享词汇表的预训练自动语音识别模型和自然语言理解模型。文献 [16] 对不同的神经接口进行了比较，并提出了一种用于基于 RNN-Transducer（RNN-T）的自动语音识别模型的新型接口。然而，为了在推理时间生成词元级表示，这些方法需要自回归解码，而这在计算上代价高昂。

在采用 CTC 损失训练的自动语音识别模型中，标签是在音频帧级别并行预测的，其效率高于需要自回归解码的方法 [17,18]。在本研究中，我们将基于 CTC 的自动语音识别模型的输出作为输入，以推断语义，并通过联合 CTC 和口语理解损失对模型进行端到端训练。我们证明，与直接从音频中推断语义而不依赖自动语音识别监督的方法，以及依赖自回归自动语音识别模型的方法相比，我们的方法具有更优的性能。与 [19]，中提出的方法相比，我们的研究展示了使用通过自监督任务预训练的声学编码器进行端到端口语理解的有效性，并强调了联合使用 CTC 损失和口语理解损失进行训练，以及使用 Logits 而非概率作为输入来提取话语嵌入的重要性。我们证明，这种简单的方法在三个公开数据集上针对三种不同任务均取得了最先进的结果。值得注意的是，我们的模型在 SLURP 数据集上的意图分类准确率比目前报告的最佳结果高出 1.3%。

2. 使用 CTC 的端到端口语理解

如图 1 所示，我们的模型主要由两部分组成：一个自动语音识别模型和一个话语编码器。该自动语音识别模型

该模型基于通过自监督任务预训练并使用连接时序分类进行微调的声学编码器。在本研究中，我们采用了预训练的 Wav2Vec2.0 [20]和 HuBERT [21] 模型。自动语音识别模型的输出序列经过最大池化处理后，输入基于全连接层的话语编码器。整个模型可以进行端到端训练。

我们将一段话语的音频序列表示为 $X=(x_1, x_2, \ldots, x_T)$，相应的转录文本表示为 $W=(w_1, w_2, \ldots, w_U)$，其中 T和 U分别是声学特征和转录文本的序列长度。每段话语都标注有话语级标签 y，例如意图或对话行为。训练数据表示为$D^{\text{tr}}=\{(X_i, W_i, y_i)\}_{i=1}^{|D^{\text{tr}}|}$。为了在 D^{tr} 上训练自动语音识别模型，我们希望最大化相对于训练数据分布 p^{tr} 的期望条件概率 $\mathbb{E}_{(X,W)\sim p^{tr}} p(W|X)$，该概率近似为 $\sum_{i=1}^{|D^{\text{tr}}|} p(W_i|X_i)$。

连接时序分类不假设输入序列和输出序列之间的对齐已知，而是在计算训练损失时考虑所有可能的对齐。对于每一对（X，W），我们将 $p(W|X)$ 计算为所有有效对齐 $\mathcal{A}_{X,\,W}$ 的概率之和：

$$p(W|X)=\sum_{A\in\mathcal{A}_{X,W}} p(A|X) \tag{1}$$

其中 $A=(a_1, a_2, \ldots, a_T)$，其中每个 a_i都可以取自所有可能输出词元的集合以及一个特殊词元ϵ，后者表示空白符号。通过从对齐 A中移除重复词元和 ϵ，我们可以恢复输出序列 W，此时 A 被视为有效对齐。

为了计算 $p(A|X)$，我们使用声学编码器提取帧级隐藏表示序列 $H=(h_1, \ldots, h_{T'})$。需要注意的是，如果声学编码器包含下采样层，则 T'会小于 T，但为简化符号表示，我们假设 $T'=T$。帧级预测如下：

$$p(a_i|X)=softmax(Wh_i+b), i=\{1, 2, \ldots, T\}. \tag{2}$$

其中，W和 b是线性分类器的参数。

条件概率 $p(W|X)$可以通过动态规划高效计算，并作为训练目标来优化自动语音识别模型。自动语音识别目标定义为：

$$\mathcal{L}^{\text{CTC}}=\frac{1}{|D^{\text{tr}}|}\sum_{(W,X)\in D^{\text{tr}}} -\log p(W|X), \tag{3}$$

为了预测意图分类和对话行为分类等任务的话语级标签，我们将帧级Logits（$Wh_1+b, \ldots, Wh_T+b$）作为话语编码器的输入。我们还尝试将隐藏表示 H 作为话语编码器的输入。在实验中，我们观察到这两种方法的表现相近。

