arXiv:2305.00926v1 [cs.CL] 1 May 2023

融合对话历史的口语理解任务联合建模

Siddhant Arora[1], *Hayato Futami*[2],*Emiru Tsunoo*[2],*Brian Yan*[1],*Shinji Watanabe*[1]

[1]卡内基梅隆大学， [2]索尼集团公司，日本

摘要

大多数人际互动都以口语对话的形式进行，其中某个话语的语义取决于上下文。口语对话中的每个话语都可以通过多种语义属性和说话人属性来表示，因此人们一直致力于构建口语理解（SLU）系统，以自动预测这些属性。近期研究表明，纳入对话历史有助于提升口语语言理解任务的性能。然而，每项口语语言理解任务都使用单独的模型，导致推理时间和计算成本增加。受此启发，我们希望探讨：能否在纳入上下文的同时，对所有口语语言理解任务进行联合建模，从而实现低延迟、轻量级的推理？为此，我们提出了一种新颖的模型架构，通过学习对话上下文，联合预测口语话语的意图、对话行为、说话人角色和情绪。需要注意的是，我们的联合预测基于自回归模型，因此必须确定对话属性的预测顺序，而这并非易事。为缓解这一问题，我们还提出了一种顺序无关训练方法。实验表明，我们的联合模型取得了与特定任务分类器相近的结果，并能有效整合对话上下文，进一步提升口语语言理解任务的性能。[1]

索引词— 口语语言理解、口语对话系统、端到端系统、联合建模、说话人属性

1. 引言

口语对话系统旨在使对话代理能够与人类进行更加自然的交流。它们通常通过一系列帧 [1, 2]来表示一种可能的对话。每个帧代表用户所寻求的任务类型，并包含能够帮助系统完成该任务的信息属性。这些口语对话系统旨在自动识别对话主题以及其他对话框架属性（例如对话行为、情绪），从而与用户进行交流。

传统的口语理解（SLU） [3, 4]系统会独立处理对话中的每个话语。然而，在口语对话中，话语的含义取决于上下文。已有研究表明，对话上下文在消解歧义和共指方面尤为有用 [5, 6]。因此，大量研究致力于结合上下文以提升自然语言理解（NLU）性能 [7–9]。先前关于对话语音的研究 [10–12] 也表明，结合对话上下文能够显著提升自动语音识别性能。因此，一些方法 [13–16] 尝试将对话历史融入基于流水线的口语理解系统。近期，也有一些研究 [17–19] 尝试结合上下文来提升端到端（E2E）口语理解性能。其中一种方法 [17]整合通过使用先前口语话语的自动语音识别转录文本，将对话历史融入端到端口语语言理解系统。另有研究 [18] 尝试直接从先前话语的音频中结合上下文。

然而，这些研究大多专注于为每项口语语言理解任务（如意图分类、对话行为分类和情感识别）构建单一模型。要联合预测所有口语语言理解任务，我们必须分别运行所有模型。因此，这些模型不仅内存占用较大，而且延迟较高，这会影响口语对话的自然度 [20]，尤其是在语音助手等商业应用中部署这些系统时。先前研究表明，在统一框架中联合建模不同的语音处理任务，性能可与特定任务模型相当 [21, 22]，同时还能降低延迟 [23]。受这项研究启发，我们提出以下问题：（i）能否在单一统一实现中联合建模所有口语语言理解任务并融入上下文，同时不造成明显的性能损失？（ii）联合模型在先前话语中预测的口语语言理解标签，是否有助于更好地建模口语对话上下文？为回答这些问题，我们提出了一种新颖的端到端口语语言理解架构，该架构能够联合建模所有口语语言理解任务，同时有效地从先前话语中学习上下文。该联合模型使用自回归解码器逐一预测对话属性，而模型预测口语语言理解属性的顺序可能会影响模型性能。因此，我们进一步探究（iii）能否采用顺序无关训练，使模型在推理时自动按照最优顺序进行预测。受自动语音识别相关研究的启发 [24],，我们还探究（iv）中间 CTC 损失能否进一步提升口语语言理解性能。我们在近期发布的 HarperValley [25] 银行数据集上进行了大量实验。该数据集包含用户与零售银行客服人员之间的对话。结果表明，我们的联合模型性能与针对每项口语语言理解任务的独立分类器相当；融入对话上下文并采用顺序无关训练，则能进一步显著提升性能。我们的代码和模型已作为 ESPnet-SLU [26] 工具包的一部分公开发布。

T我们工作的主要贡献总结如下 w.

