arXiv:2302.08583v1 [eess.AS] 16 Feb 2023

JEIT：端到端模型与内部语言模型联合训练用于语音识别

*Zhong Meng**、*Weiran Wang**、*Rohit Prabhavalkar*、*Tara N. Sainath*、*Tongzhou Chen*、*Ehsan Variani*、*Yu Zhang*、*Bo Li*、*Andrew Rosenberg*、*Bhuvana Ramabhadran*

谷歌有限责任公司，美国

摘要

我们提出 JEIT，这是一种端到端（E2E）模型与内部语言模型（ILM）联合训练方法，可在 E2E 训练期间将大规模未配对文本注入 ILM，从而提升罕见词语音识别性能。通过 JEIT，E2E 模型在音频-转录文本对上计算端到端损失，而其 ILM 则在未配对文本上估计交叉熵损失。训练 E2E 模型的目标是最小化 E2E 损失与 ILM 损失的加权和。在 JEIT 训练期间，ILM 从未配对文本中吸收知识，而 E2E 训练则发挥正则化作用。与 ILM 适配方法不同，JEIT 无需单独的适配步骤，也避免了对 ILM 进行库尔贝克-莱布勒散度正则化。我们还表明，在 JEIT 框架下，模块化混合自回归转导器（MHAT）的表现优于 HAT，且在 ILM 适配期间比 HAT 稳健得多。为进一步提升未配对文本注入的效果，我们还提出 JEIT 与 JOIST 联合训练（CJJT），该方法受益于模态匹配、编码器文本注入和 ILM 训练。JEIT 和 CJJT 均可促进更有效的语言模型融合。使用 1000 亿条未配对句子时，与未使用未配对文本训练的模型相比，JEIT/CJJT 可使罕见词识别准确率最高提升 16.4%。

索引词— 语音识别、文本注入、内部语言模型

1. 引言

端到端（E2E）模型通过直接将语音信号映射到词序列，在自动语音识别（ASR）方面取得了优异的性能 [1, 2, 3, 4, 5] 。然而，即使使用大量音频-转录文本对进行训练，端到端模型在评估包含训练数据中罕见词语（罕见词）的语句时，表现仍然不佳 [6]。此外，通过人工转录获得这类监督语音的成本高昂。为解决这一问题，在训练或推理期间利用大规模未配对文本中的知识是一种很有前景的方案，因为未配对文本的数量比音频-转录文本对多出几个数量级，且涵盖的词汇量也大得多。

语言模型（LM）融合是利用未配对文本提升端到端自动语音识别性能的常用方法。首先使用未配对文本训练外部语言模型。在浅层融合中 [7, 8], , 束搜索的每一步都会计算端到端模型得分与语言模型得分的对数线性插值。为改进浅层融合，研究者提出了基于内部语言模型估计的融合 [9, 10, 11, 12, 13, 14] , 通过估计内部语言模型（ILM）得分并将其从浅层融合得分中减去来实现。然而，所有这些方法在推理期间都需要外部语言模型，从而增加了解码时间和计算成本。

为解决这一问题，许多研究尝试将未配对文本纳入端到端模型的训练阶段。一种直观的解决方案是根据未配对文本合成语音，并用它来训练端到端模型 [15, 16, 17, 18]。然而，训练文本转语音（TTS）模型和合成语音的计算成本都很高。为避免这一问题，研究者提出了模态匹配方法[19, 20, 21, 22, 23] , 将未配对文本映射到语音与文本共享的潜在空间，再利用潜在嵌入训练端到端模型。

另一种方法是，将端到端模型的解码器（对于转换器模型，还包括联合网络）中的编码器输出置零 [10, 11, 24], 此时它的行为类似于内部语言模型。为了实现快速的纯文本自适应，可将未配对文本注入训练充分的端到端模型的解码器中[25, 26, 27, 28, 29]。这些方法会在一或两阶段的端到端训练后，额外执行一个适配步骤：使用纯文本数据对端到端模型的内部语言模型进行微调，以最小化交叉熵损失。此外，还会进行 Kullback-Leibler 散度（KLD） [30] 正则化，以维持源域自动语音识别性能。