将话语编码器的输入记为 $H^u=(h_1^u, \ldots, h_T^u)$。我们的 utterance encoder 包含一个最大池化

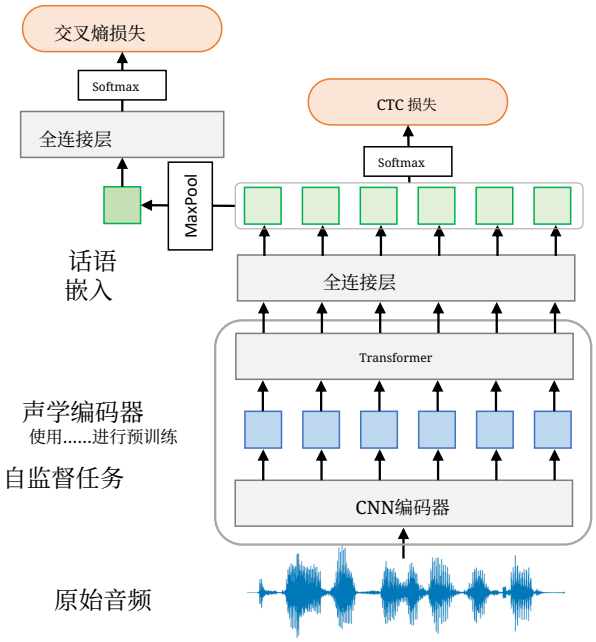

图1。所提出方法的框图。

层和两个全连接层：

$$\begin{aligned} h^{pool} &= maxpool(h_1^u, \ldots, h_T^u) \\ h^{utt} &= g(W^{(2)}g(W^{(1)}h^{pool}+b^{(1)})+b^{(2)}) \end{aligned} \tag{4}$$

其中，$W^{(1)}$，$b^{(1)}$和 $W^{(2)}$，$b^{(2)}$分别是两个全连接层的参数，g 是激活函数。对于分类，我们使用线性分类器和交叉熵损失进行训练：

$$\begin{aligned} p(y|X) &= softmax(W^u h^{utt}+b^u) \\ \mathcal{L}^{\text{SLU}} &= \frac{1}{|D^{\text{tr}}|}\sum_{(y,X)\in D^{\text{tr}}} -\log p(y|X). \end{aligned} \tag{5}$$

我们采用多任务方式训练整个模型，使用 $\mathcal{L}^{\text{CTC}}$ 和 $\mathcal{L}^{\text{SLU}}$的线性组合：

$$\mathcal{L}=\alpha^{\text{CTC}}\mathcal{L}^{\text{CTC}}+\alpha^{\text{SLU}}\mathcal{L}^{\text{SLU}} \tag{6}$$

其中，我们将标量 α^{CTC} 和 α^{SLU} 视为超参数。

良好的自动语音识别性能是理解语义的关键，因此我们首先在训练期间使用领域内数据微调自动语音识别模型，方法是设置 $\alpha^{\text{SLU}}=0, \alpha^{\text{CTC}}=1$。当自动语音识别模型在验证集上的表现不再提升后，我们设置 $\alpha^{\text{SLU}}>0, \alpha^{\text{CTC}}>0$，根据验证表现调整这些值，并以端到端方式训练整个模型。

3. 实验

我们在对话行为分类、关键词检出和意图分类任务上进行了实验，其中输入】【。

输入为音频信号，输出为话语级标签。

3.1. 数据集

我们使用了表1中列出的三个公开数据集，其中展示了每个数据集用于训练、验证和测试的话语数量。

对话行为分类： 我们使用DSTC2数据集 [22]来完成此任务，其中每个话语都标注了包含动作类型、槽名称和槽值的对话行为。与 [23], 类似，我们专注于话语级分类，不考虑同一对话中的相邻话语，并将同一话语中出现的多个标签合并为一个标签。

关键词检测： 我们使用Google语音命令（GSC）数据集 V2 [24], , 该数据集总共包含105K个音频片段。每个音频片段时长为1秒，并包含一个单词。总共有35个不同的单词，这些单词也被用作标签。目标是根据音频片段预测该单词。

意图分类： 对于此任务，我们使用SLURP数据集 [25]。SLURP是一个具有挑战性的数据集，其中的话语在语言上更加多样化，并且涵盖18个领域，拥有更大的标签空间。每个话语都标注了一个场景 和动作 标签。遵循先前的研究 [25, 26], , 我们将意图 定义为场景 和 动作的组合。训练集中总共有69个不同的意图。