- 我们提出了一种新颖的联合模型，利用对话上下文联合预测意图、对话行为、说话人角色和情绪。
- 我们提出了联合模型的顺序无关训练方法，并展示了其在所有口语语言理解任务上的性能提升。
- 我们研究了利用从先前话语中预测的口语语言理解标签来建模对话上下文的有效性。
- 我们表明，结合中间 CTC 损失的使用可以提升口语语言理解性能。

2. 背景：口语语言理解中的对话上下文

集成对话上下文的口语语言理解（SLU）建模方法扩展了自然语言理解（NLU）系统中经过深入研究的框架 [7–9]。对于 NLU 系统，

[1]我们的代码 和模型也作为 ESPNet-SLU 的一部分公开提供 工具包。

对话上下文序列表示为由 C 个话语组成的序列，即 $S = \{s_c | c = 1, \ldots, C\}$。对话上下文序列中的每个话语表示为 $s_c = \{w_{cn} \in \mathcal{V} | n = 1, \ldots, N_c\}$，其长度为 N_c，词汇表为 $\mathcal{V}$。每个话语都带有一个标记，用于表示各项 NLU 任务。在本研究中，每个话语都带有一个来自标签集 $\mathcal{L}^{\text{da}}$、$\mathcal{L}^{\text{ic}}$、$\mathcal{L}^{\text{sr}}$ 和 $\mathcal{L}^{\text{er}}$ 的标记，分别表示对话行为分类、意图分类、说话人角色预测和情绪识别。这样，每项任务都会生成一个长度相同的标签序列 C，例如 $Y_{\text{对话行为准确率}} = \{y_c^{\text{da}} \in \mathcal{L}^{\text{da}} | c = 1, \ldots, C\}$。根据最大后验理论，NLU 模型旨在输出 $\hat{Y}_{\text{对话行为准确率}}$、$\hat{Y}_{\text{意图分类}}$、$\hat{Y}_{\text{语义槽位识别}}$和 $\hat{Y}_{\text{实体识别}}$，使给定 S 时后验分布 $P(Y_{\text{对话行为准确率}}|S)$、$P(Y_{\text{意图分类}}|S)$、$P(Y_{\text{语义槽位识别}}|S)$ 和 $P(Y_{\text{实体识别}}|S)$ 分别达到最大。

SLU 引入了根据口语话语建模对话上下文的额外复杂性。对话上下文序列由 C 个口语话语组成，即 $X = \{x_c | c = 1, \ldots, C\}$。每个口语话语 $x_c = \{\mathbf{x}_{ct} \in \mathbb{R}^d | t = 1, \ldots, T_c\}$ 都是由长度为 T_c 帧的 d 维语音特征构成的序列。与 NLU 的形式化方法类似，口语理解系统旨在估计标签序列 $\hat{Y}_{\text{da}}$、$\hat{Y}_{\text{ic}}$、$\hat{Y}_{\text{sr}}$ 和 $\hat{Y}_{\text{er}}$，使给定 X时的后验分布$P(Y_{\text{da}}|X)$、$P(Y_{\text{ic}}|X)$、$P(Y_{\text{sr}}|X)$和 $P(Y_{\text{er}}|X)$分别最大化。我们可以按照下面各小节中的描述对这些后验分布进行建模。

2.1. 带对话上下文的独立端到端模型

以往的工作 [17] 分别对每个后验分布进行建模，例如使用转录文本序列 S 建模 $P(Y_{\text{da}}|X)$，并采用维特比近似：

$$P(Y^{\text{da}}|X) = \sum_S P(Y^{\text{da}}|S,X)P(S|X) \tag{1}$$

$$\approx \max_S P(Y^{\text{da}}|S,X)P(S|X) \tag{2}$$

因此，他们假设$y_c^{\text{da}}|s_{1:c-1}$ 与 $x_{1:c-1}$,$y_{1:c-1}^{\text{da}}$ 条件独立，以简化公式 2：

$$P(Y^{\text{da}}|X) \approx \max_S \prod_{c=1}^{C} P(y_c^{\text{da}}|s_{1:c-1}, x_c)P(S|X) \tag{3}$$

转录文本由单独的自动语音识别模块生成，该模块旨在估计使 $P(S|X)$ 最大化的 $\hat{S}$。使用 $\hat{S}$，我们可以修改公式 3：

$$P(Y^{\text{da}}|X) \sim \prod_{c=1}^{C} P(y_c^{\text{da}}|\hat{s}_{1:c-1}, x_c) \tag{4}$$

先前的研究 [17] 在公式 4 中通过传入自动语音识别转录文本 $P(y_c^{\text{da}}|\hat{s}_{1:c-1}, x_c)$ 来构建模型，并将其输入 BERT 等预训练语言模型（LM）$\hat{s}_{1:c-1}$，然后将这些上下文嵌入与从 [27] 获得的声学嵌入拼接起来 x_{co}。他们专注于为每个口语语言理解任务构建单独的模型；在上述描述中，该任务是对话行为分类。因此，要预测所有口语语言理解任务，需要使用所有这些独立模型来估计 $P(Y_{\text{da}}|X)$、$P(Y_{\text{ic}}|X)$、$P(Y_{\text{sr}}|X)$和 $P(Y_{\text{er}}|X)$ 独立地执行，这可能会增加延迟和计算成本，而且没有考虑 SLU 任务之间的依赖关系。