本工作的创新贡献如下：(1) 我们提出端到端模型与内部语言模型联合训练（JEIT），将解码器文本注入整合到单阶段端到端训练中，从而简化了这一过程。JEIT 的表现优于未采用 KLD 正则化的纯文本自适应。(2) 我们进一步提出结合 JEIT 与 JOIST 的训练方法（CJJT）[23], 并证明通过内部语言模型进行的解码器文本注入与通过 JOIST 进行的编码器文本注入互为补充，二者带来的提升可以叠加。(3) 我们表明，所有文本注入方法都能促进更有效的语言模型融合。(4) 我们在 Google 的大规模流式生产任务中验证了这些方法：在不影响语音搜索性能的情况下，JEIT 和 CJJT 分别使罕见词测试集上的词错误率相对降低最高达 10.2% 和 16.4%。

2. 相关工作

2.1. 混合自回归转换器（HAT）

端到端模型根据输入语音特征序列 $\mathbf{X} = \{\mathbf{x}_1 \dots U \quad _t \in {}^{d_x} \ t = 1 \dots T \ \mathcal{V} \quad u = \quad , \dots, \mathbf{x}_T\}$，估计输出标签序列 $\mathbf{Y} = \{y_1, \dots, y_U\}$ 的后验分布 $P(\mathbf{Y}|\mathbf{X}; \theta_{\text{E2E}})$，其中 $y_u \in \mathcal{V}$,,,, 且 $\mathbf{x}\mathbb{R}$,,,。是所有可能标签的集合，例如子词单元等。 y_0 是句首标记。

HAT [10]由声学编码器、标签解码器和联合网络组成。在图 1 中，编码器将输入语音特征$\mathbf{X}$ 转换为声学嵌入向量 $\mathbf{F} = \{\mathbf{f}_1, \dots, \mathbf{f}_T\}, \mathbf{f}_t \in \mathbb{R}^{d_f}$，即 $\mathbf{F} = \text{Encoder}(\mathbf{X})$。标签解码器接收先前的标签，以生成当前标签嵌入 $\mathbf{g}_u^{\text{L}} \in \mathbb{R}^{d_{\text{L}g}}$，即 $\mathbf{g}_u^{\text{L}} = \text{LabelDecoder}(\mathbf{Y}_{0:u-1})$。联合网络结合声学嵌入和标签嵌入，计算空白分布

$$b_{t,u} = \text{Sigmoid}[\mathbf{w}^\intercal \phi(\mathbf{W}_1 \mathbf{f}_t + \mathbf{W}_2 \mathbf{g}_u^{\text{L}})], \tag{1}$$

其中 $\mathbf{w} \in \mathbb{R}^{d_h}$ 是一个向量，$\mathbf{W}_1 \in \mathbb{R}^{d_h \times d_f}$ 和 $\mathbf{W}_2 \in \mathbb{R}^{d_h \times d_{\text{L}g}}$是投影矩阵。 $\phi(\cdot)$ 是一个非线性函数。标签

*共同贡献。

后验概率 根据先前的语音特征和标签计算得出 为

$$P(y_u|\mathbf{X}_{1:t}, \mathbf{Y}_{0:u-1}) = \text{Softmax}[\mathbf{W}\phi(\mathbf{W}_1\mathbf{f}_t + \mathbf{W}_2\mathbf{g}_u^{\text{L}})], \quad (2)$$

其中 $\mathbf{W} \in \mathbb{R}^{|\mathcal{V}|\times d_h}$ 是一个投影矩阵。因此，给定对齐历史的标签后验概率为 $(1-b_{t,u})P(y_u|\mathbf{X}_{1:t}, \mathbf{Y}_{0:u-1})$。