3.2. 基线方法

对于对话行为分类任务，我们采用 [23] 中提出的模型作为基线，该模型是针对该数据集的最新研究成果。作者提出了一种用于对话行为分类的神经韵律编码器，以韵律特征和原始音频信号作为输入。我们遵循相同的流程来准备DSTC2数据集。

对于关键词检测任务，我们与一项名为HTS-AT [27], 的近期工作进行比较，该模型是基于Transformer的模型，具有层次结构。

对于意图分类，我们与以下最先进模型进行比较。*ESPNET-SLU*： ESPnet-SLU工具包为SLURP提供了多种口语理解配方 [26]。其口语理解模型是一个序列到序列模型，其中意图被解码为一个词。我们纳入了其使用预训练HuBERT作为特征提取器在SLURP上取得的结果。*Seo*等人, *2022*： 该工作使用了一个基于Wav2Vec2.0的自动语音识别模型 [15]。解码器的输出被输入基于RoBERTa的自然语言理解模型，用于意图分类和槽位填充。该模型的每个组件首先分别进行预训练，然后进行联合微调。我们将其在原始训练集和合成训练集上取得的结果都纳入比较。

3.3. 实现与超参数

我们的方法可以受益于任何使用 CTC 损失训练的预训练自动语音识别模 型。我们利用预训练的

表1。数据集信息。

Task	数据集	训练集	验证集	Test
DAC	DSTC2	12,930	1,437	9,116
KWS	Speech Commands	84,843	9,981	11,005
IC	SLURP-Synth	119,881	8,690	13,078

Wav2Vec2.0 [20] 和 HuBERT [21] 声学编码器使用 CTC 进行了微调，这与先前的最先进的口语语言理解模型 [15, 26] 所采用的方法相同。根据数据集的大小，我们使用不同规模的预训练自动语音识别模型。我们使用了 TorchAudio [28] 提供的预训练自动语音识别模型。对于 DSTC2 和 Speech Commands 等较小的数据集，我们使用 TorchAudio 中可用的最小模型“WAV2VEC2 ASR BASE 960H” [20]。对于更具挑战性的 SLURP 数据集，我们使用“HUBERT ASR LARGE” [21]。这两个模型都在未标注的音频数据上进行了预训练，并在 LibriSpeech 数据集 [29] 上进行了微调。作为话语编码器的输入，我们使用最大池化，后接两个全连接层，每层有 128 个隐藏单元，并使用 GELU 激活函数 [30]。

在微调自动语音识别模型时，如果连续 5 个轮次的 ASR损失没有改善，我们就停止训练。在联合训练期间，我们进行 50 个轮次的训练，并选择在验证集上表现最佳的检查点。由于我们研究的任务均为话语级分类任务，因此使用准确率作为评估指标。我们根据验证集上的性能调整学习率、批量大小和 α^{CTC} 。对于 DSTC2，我们使用 0.00001 的学习率、16 的批量大小以及 0.5 的 α^{CTC} 。对于 Speech Commands，我们使用 0.00001 的学习率、128 的批量大小以及 1.0 的 α^{CTC}。对于 SLURP，我们使用 0.00005 的学习率、128 的批量大小以及 0.5 的 α^{CTC} 。我们使用 PyTorch 中实现的 AdamW 优化器，并采用其默认配置。

4. 实验结果

4.1. 与既有工作的比较

三个数据集上的结果如表 2 所示。

对话行为分类： 我们的模型分别以隐藏表示和帧级 Logits作为输入，在 DSTC2 上达到了 97.6% 和 97.5% 的准确率，较之前直接从音频输入预测对话行为的工作 [23]高出 4%。

关键词检测：由于先前的研究已经取得了非常出色的性能，因此 GSC V2 上的提升空间很小。尽管我们的方法并非专为关键词检测而设计，但其性能仍可与 HTS-AT [27], 相媲美，准确率达到 98.0%。需要注意的是，GSC V2 中的所有话语都只包含一个单词，因此，我们的方法中自动语音识别模型所恢复的文本信息可能不像在其他数据集（如 DSTC2 和 SLURP）中那样有帮助，因为后者包含语言结构更加复杂的话语。