3. 结合对话上下文的联合端到端模型

在这项工作中，我们扩展了第 2.1 节讨论的先前对话整合研究 [17]，并提出对所有口语语言理解任务进行联合建模。我们将包含所有口语语言理解标签的单一目标表示为

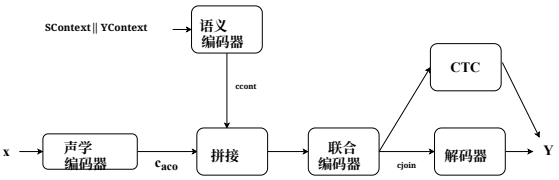

图1：我们提出的联合端到端模型示意图，该模型融入了对话历史，用于联合预测所有口语语言理解任务

$R = (Y_{\text{对话行为准确率}} Y_{\text{意图分类}} Y_{\text{语义槽位识别}} Y_{\text{实体识别}})$，并将公式 4 修改为 $r_c = (y_c^{\text{da}}, y_c^{\text{ic}}, y_c^{\text{sr}}, y_c^{\text{er}})$，如下所示：

$$P(R|X) \sim \prod_{c=1}^{C} P(r_c|r_{1:c-1}, \hat{s}_{1:c-1}, x_c) \tag{5}$$

先前口语话语所预测的口语语言理解标签，即$r_{1:c-1} = (y_{1:c-1}^{\text{da}}, y_{1:c-1}^{\text{ic}}, y_{1:c-1}^{\text{sr}}, y_{1:c-1}^{\text{er}})$，可以纳入模型（公式 5）以更好地建模对话上下文；这与先前工作 [17] 不同，后者假设 $y_c^{\text{da}}|s_{1:c-1}$ 与 $y_{1:c-1}^{\text{da}}$ 条件独立。在 §5.1 中，我们通过实验验证这种先前口语理解标签条件 是否有助益。此外，通过联合建模所有口语语言理解任务，我们期望显著降低延迟并实现轻量级推理。

为实现这一建模方式，我们提出了图 1 所示的联合模型架构。每个话语的输入语音信号（即公式 5 中的 x_c）都会经过声学编码器（$\text{Encoder}_{\text{aco}}$），以生成声学嵌入$\mathbf{c}_{\text{aco}}$。

$$\mathbf{c}_{\text{aco}} = \text{Encoder}_{\text{aco}}(x_c) \tag{6}$$

我们将所有先前口语话语的自动语音识别转录文本 $\hat{s}_{1:c-1}$ 和口语语言理解标签 $r_{1:c-1}$ 拼接起来，并将其输入语义编码器（$\text{Encoder}_{\text{sem}}$），例如预训练语言模型，以编码对话历史：

$$\mathbf{c}_{\text{cont}} = \text{Encoder}_{\text{sem}}(\text{concat}(\hat{s}_{1:c-1}, r_{1:c-1})) \tag{7}$$

语义编码器的输出也会传入一个线性层，以确保上下文嵌入 $\mathbf{c}_{\text{cont}}$ 与声学嵌入$\mathbf{c}_{\text{aco}}$具有相同的隐藏维度。声学嵌入和上下文嵌入拼接在一起（$\text{concat}(\mathbf{c}_{\text{aco}}, \mathbf{c}_{\text{cont}})$），并由联合编码器 $\text{Encoder}_{\text{join}}$ 进行注意力处理，从而生成联合嵌入$\mathbf{c}_{\text{join}}$：

$$\mathbf{c}_{\text{join}} = \text{Encoder}_{\text{join}}(\text{concat}(\mathbf{c}_{\text{aco}}, \mathbf{c}_{\text{cont}})) \tag{8}$$

模型采用联合连接时序分类-注意力训练 [28],，其中连接时序分类目标函数作为辅助任务用于训练注意力模型编码器。由于我们使用自回归解码器逐个预测标签（公式 11），似然$P(r_c|x_c, \hat{s}_{1:c-1}, r_{1:c-1})$取决于目标序列 r_c中口语语言理解标签的顺序。先前的研究将此称为标签歧义（或排列）问题 [29, 30]。受先前关于置换不变训练 [30–32], 的研究启发，我们使用连接时序分类目标函数进行无排列训练，如公式 9 所示；本文将其称为**顺序无关训练**。设$\mathbf{z}$为由联合嵌入$\mathbf{c}_{\text{join}}$计算得到的输出序列变量，则最优排列顺序$\hat{\pi}$ 计算如下：

$$\hat{\pi} = \underset{\pi \in \mathcal{P}}{\text{argmin}} \, \text{Loss}_{\text{CTC}}(\mathbf{z}, r_c^{\pi}), \tag{9}$$