2.2. 模块化 HAT（MHAT）

为实现更稳健的纯文本自适应，我们在 [28] 中提出了 MHAT，通过结构设计将内部语言模型分数预测与声学模型分数或空白分数预测分离。如图 2 所示，MHAT 引入了一个空白解码器，该解码器与标签解码器接收相同的先前标签，以生成当前标签嵌入，如下所示

$$\mathbf{g}_u^{\text{B}} = \text{BlankDecoder}(\mathbf{Y}_{0:u-1}), \quad (3)$$

其中 $\mathbf{g}_u^{\text{B}} \in \mathbb{R}^{d_{\text{B}g}}$ 是标签解码器的标签嵌入。空白后验概率 $b_{t,u}$ 通过将 $\mathbf{g}_u^{\text{B}}$ 代入公式 (1) 得到。$\mathbf{f}_t$ 和 $\mathbf{g}_u^{\text{L}}$ 经过投影并归一化后，分别成为 $|\mathcal{V}|$ 维（dim）的声学模型对数概率向量 $\mathbf{a}_t$ 和 ILM 对数概率向量 $\mathbf{l}_u$

$$\mathbf{a}_t = \text{LogSoftmax}(\mathbf{W}_3\mathbf{f}_t), \quad \mathbf{l}_u = \text{LogSoftmax}(\mathbf{W}_4\mathbf{g}_u^{\text{L}}), \quad (4)$$

其中 $\mathbf{W}_3 \in \mathbb{R}^{|\mathcal{V}|\times d_f}$ 和 $\mathbf{W}_4 \in \mathbb{R}^{|\mathcal{V}|\times d_{\text{L}g}}$ 是投影矩阵。$\mathbf{a}_t$ 和 $\mathbf{l}_u$ 相加后进行归一化，以计算标签后验概率

$$P(y_u|\mathbf{X}_{1:t}, \mathbf{Y}_{0:u-1}) = \text{Softmax}\left(\mathbf{a}_t + \mathbf{l}_u\right). \quad (5)$$

2.3. ILM 训练（ILMT）与 ILM 适应（ILMA）

ILMT [10, 24] 在端到端模型训练期间会最小化额外的内部语言模型损失。端到端损失基于音频-转录文本对计算，而内部语言模型损失仅根据训练转录文本得出。ILMT 旨在促使内部语言模型也能像独立的神经语言模型一样运行，从而（1）估算准确的内部语言模型分数，以改进基于 ILME 的融合； [24]（2）使内部语言模型能够进一步适应纯文本数据 [26]。ILMT 不使用未配对文本，也不会提升源域或罕见词测试集上的自动语音识别性能 [24]。与 ILMT 不同，JEIT 在端到端训练期间将未配对文本注入内部语言模型，以提升罕见词识别。

ILMA [26] 可快速对端到端模型进行纯文本自适应，从而提升罕见词的自动语音识别性能。在 ILMA 中，我们首先对端到端模型进行 ILMT，然后使用未配对文本微调内部语言模型，以最小化交叉熵 ILM 损失。为防止源域自动语音识别性能下降，我们在 ILMA 过程中最小化未自适应与自适应内部语言模型输出分布之间的额外 KLD。为简化 ILMA，JEIT 将 ILMT 和 ILMA 两个阶段合并为一个训练阶段，无需进行 KLD 正则化。

2.4. 语音与文本联合建模（JOIST）

JOIST [23] 将未配对文本融入端到端训练，显著提升了罕见词识别能力。它通过编码器注入未配对文本，使文本数据能够惠及整个端到端模型。在 JOIST 中，未配对文本首先被切分为子词或音素序列，然后通过将每个词元复制固定或随机次数进行上采样。上采样后的文本经过掩码处理，再输入文本编码器以生成词元嵌入，随后进一步传递至解码器输入或编码器的一层。JOIST 最小化由音频-转录文本对 $\mathcal{D}_{\text{P}}$ 和未配对文本 $\mathcal{D}_{\text{UP}}$分别得出的两个端到端损失的加权和

$$\mathcal{L}_{\text{JOIST}}(\mathcal{D}_{\text{P}}, \mathcal{D}_{\text{UP}}) = \mathcal{L}_{\text{E2E}}(\mathcal{D}_{\text{P}}; \theta_{\text{E2E}}) + \alpha\mathcal{L}_{\text{E2E}}(\mathcal{D}_{\text{UP}}; \theta_{\text{E2E}}), \quad (6)$$