表 2。DSTC2、Speech Commands 和 SLURP 上的测试集结果。*数据来自原论文。

数据集	方法	准确率
DSTC2	Wei 等人， 2022 [23]	93.6*
	我们的方法（Wav2Vec2.0）	
	+以隐藏表示作为输入	**97.6**
	+ Logits 作为输入	97.5
GSC	HTS-AT [27]	98.0*
	我们的方法（Wav2Vec2.0）	
	+隐藏表示作为输入	98.0
	+ 将 Logits 作为输入	98.0
SLURP	ESPnet-SLU [26]	86.3*
	Seo等人， 2022 [15]	86.9*
	我们的方法（HuBERT）	
	+隐藏表示作为输入	88.1
	+ Logits 作为输入	**88.2**

意图分类：在 SLURP数据集上，我们的方法分别使用隐藏表示和帧级Logits作为输入，取得了 88.1% 和 88.2% 的准确率，相较于最先进结果绝对提升了 1.3%。

4.2. 消融与分析

我们在SLURP数据集上进行消融研究，以探究：1）我们的端到端模型是否能够比级联模型表现更好；2）CTC 损失对端到端口语理解建模是否有用；3）是否有必要针对口语理解对自动语音识别模型进行微调；4）我们是否需要更强大的话语编码器。结果列于表3中。

HuBERT + 双向长短期记忆网络/BERT 自然语言理解： 我们使用微调后的 HuBERT 自动语音识别模型生成的最佳假设来训练基于双向长短期记忆网络和 BERT 的自然语言理解模型。双向长短期记忆网络和 BERT 自然语言理解模型的参数数量分别为 24.7M 和 110M。我们的模型的文本编码器仅包含两个全连接层（20K 个参数），规模远小于双向长短期记忆网络和 BERT 模型，并且从头开始训练，没有像 BERT 那样在文本数据上进行预训练。尽管如此，我们的方法仍优于这两种级联方法：这两种方法的测试准确率分别为 83.95% 和 87.44%，而我们的方法为 88.18%。

HuBERT + 不使用 ASR 损失的线性分类器： 该方法与我们提出的方法之间唯一的区别在于，该方法不使用 CTC 损失进行口语理解，而是直接根据音频信号预测口语语言理解标签。该方法在测试集上的准确率为 84.81%，低于我们的方法，也低于此前基于 LAS 自动语音识别模型 [15] 的研究结果。这一结果证明了 ASR 损失所恢复的文本信息对于口语理解的重要性。

HuBERT（冻结） +话语编码器： 在使用域内数据对自动语音识别模型进行微调后，我们不再使用口语理解任务对其进一步微调，而是冻结自动语音识别模型。

表3. SLURP 上的消融研究结果。

方法	准确率
HuBERT + 双向长短期记忆网络 自然语言理解	83.95
HuBERT + BERT 自然语言理解	87.44
HuBERT + 不使用ASR损失的线性分类器	84.81
HuBERT（冻结） + 话语编码器	72.16
以概率作为输入的 HuBERT	87.00
使用 CNN 文本编码器的 HuBERT	88.05
我们的方法（以 Logits 作为输入的 HuBERT）	88.18

该模型仅训练话语编码器。这种方法的准确率仅为 72.16%。这表明，以端到端的方式进一步微调自动语音识别模型，使其适用于口语语言理解任务，是取得良好性能的关键因素之一。

以概率作为输入的 HuBERT与先前的工作 [19], 类似，我们尝试使用 softmax层之后的概率作为话语编码器的输入。这种方法的表现不如使用 Logits 或隐藏表示作为输入，准确率为 87.0%。

使用 CNN 文本编码器的 HuBERT：我们探究采用更强大的基于 CNN 的话语编码器能否提升性能。我们在 Logits 之上应用四个具有不同卷积核尺寸的 CNN 层，并通过最大池化和全连接层提取话语级嵌入。我们发现，尽管 CNN 编码器（190 万个参数）远大于我们的话语编码器（2 万个参数），但使用 CNN 编码器时的准确率为 88.05%，略低于我们的方法。

我们进一步分析了自动语音识别的性能。进行口语理解训练前，自动语音识别模型的词错误率和字符错误率分别为 18.2% 和 8.5%；进行口语理解训练后，分别降至 17.4% 和 7.8%。这表明口语理解损失与自动语音识别损失并不冲突，反而有助于提升自动语音识别性能。另一方面，对于包含自动语音识别错误的语句，我们的模型仍能正确预测其中 83.4% 的意图标签，这表明我们的方法能够抵御自动语音识别错误。