其中，$\mathcal{P}$ 是口语语言理解标签（da、ic、sr、er）的 4! 种可能排列集合，π 是其中一种排列，r_c^{π} 是参考目标，其口语语言理解标签的顺序由 π指示。随后，最优

模型	对话行为准确率 (↑)	意图分类 (↑)	语义槽位识别 (↑)	实体识别 (↑)	RTF (↓) 最慢	RTF (↓) 总计	延迟（毫秒）（↓） 最慢	延迟（毫秒）（↓） 总计	参数（↓）	训练时间（秒）（↓）
无上下文独立端到端模型* ([18])	54.0									
带上下文的独立端到端模型* ([18])	58.3									
无上下文独立端到端模型 §2.1	55.5	47.1	91.9	91.4	1.58	3.28	1153	4515	457.6M	4296
无上下文的联合端到端模型 §3	55.8	47.7	91.5	90.6	1.15	1.58	293	1153	114.4M	1381
有上下文的联合端到端模型 §3	58.1	86.1	95.0	90.9	1.14	1.57	286	1133	152.7M	1960
+ 顺序无关训练（公式 9）（提出的方法）	**58.8**	**86.5**	**95.1**	**91.1**	1.15	1.58	289	1140	152.7M	2040
+ 先前口语理解标签条件（公式 5）（消融实验）	58.3	86.5	94.6	90.8	1.23	1.54	310	1080	165.4M	2613

表1：表明我们的联合模型在使用和不使用对话上下文两种情况下，在对话行为（DA）识别、意图分类（IC）、说话人角色（SR）预测和情感识别（ER）方面的性能。* 虽然先前工作的数据准备设置不同（第4.1节），我们仍展示其DA结果。最佳联合模型结果以**粗体**标示；若优于独立的端到端模型，则进一步**加下划线**。

模型	对话行为准确率 (↑)	意图分类 (↑)	说话人角色 (↑)	实体识别 (↑)
无上下文的联合端到端模型	54.8	47.6	91.1	90.9
带中间 CTC	55.8	47.7	91.5	90.6

表 2：展示中间 CTC 对提升口语语言理解性能影响的结果。

模型	DA (↑)	意图分类 (↑)	语义槽位识别 (↑)	实体识别 (↑)
有上下文的联合端到端模型	58.1	86.1	95.0	90.9
使用预言机上下文的联合端到端模型	58.5	87.0	95.2	91.1

表 3：使用自动语音识别转录文本而非预言机转录文本时的鲁棒性结果。

排列$\hat{\pi}$ 用于计算注意力解码器损失[2]。

对于每一组最优解码顺序和参考目标 ($\hat{\pi}$,r_c)，注意力似然按如下方式计算。

$$\mathbf{h}_l = \text{Decoder}(\mathbf{c}_{\text{JOIN}}, (r_c^{\hat{\pi}})_{1:l-1}) \tag{10}$$

$$P_{\text{Attn}}((r_c^{\hat{\pi}})_l | x_c, \hat{s}_{1:c-1}, r_{1:c-1}, (r_c^{\hat{\pi}})_{1:l-1}) = \text{SoftmaxOut}(\mathbf{h}_l), \tag{11}$$

其中，SoftmaxOut 表示一个线性层后接 softmax 函数，而 $(r_c^{\hat{\pi}})_l$ 是目标序列中的第 l^{th} 项。联合似然 $P_{\text{Attn}}(r_c^{\hat{\pi}}|x_c, \hat{s}_{1:c-1}, r_{1:c-1})$ 随后可计算为各个口语语言理解标签的单独似然的乘积（即从 l 到 [1,4]）。值得注意的是，这种“顺序无关训练”不会增加任何新的模型参数。在推理期间，模型会自动选择标签顺序；也就是说，与训练不同，我们不会强制预测的标签顺序达到最小 CTC 损失。此外，我们的联合模型也可以通过让模型同时生成口语语言理解标签 $r_c^{\hat{\pi}}$ 和自动语音识别转录文本 s_c，使用辅助自动语音识别目标 [26, 33] 进行训练。

4. 实验设置

4.1. 数据集

为展示联合模型的有效性，我们在公开可用的 HarperValleyBank [25]口语对话语料库上进行了实验。该语料库包含银行员工与客户之间的模拟对话，共有 1,446 段对话、23 小时音频以及 25,730 条由人工标注转录文本的话语。这些话语由 59 位不同的说话人说出，并标注了对话行为、对话意图、说话人角色和情绪效价。