其中，两个端到端损失定义如下：

$$\mathcal{L}_{\text{E2E}}(\mathcal{D}_{\text{P}}; \theta_{\text{E2E}}) = -\sum_{(\mathbf{X},\mathbf{Y})\in\mathcal{D}_{\text{P}}} \log P(\mathbf{Y}|\mathbf{X}; \theta_{\text{E2E}}), \quad (7)$$

$$\mathcal{L}_{\text{E2E}}(\mathcal{D}_{\text{UP}}; \theta_{\text{E2E}}) = -\sum_{\mathbf{Y}\in\mathcal{D}_{\text{UP}}} \log P\left(\mathbf{Y}|F(\mathbf{Y}); \theta_{\text{E2E}}\right), \quad (8)$$

其中， $F(\cdot)$ 是一个函数，用于对 $\mathcal{D}_{\text{UP}}$ 中未配对的句子进行词元化、下采样和掩码处理。 $\alpha > 0$ 是未配对端到端损失的权重。在本研究中，我们将内部语言模型损失纳入 JOIST，以进一步提升自动语音识别性能。

3. 端到端与内部语言模型联合训练（JEIT）

将编码器输出清零后，可以根据端到端模型的输出估计内部语言模型概率 [10, 11]。HAT 的内部语言模型由解码器和联合网络构成；MHAT 的内部语言模型由标签解码器和输出投影 $\mathbf{W}_4$ 构成；注意力编码器-解码器模型的内部语言模型则是其解码器。

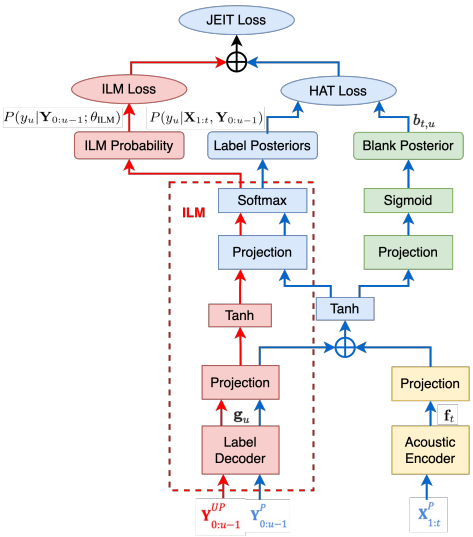


图1。HAT 的 JEIT。蓝色和红色箭头分别表示音频-转录文本对（$\mathbf{X}^{\text{P}}_{1:t}$， $\mathbf{Y}^{\text{P}}_{0:u-1}$）和未配对文本$\mathbf{Y}^{\text{UP}}_{0:u-1}$的前向传播路径。

我们的目标是利用大规模未配对文本 提高罕见词测试集上的自动语音识别准确率，同时保持源域任务（例如语音搜索）上的词错误率。在这项工作中，我们提出 JEIT，即端到端模型与内部语言模型的联合训练方法，在端到端训练期间将未配对文本注入内部语言模型。如图1和图2所示，内部语言模型通过未配对文本 训练，以最小化内部语言模型损失；与此同时，整个端到端模型则使用音频-转录文本对训练，以最小化端到端损失。最小化内部语言模型损失，使内部语言模型成为目标域中强大的神经语言模型；而端到端损失则起到正则化作用，确保内部语言模型能够与端到端模型的其他组件协同工作，从而预测准确的端到端分数。具体而言，JEIT 最小化以下端到端损失与内部语言模型损失的加权和

$$\mathcal{L}_{\text{JEIT}}(\mathcal{D}_{\text{P}}, \mathcal{D}_{\text{UP}}) = \mathcal{L}_{\text{E2E}}(\mathcal{D}_{\text{P}}; \theta_{\text{E2E}}) + \beta\mathcal{L}_{\text{ILM}}(\mathcal{D}_{\text{UP}}; \theta_{\text{ILM}}), \quad (9)$$