5. 结论

在这项工作中，我们研究了将基于CTC的ASR模型用于话语级SLU任务。实验结果表明，采用CTC损失和SLU损失进行联合训练，能够在多个数据集上取得最先进结果。借助预训练声学编码器，一个基于小型全连接层的话语编码器也能实现出色的性能。我们的方法同样无需自回归解码，因此推理时间效率很高。未来，我们将研究如何扩展这一框架以适用于序列标注任务，例如槽位填充。关键问题在于，如何在不进行自回归解码的情况下，以可微且高效的方式从基于CTC的ASR模型输出中恢复实体名称。

6. 参考文献

[1] Y.-Y. Wang、L. Deng 和 A. Acero，《口语理解》，*IEEE*《信号处理杂志》，第 22 卷，第 5 期，第 16–31 页，2005 年。[2] G. Tur 和 R. D. Mori，《口语理解：从语音中提取语义信息的系统》，*Wiley*，2011 年。

[3] A. Bhargava、Saffikaya、Celikyilmaz 和 Hakkani-Tur，“轻松实现上下文意图预测和槽位检测”，*2013*年 *IEEE* 声学、语音与信号处理国际会议，第 8337–8341 页，2013年。

[4] M. Larson、G. J. Jones等，“口语内容检索：技术与方法综述”，信息检索基础与趋势®，第 5 卷，第 4–5 期，第 235–422 页，2012年。

[5] L. Lugosch、M. Ravanelli、P. Ignoto、V. S. Tomar 和 Y. Bengio，“用于端到端口语理解的语音模型预训练”，*arXiv* 预印本 *arXiv:1904.03670*，2019年。

[6] D. Serdyuk、Y. Wang、C. Fuegen、A. Kumar、B. Liu 和 Y. Bengio，“迈向端到端口语理解”，载于 *2018*年 *IEEE*声学、语音与信号处理国际会议（*ICASSP*）。IEEE，2018年，第 5754–5758 页。

[7] B. Agrawal, M. M¨uller, S. Choudhary, M. Radfar, A. Mouchtaris, R. McGowan, N. Susanj, 和 S. Kunzmann，“固定嵌入：用于端到端口语理解的跨模态潜在空间”，载于*ICASSP 2022-2022 IEEE*声学、语音与信号处理国际会议 *(ICASSP)*。IEEE，2022年，第 7157–7161 页。

[8] M. Kim、G. Kim、S.-W. Lee 和 J.-W. Ha，“St-bert：用于端到端口语语言理解的跨模态语言模型预训练”，载于 *ICASSP 2021-2021 IEEE International Conference on Acous- tics, Speech and SignalProcessing (ICASSP)*。IEEE，2021年，第 7478–7482 页。

[9] P. Haghani、A. Narayanan、M. Bacchiani、G. Chuang、N. Gaur、P. Moreno、R. Prabhavalkar、Z. Qu 和 A. Waters，“从音频到语义：端到端口语理解方法”，载于*2018*年 *IEEE* 口语技术研讨会（*SLT*）。IEEE，2018年，第720–726页。

[10] H.-K.J. Kuo、Z. T¨uske、S. Thomas、Y. Huang、K. Audhkhasi、B. Kingsbury、G. Kurata、Z. Kons、R. Hoory 和 L. Lastras，《无需完整转录文本的端到端口语语言理解》，*arXiv* 预印本 *arXiv:2009.14386*，2020年。

[11] M. Radfar、A. Mouchtaris、S. Kunzmann 和 A. Rastrow，“粉丝：Fused-端到端神经transformer S模型的口语语言理解”，*arXiv* 预印本 *arXiv:2008.10984*，2020年。

[12] M. Radfar、A. Mouchtaris、S. Kunzmann 和 A. Rastrow，“粉丝：Fus - 用于设备端口语理解的自动语音识别和自然语言理解，”*arXiv* 预印本 *arXiv:211100400*，2021年.。

[13] M. Rao、A. Raju、P. Dheram、B. Bui、和 A. Rastrow，“语音到 se - 义：通过全神经接口联合改进自动语音识别和自然语言理解，”*arXiv* 预印本 *arXiv:2008.06173*，2020年。