在这项工作中，我们训练了一个联合模型，用于执行意图分类、对话行为识别、情感识别和说话人属性预测。对话行为识别是多标签多分类任务，而其他任务均为单标签多分类任务。我们遵循先前工作的设置 [18]，将对话划分为训练集、验证集和测试集[3]。与 [17], 类似，我们也从转录文本中移除了 [噪声],[笑声] 等非词汇标记。训练集包含 1,174 段对话（9.2 小时音频，15,424 条话语），验证集包含 73 段对话（0.6 小时音频，964 条话语），测试集包含 199 段对话（1.6 小时音频，2,903 条话语）。按照先前工作的建议 [17, 25]，我们报告对话行为预测的宏平均F1，以及意图分类、说话人角色预测和情感识别的准确率。我们使用两项指标报告系统延迟：（1）实时因子（RTF），即处理输入音频文件所需的平均时间与输入时长之比；（2）端点延迟，即从话语结束到获得口语语言理解任务预测结果所经过的时间。我们还展示了所有模型每个训练轮次所需的训练时间。

4.2. 架构细节与训练

我们将所提出的方法与最先进的口语语言理解特定任务基线进行比较，这些基线称为“无上下文独立端到端模型”（在公式 4 中优化 $P(y_c^{\text{da}}|x_c)$，而非 $P(y_c^{\text{da}}|\hat{s}_{1:c-1}, x_c)$）。我们还将其与所提出的无上下文模型进行比较，即“无上下文的联合端到端模型”，该模型在公式 5 中优化 $P(r_c|x_c)$，而非 $P(r_c|r_{1:c-1}, x_c, \hat{s}_{1:c-1})$。该联合端到端模型基线不采用公式 9 中定义的顺序无关训练，而是按照固定顺序（$y^{da}, y^{ic}, y^{sr}, y^{er}$）训练口语语言理解标签。为了解所提出的修改的效果，我们首先训练了一个联合模型（称为“有上下文的联合端到端模型”），该模型不使用先前口语话语的口语语言理解标签（即，在公式 5 中优化 $P(r_c|x_c, \hat{s}_{1:c-1})$，而非 $P(r_c|r_{1:c-1}, x_c, \hat{s}_{1:c-1})$），并且也不采用顺序无关训练。在消融实验中，我们还进一步训练了所提出的联合模型，该模型采用公式 9 中定义的“顺序无关训练”，以及另一个联合模型，该模型采用“先前口语理解标签条件”，即先前口语话语的预测标签（公式 5 中的 $r_{1:c-1}$）。

我们的模型使用 PyTorch [34]实现，实验通过 ESPNet-SLU [26, 35] 工具包进行。所有模型均使用辅助自动语音识别目标进行训练。针对具体任务的基线模型由一个 12 层 Conformer [36, 37] 编码器和一个 6 层 Transformer [39, 40] 解码器组成；编码器输入由强大的自监督语音模型 WavLM [38] 提取的特征。我们的“无上下文的联合端到端模型”与针对具体任务的基线模型架构相似。先前的工作 [24] 在编码器网络的一个中间层附加了中间 CTC 损失，以正则化 CTC 训练并提升性能。受这项工作的启发，我们尝试在 Conformer 编码器的第 4、6 层以及

[2]为实现轻量化建模，我们使用 CTC 损失而非解码器损失来计算最优顺序。[3]然而，先前的工作 [17, 18] 还使用现成的自动语音识别模型，将音频与转录文本重新对齐，因此其结果无法与通过我们的设置获得的结果直接比较。

第 8 层添加中间 CTC 损失。“无上下文的联合端到端模型”用于生成对话上下文[4] $\hat{s}_{1:i-1}$。我们使用 Transformers库 [41] 获取 BERT-base-uncased [27] 作为语义编码器（方程 7 中的编码器$_{\text{SEM}}$）。在我们提出的模型中，我们采用带有中间 CTC 损失的 Conformer 架构作为联合编码器（方程 8 中的编码器$_{\text{JOIN}}$）。所有集成对话上下文的模型均采用教师强制，即训练期间使用真实转录文本创建上下文。我们的模型使用单块 NVIDIA Tesla V100-32GB GPU进行训练和推理。推理由 4 个并行作业执行；我们采用所有作业均处理完话语的时间（即最慢作业所用的时间）以及所有作业所用时间的总和来计算延迟指标。所有超参数均根据验证性能选定。