其中 $\beta > 0$ 是内部语言模型损失的权重。内部语言模型损失是在未配对文本 $\mathcal{D}_{\text{UP}}$ 上由内部语言模型预测的所有标签序列的负对数概率之和，具体如下

$$\mathcal{L}_{\text{ILM}}(\mathcal{D}_{\text{UP}}; \theta_{\text{ILM}}) = -\sum_{\mathbf{Y}\in\mathcal{D}_{\text{UP}}} \sum_{u=1}^{U} \log P(y_u|\mathbf{Y}_{0:u-1}; \theta_{\text{ILM}}), \quad (10)$$

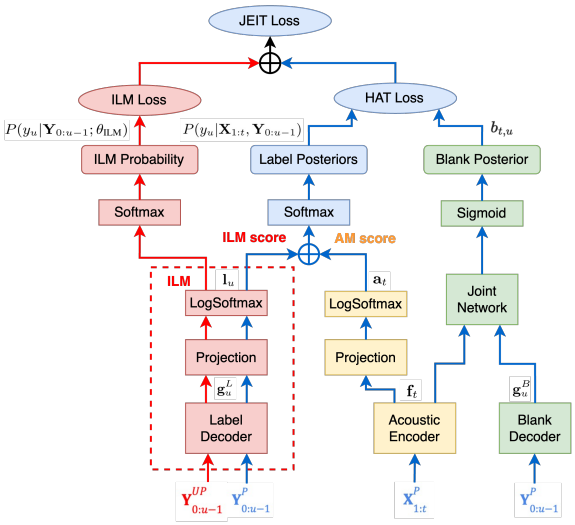


图 2。MHAT 的 JEIT。蓝色和红色箭头分别表示音频-转录文本对（$\mathbf{X}^{\text{P}}_{1:t}$, $\mathbf{Y}^{\text{P}}_{0:u-1}$）和未配对文本$\mathbf{Y}^{\text{UP}}_{0:u-1}$的前向传播路径。

其中 $\theta_{\text{ILM}} \subseteq \theta_{\text{E2E}}$ 表示 ILM 参数。

与纯文本自适应 [25, 26, 27, 28], 相比，JEIT 大幅简化了整个学习过程：1）JEIT 将音频-转录文本训练和未配对文本自适应这两个步骤合并为一个联合训练步骤，从而降低了计算成本以及训练/自适应所需的时间。2）JEIT 无需对 ILM 输出分布进行 KLD 正则化。

为了提升罕见词识别能力，JEIT 将未配对文本注入端到端模型的标签解码器，以最小化 $\mathcal{L}_{\text{E2E}}(\mathcal{D}_{\text{P}};\theta_{\text{E2E}})$ 和 $\mathcal{L}_{\text{ILM}}(\mathcal{D}_{\text{UP}};\theta_{\text{ILM}})$；而 JOIST 则通过编码器注入未配对文本，以最小化 $\mathcal{L}_{\text{E2E}}(\mathcal{D}_{\text{P}};\theta_{\text{E2E}})$ 和 $\mathcal{L}_{\text{E2E}}(\mathcal{D}_{\text{UP}};\theta_{\text{E2E}})$。为了兼顾这两种方法的优势，我们提出了 JEIT 与 JOIST 联合训练（CJJT），通过最小化由音频-转录文本对导出的端到端损失、由未配对文本导出的端到端损失以及由未配对文本导出的内部语言模型损失的加权和，如下所示

$$\mathcal{L}_{\text{CJJT}}(\mathcal{D}_{\text{P}},\mathcal{D}_{\text{UP}}) = \mathcal{L}_{\text{E2E}}(\mathcal{D}_{\text{P}};\theta_{\text{E2E}}) + \alpha\mathcal{L}_{\text{E2E}}(\mathcal{D}_{\text{UP}};\theta_{\text{E2E}}) + \beta\mathcal{L}_{\text{ILM}}(\mathcal{D}_{\text{UP}};\theta_{\text{ILM}}). \quad (11)$$