[14] M. Saxon、S. Choudhary、J. P. McKenna 和 A. Mouchtaris，《面向通用语音助手的端到端口语理解》，*arXiv* 预印本 *arXiv:2106.09009*，2021年。[15] S. Seo、D. Kwak 和 B. Lee，《将预训练网络与连续令牌接口相结合，实现端到端口语理解》，载于 *ICASSP 2022-2022 IEEE* 国际声学、语音与信号处理会议（*ICASSP*）。IEEE，2022年，第 7152–7156 页。[16] A. Raju、M. Rao、G. Tiwari、P. Dheram、B. Anderson、Z. Zhang、C. Lee、B. Bui 和 A. Rastrow，《关于采用接口进行口语理解的联合训练》，*Interspeech 2022* 论文集，1257，2022年，1253页。

[17] A. Graves，“使用循环神经网络进行序列转导”，*arXiv*预印本 *arXiv:1211.3711*，2012年。[18] W. Chan、N. Jaitly、Q. Le 和 O. Vinyals，“听、关注与拼写：一种用于大词汇量会话语音识别的神经网络”，载于*ICASSP*。IEEE，2016年，第4960–4964页。[19] Y.-P. Chen、R. Price 和 S. Bangalore，“无需语音识别的口语理解”，载于 *2018*年 *IEEE*声学、语音与信号处理国际会议（*ICASSP*）。IEEE，2018年，第6189–6193页。[20] A. Baevski、Y. Zhou、A. Mohamed 和 M. Auli，“wav2vec 2.0：一种用于语音表示自监督学习的框架”，神经信息处理系统进展，第33卷，第12 449–12 460页，2020年。[21] W.-N. Hsu、B. Bolte、Y.-H. H. Tsai、K. Lakhotia、R. Salakhutdinov 和 A. Mohamed，“HuBERT：通过对隐藏单元进行掩码预测实现自监督语音表示学习”，*IEEE/ACM*音频、语音与语言处理汇刊，第29卷，第3451–3460页，2021年。[22] M. Henderson、B. Thomson 和 J. D. Williams，“第二届对话状态跟踪挑战赛”，载于第*15*届话语与对话特别兴趣组（*SIGDIAL*）年度会议论文集，2014年，第263–272页。[23] K. Wei、D. Knox、M. Radfar、T. Tran、M. M¨uller、G. P. Strimel、N. Su- sanj、A. Mouchtaris 和 M. Omologo，“用于端到端对话行为分类的神经韵律编码器”，载于 *ICASSP 2022*年 *IEEE*声学、语音与信号处理国际会议（*ICASSP*）。IEEE，2022年，第7047–7051页。[24] P. Warden，“Speech Commands：一个用于有限词汇语音识别的数据集”，*arXiv*预印本 *arXiv:1804.03209*，2018年。[25] E. Bastianelli、A. Vanzo、P. Swietojanski 和 V. Rieser，“SLURP：一个口语理解资源包”，*arXiv*预印本 *arXiv:2011.13205*，2020年。[26] S. Arora、S. Dalmia、P. Denisov、X. Chang、Y. Ueda、Y. Peng、Y. Zhang、S. Kumar、K. Ganesan、B. Yan 等，“ESPnet-SLU：通过ESPnet推进口语理解”，载于*ICASSP 2022*年 *IEEE*声学、语音与信号处理国际会议（*ICASSP*）。IEEE，2022年，第7167–7171页。[27] K. Chen、X. Du、B. Zhu、Z. Ma、T. Berg-Kirkpatrick 和 S. Dubnov，“HTS-AT：一种用于声音分类与检测的分层令牌语义音频变换器”，载于*ICASSP 2022*年 *IEEE*声学、语音与信号处理国际会议（*ICASSP*）。IEEE，2022年，第646–650页。[28] Y.-Y. Yang、M. Hira、Z. Ni、A. Chourdia、A. Astafurov、C. Chen、C.- F. Yeh、C. Puhrsch、D. Pollack、D. Genzel、D. Greenberg、E. Z. Yang、J. Lian、J. Mahadeokar、J. Hwang、J. Chen、P. Goldsborough、P. Roy、S. Narenthiran、S. Watanabe、S. Chintala、V. Quenneville-B´elair 和 Y. Shi，“TorchAudio：构建音频与语音处理的基础模块”，*arXiv*预印本 *arXiv:2110.15018*，2021年。[29] V. Panayotov、G. Chen、D. Povey 和 S. Khudanpur，“LibriSpeech：一个基于公有领域有声读物的自动语音识别语料库”，载于*2015*年 *IEEE*声学、语音与信号处理国际会议（*ICASSP*），2015年，第5206–5210页。[30] D. Hendrycks 和 K. Gimpel，“高斯误差线性单元（GELU）”，*arXiv*预印本 *arXiv:1606.08415*，2016年。