5. 结果

5.1. 主要结果

表1展示了我们的联合模型在使用和不使用对话上下文时的结果。我们的“无上下文独立端到端模型”（第 2.1 节）的表现与先前工作报告的基线结果相近 [18], ，不过由于数据准备方案不同，这些结果并不能直接比较。我们的“无上下文的联合端到端模型”表现与这些特定任务模型相当，同时显著降低了延迟[5]和可训练模型参数的数量，展现了联合建模所有口语语言理解任务的效用。我们按照第 3 节所述的公式，将对话上下文整合进联合模型（表1中的“有上下文的联合端到端模型”），并观察到性能显著提升，尤其是在意图、对话行为和说话者角色识别方面，这证明我们提出的方法能够有效编码上下文。我们还观察到，采用公式 9 中定义的“顺序无关训练”能够进一步提升性能，这验证了我们的假设：训练时口语语言理解标签的排列顺序会影响模型性能。尽管我们的顺序无关训练框架需要计算口语语言理解标签所有可能排列的 CTC 损失（公式 9），因而会增加训练时间，但增幅并不显著。我们还尝试了“先前口语理解标签条件”（即在公式 5 中进行注意力处理$r_{1:c-1}$），仅使用话语级标注标签（即对话行为、情绪和说话人角色）来编码上下文。该模型在意图分类和对话行为预测方面取得了性能提升，同时对其他口语语言理解任务的预测结果也相当。我们计划在未来的工作中进一步研究先前口语话语中的口语语言理解标签的效用。我们的有上下文的联合端到端模型的表现也与“有上下文的独立端到端模型” [18] 相近，且未使用基于文本的系统进行知识蒸馏，尽管这两个模型采用了不同的数据准备方案。

为了更好地理解我们的联合模型，我们还在表 2 中进行了消融研究，并推断出使用中间 CTC 损失可以稳定训练并提升联合模型的性能。

5.2. 使用预言机上下文

为了解自动语音识别转录文本中的错误所产生的影响，我们计算了联合端到端模型在预言机上下文下的结果，即使用真实转录文本作为公式 5 中的$\hat{s}_{1:c-1}$。表3显示，与使用自动语音识别转录文本编码上下文的模型相比，性能仅略有提升，这表明该模型能够抵御 ASR 错误。

[4]自动语音识别转录文本的词错误率（WER）为 11.7。[5]特定任务基线按顺序依次执行。

顺序	训练（%）	推理（%）	训练语句示例
（语义槽位识别、意图分类、对话行为准确率、实体识别）	35.0	40.1	我已经为您订购了补发借记卡
（对话行为准确率、意图分类、语义槽位识别、实体识别）	27.9	36.0	呃，詹姆斯，我叫詹妮弗·琼斯而且我想重置密码
（意图分类、语义槽位识别、对话行为准确率、实体识别）	16.4	6.8	请感谢您的来电，祝您今天愉快
（意图分类、对话行为准确率、语义槽位识别、实体识别）	14.5	7.5	您也一样，再见
（语义槽位识别，对话行为准确率，意图分类，实体识别）	6.2	9.5	您好，这里是哈珀谷国家银行，我的我叫 Patricia，今天有什么可以帮您?

表 4：展示使用公式 9 在训练期间发现的口语语言理解标签最优顺序，以及推理期间预测的口语语言理解标签顺序的结果。我们为每种标签顺序报告一个训练语句示例。

5.3. 口语语言理解标签的预测顺序

我们使用公式 9 分析了训练过程中发现的口语语言理解标签最优顺序（见表4），并观察到，最优排列序列$\hat{\pi}$仅出现在口语语言理解标签所有 4! 种可能排列中的 5 种之内。我们将其与有上下文且采用上下文与顺序无关训练的联合端到端模型所预测的标签顺序进行比较。推理时预测的口语语言理解标签顺序，与训练期间发现的最优顺序呈现相似趋势，其中 (sr, ic, da, er) 和 (da, ic, sr, er) 是最常见的两种顺序。此外，如表4所示，最优顺序为 (sr, da, ic, er) 的训练话语通常出现在对话开头，且主要是问候语。我们推测，对于这些话语而言，这种标签顺序是最优的，因为从这些问候语中识别对话意图颇具挑战，因此模型会在解码对话行为和说话者角色标签之后才预测意图。这样，模型便能利用对这些标签的依赖关系，更好地提取当前话语的意图。对于其他标签顺序，我们也进行了类似分析，并据此推断，我们能够验证训练话语中发现的最优标签顺序。经过更细致的分析，我们观察到，与最优标签顺序为 π 的训练话语相似的测试话语，在推理时也会由联合模型按照相同的标签顺序 π 进行预测。这一发现初步支持了如下假设：通过顺序无关训练，联合模型能够在推理时学会自动按照最优顺序进行预测。基于这一有趣的发现以及表1中观察到的性能提升，我们建议未来关于联合建模不同口语语言理解任务的研究将顺序无关训练纳入其框架。