我们在实验中展示，JEIT 和 JOIST 彼此互补，而 CJJT 的自动语音识别性能优于单独使用其中任一种方法。

在推理阶段，我们可以在采用 JEIT 或 CJJT 后，将外部语言模型集成到端到端模型中，以进一步提升罕见词的自动语音识别效果。我们展示了语言模型融合与 JEIT 和 CJJT 均具有互补性，即使外部语言模型使用与 JEIT 中相同的未配对文本 $\mathcal{D}_{\text{UP}}$ 进行训练。

4. 实验

4.1. 数据集

我们使用 ~650M 个多领域英语音频-转录文本对作为有监督训练数据 [23]。这些数据涵盖多个领域，包括语音搜索、听写、YouTube、电话通信等。YouTube 转录文本以半监督方式生成 [31]，而其他数据均经过匿名化并由人工转录 [32]。此外，我们采用多条件训练 [33], 、随机 8kHz 降采样 [34]和 SpecAug [35] 来扩充数据并提升其多样性。

训练或适应过程中使用的未配对文本包含来自地图、Google Play、网页和 YouTube 等领域的 1000 亿条匿名化句子，其规模比音频-转录文本对大两个数量级以上。外部语言模型使用 50% 的配对数据转录文本和 50% 的未配对文本进行训练，以确保基线语音搜索任务的质量不会下降。

我们在语音搜索（VS）测试集上评估模型，该测试集包含 ~12 K 条匿名化且经过人工转录的语音搜索话语，平均时长为 5.5 秒。为评估自动语音识别性能在长尾词上的表现，我们针对 5 个域分别构建了罕见词测试集：地图、Google Play、网页和 YouTube（YT）域。所有测试集均包含在训练集中出现次数少于 5 次、并由文本转语音系统 [36] 合成的罕见专有名词。我们的目标是在不降低语音搜索词错误率的情况下，提高自动语音识别在 4 个罕见词测试集上的准确率。

4.2. 建模

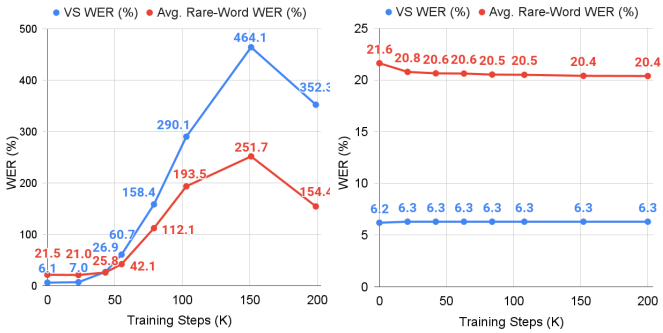
我们按照 [37, 38] 中的方法，使用两遍级联编码器和独立解码器训练 HAT 和 MHAT。它们共享相同的前端和编码器架构。具体而言，我们从语音信号中提取 128 维对数梅尔滤波器组，并对其进行下采样，以每 30 毫秒生成一个 512 维特征。每个语音特征均附加一个 16 维域标识 [39]。因果编码器由一个采用因果卷积和左上下文注意力的 7 层 Conformer 构成。非因果编码器由一个采用右上下文注意力的 10 层 Conformer 构成，可处理未来 900 毫秒的语音。每个 Conformer层均采用 512 维、8 头的自注意力，以及大小为 15 的卷积核。

HAT 或 MHAT 的因果解码器和非因果解码器分别使用因果编码器和非因果编码器的输出进行解码。HAT 和 MHAT 的标签解码器均为两层长短期记忆网络，每层包含 2048 个隐藏单元。在 HAT 中，标签解码器的输出先经过一个 640 维的前馈联合网络，再投影到由 4096 个输出单元构成、用于表示子词单元的输出层 [40]。在 MHAT 中，标签解码器的输出直接投影到大小相同的输出层。HAT 和 MHAT 的内部语言模型分别包含 3070 万和 3000 万个参数。MHAT 的空白解码器是一个 320 维的 V^2 嵌入解码器 [41, 42]，其查找表由最后两个词元共享，且包含 150 万个参数。总体而言，HAT 和 MHAT 分别包含 2.05 亿和 2.1 亿个模型参数。本文仅报告第二遍词错误率。我们使用仅包含音频-转录文本对的数据训练基线系统，并在表1中展示其词错误率。