6. 结论

我们提出了一种新颖的模型架构，能够在口语对话中充分整合对话历史，对意图分类、对话行为预测、说话者角色识别和情感识别进行联合建模。结果表明，与特定任务模型相比，联合模型取得了可比的性能，同时还具备低延迟和轻量级推理的额外优势。我们的联合模型还能够成功捕获对话上下文，从而显著提升所有口语语言理解任务的预测性能。我们通过实验确认，顺序无关训练能够进一步提升性能。在未来的工作中，我们计划探索对话上下文的端到端整合，以及从基于文本的系统中进行知识蒸馏。

7. 致谢

本研究使用了极端科学与工程探索环境（XSEDE） [42], ，该环境由美国国家科学基金会通过资助编号 ACI-1548562 提供支持。具体而言，本研究使用了匹兹堡超级计算中心（PSC）的 Bridges 系统 [43], ，该系统由美国国家科学基金会通过资助编号 ACI-1445606 提供支持。

8. 参考文献

[1] R. De Mori、F. Bechet、D. Hakkani-Tur 等，“口语理解”，IEEE Signal Processing Magazine，第 25 卷第 3 期，第 50–58 页，2008年。[2] J. F. Allen、D. K. Byron、M. Dzikovska 等，“迈向对话式人机交互”，AI magazine，第 22 卷，27，2001年 4，27第。页 – 。[3] B. Agrawal、M. M¨uller、M. Radfar 等，“约束嵌入：用于端到端口语理解的跨模态潜在空间”，arXiv 预印本 arXiv:2011.09044，2020年。[4] S. Cha、W. Hou、H. Jung 等，“与我说话或聊天：支持灵活输入的端到端口语理解”，arXiv，2021年。[5] A. Bhargava、A. Celikyilmaz、D. Hakkani-T¨ur 等，“简易上下文意图预测与槽位检测”，载于 Proc. ICASSP，2013年，第 8337–8341 页。[6] P. Xu 和 R. Sarikaya，“使用循环神经网络进行口语理解系统中的上下文领域分类”，载于 Proc. ICASSP，2014年，第 136–140 页。[7] P. Colombo、E. Chapuis、M. Manica 等，“在序列到序列模型中引导注意力以预测对话行为”，载于 AAAI人工智能大会论文集，第 34 卷，2020年，7594 7601页 – 。[8] C. Bothe、C. Weber、S. Magg 等，“使用简单循环神经网络进行对话行为识别的基于上下文的方法”，arXiv 预印本 arXiv:1805.06280，2018年。[9] V. Raheja 和 J. Tetreault，“结合上下文感知自注意力的对话行为分类”，arXiv 预印本 arXiv:1904.02594，2019年。[10] S. Kim、S. Dalmia 和 F. Metze，“用于对话上下文融合的端到端语音识别门控嵌入”，载于 Proc. ACL，2019年，第 1131–1141 页。[11] ——，“面向双人对话的交叉注意力端到端自动语音识别”，载于 Proc. Interspeech，2019年，第 4380–4384 页。[12] T. Hori、N. Moritz、C. Hori 等，“基于Transformer的长上下文端到端语音识别”，载于 Proc. Interspeech，2020年，第 5011–5015 页。

[13] Y. -N. Chen, D. Hakkani-T¨ur, G. T¨ur, 等人.，“结束-to-端记忆网络-通过知识延续处理多轮口语理解的工作。」，发表于*Interspeech* 论文集，2016 年，第 3245–3249 页。

[14] C. Sankar、S. Subramanian、C. Pal、等，“神经对话系统是否有效利用对话历史？一项实证研究”，*arXiv* 预印本 *arXiv:1906.01603*，2019年。[15] A. Bapna、G. Tur、D. Hakkani-Tur、等，“用于口语理解的序列对话上下文建模”，*arXiv* 预印本 *arXiv:1705.03455*，2017年。[16] V. Vukoti´c、C. Raymond 和 G. Gravier，“借助具备对话感知能力的双向 GRU 网络，在口语理解中迈出超越局部观测的一步”，载于*Interspeech* 论文集，2016年，第 3241–3244页。[17] J. Ganhotra、S. Thomas、H.-K. J. Kuo、等，“将对话历史整合到端到端口语理解系统中”，*arXiv* 预印本 *arXiv:2108.08405*，2021年。[18] V. Sunder、S. Thomas、H.-K. J. Kuo、等，“面向端到端整合对话历史，以提升口语理解能力”，载于 *ICASSP* 论文集，2022年，第7497–7501页。[19] N. Tomashenko、C. Raymond、A. Caubri`ere、等，“将对话历史整合到端到端信号到概念口语理解系统中”，载于*ICASSP* 论文集，2020年，第8509–8513页。[20] S. Arora、S. Dalmia、X. Chang、等，“两遍式低延迟端到端口语理解”，载于*Interspeech* 论文集，2022年，第3478–3482页。[21] T. O’Malley、A. Narayanan、Q. Wang、等，“用于联合声学回声消除、语音增强和语音分离的基于 Conformer 的自动语音识别前端”，载于*ASRU* 论文集，2021年，第304–311页。

[22] W. Zhang、C. Boeddeker、S. Watanabe、等，“采用改进数值稳定性和先进前端的端到端去混响、波束形成与语音识别”，载于 *ICASSP* 论文集，2021年，第6898–6902页。[23] S.-Y. Chang、R. Prabhavalkar、Y. He、等，“采用端到端模型联合进行端点检测与解码”，载于*ICASSP* 论文集，2019年，第5626–5630页。[24] J. Lee 和 S. Watanabe，“基于连接时序分类的语音识别中间损失正则化”，载于 *ICASSP* 论文集，2021年，第6224–6228页。[25] M. Wu、J. Nafziger、A. Scodary、等，“Harpervalleybank：一个特定领域的口语对话语料库”，*arXiv* 预印本 *arXiv:2010 13929*，2020年。。[26] S. Arora、S. Dalmia、P. Denisov、等，“ESPnet-SLU：通过 ESPnet 推进口语理解”，载于*ICASSP* 论文集，2022年，第7167–7171页。[27] J. Devlin、M. Chang、K. Lee、等，“BERT：用于语言理解的深度双向 Transformers 预训练”，载于*NAACL-HLT* 论文集，2019年，第4171–4186页。[28] S. Kim、T. Hori 和 S. Watanabe，“基于多任务学习的联合 CTC-注意力端到端语音识别”，载于*ICASSP* 论文集，2017年，第4835–4839页。[29] C. Weng、D. Yu、M. L. Seltzer、等，“用于单通道多说话人语音识别的深度神经网络”，*IEEE/ACM* 音频、语音与语言处理汇刊，第23卷，第10期，第1670–1679页，2015年。[30] J. R. Hershey、Z. Chen、J. Le Roux、等，“深度聚类：用于分割与分离的判别式嵌入”，载于*ICASSP* 论文集，2016年，第31–35页。

[31] D. Yu, M. Kolbæk, Z.-H. Tan, 等人，“面向说话人无关的多说话人语音分离的深度模型置换不变训练”，载于*ICASSP* 论文集，2017年，第 241–245 页。

[32] X. Chang、Y. Qian、K. Yu、等，《无需预训练的端到端单声道多说话人自动语音识别系统》，载于*ICASSP* 论文集，2019年，第 6256–6260 页。[33] A. Deoras、R. Sarikaya、G. Tur、等，《语音识别与语义标注的联合解码》，载于*Interspeech* 论文集，2012年，第 1067–1070 页。[34] A. Paszke、S. Gross、F. Massa、等，《PyTorch：一种命令式风格的高性能深度学习库》，*NeurIPS* 论文集，第 32 卷，第 8024–8035 页，2019年。[35] S. Watanabe、T. Hori、S. Karita、等，《ESPnet：端到端语音处理工具包》，载于*Interspeech* 论文集，2018年，第 2207–2211 页。[36] A. Gulati、J. Qin、C. Chiu、等，《Conformer：用于语音识别的卷积增强型 Transformer》，载于 *Interspeech* 论文集，2020年，第 5036–5040 页。[37] P. Guo、F. Boyer、X. Chang、等，《Conformer 助力 ESPnet 工具包的最新进展》，载于*ICASSP* 论文集，2021年，第 5874–5878 页。[38] S. Chen、C. Wang、Z. Chen、等，《WavLM：面向全栈语音处理的大规模自监督预训练》，*arXiv* 预印本 *arXiv:2110.13900*，2021年。[39] A. Vaswani、N. Shazeer、N. Parmar、等，《Attention is All You Need》，*NeurIPS* 论文集，第 30 卷，第 5998–6008 页，2017年。[40] S. Karita、N. Chen、T. Hayashi、等，《语音应用中 Transformer 与循环神经网络的比较研究》，载于*ASRU* 论文集，2019年，第 449–456 页。[41] T. Wolf、L. Debut、V. Sanh、等，《Transformers：自然语言处理领域的最新进展》，载于*EMNLP* 论文集，2020年，第 38–45 页。[42] J. Towns、T. Cockerill、M. Dahan、等，《XSEDE：加速科学发现》，*Computing in Science & Engineering*，第 16 卷，第 5 期，第 62–74 页，2014年。

[43] N. A. Nystrom、M. J. Levine、R. Z. Roskies，等，“Bridges：面向新兴社群和数据分析的独特灵活的高性能计算资源”，载于《*XSEDE*会议：增强型网络基础设施促进的科学进展》，2015年。