此外，我们训练了一个 12 层的 Conformer 语言模型，其自注意力维度为 384，前馈层维度为 3072 [6]。该外部语言模型的左侧注意力上下文为 31，总计包含 7000 万个参数。

4.3. HAT 和 MHAT 的 ILMA

我们首先训练一个 ILMT [24] 模型，ILM损失权重设为 0.1，并将其用作 ILMA [26]的初始模型。对于 ILMA 和 JEIT，我们分别为配对音频-转录文本数据和未配对文本采用 4,096 和 32,768 的小批量大小。在 ILMA 期间，HAT 和 MHAT 均采用权重为 0.5 的 KLD 正则化。在图3中，我们绘制了 HAT ILMA 和 MHAT ILMA 的词错误率随训练步数的变化情况。HAT ILMA 的词错误率在训练步数达到 5K、取得最佳值后急剧上升，而 MHAT 的词错误率则逐渐下降，直至训练步数超过 200K。这是因为缺少结构分解时，HAT 无法适应不断增强的内部语言模型，因此会失去执行端到端自动语音识别的功能。这表明，MHAT 在 ILMA 中优于 HAT，因为其结构上独立的内部语言模型使 MHAT 能够随着内部语言模型不断增强而持续提升自动语音识别能力。我们在表1中列出了 ILMA 的最佳词错误率。



(a) HAT ILMA (b) MHAT ILMA

图3。采用 LSTM 标签解码器的 HAT 和 MHAT 在不同训练步数（K）下的内部语言模型适应（ILMA）词错误率（%）。

4.4. HAT 和 MHAT 的 JEIT

我们采用与 ILMA 相同的小批量大小进行 JEIT，并分别为 HAT 和 MHAT 使用 β 0.2 和 4.0 的 ILM损失权重。MHAT 的最优 ILM损失权重显著更大，这体现了因子分解带来的优势：MHAT 能够适应不断增强的内部语言模型，从而提升自动语音识别性能，而 HAT 则不能。在表1中，MHAT JEIT 在所有方法中表现最佳，在罕见词测试集上相较于基线HAT 的词错误率相对降低了 4.8%–10.2%。对于 MHAT，JEIT 在所有测试集上的词错误率均优于 ILMA。MHAT JEIT 的表现始终优于 HAT JEIT，词错误率相对最多降低 3.5%。随着 JEIT 的进行，HAT 和 MHAT 的词错误率均持续下降，没有出现突然上升。这表明，JEIT 中的端到端损失比 ILMA 中的 KLD 起到了更好的正则化作用。总的来说，我们首次证明，联合训练内部语言模型优于适应。

模型	Exp	VS	罕见词			
			Maps	Play	Web	YT
HAT	Base	**6.1**	14.0	37.3	21.6	24.6
	ILMT	6.1	14.1	37.6	21.7	24.8
	ILMA	6.2	13.6	36.3	20.8	24.2
	JEIT	6.3	13.3	36.8	20.1	23.2
MHAT	Base	6.2	14.1	37.4	21.4	24.6
	ILMT	6.2	14.3	37.8	21.8	25.0
	ILMA	6.3	13.3	35.5	20.3	23.5
	JEIT	6.2	**13.2**	**35.5**	**19.4**	**22.6**

表1。采用不同训练或适配方法的 HAT、MHAT 与 LSTM标签解码器的词错误率（%）。基线和 ILMT [24]模型使用 6.5 亿个多领域（MD）音频-转录文本对进行训练。ILMA [26] 和所提出的 JEIT 使用相同的配对数据以及 1000 亿个多领域未配对句子。

4.5. 使用不同解码器的 JEIT

在第 4.2 节中，我们保持级联编码器不变，同时改变 MHAT 标签解码器的类型和规模。除长短期记忆网络外，我们还探索了更简单、更小型的标签解码器：V^2 嵌入和 V^4嵌入 [41]，它们的嵌入维度为 640，分别以最近 2 个和 4 个词元作为条件。每个先前词元都有独立的查找表。使用第 4.2 节中的相同空白解码器。采用 V^2 和 V^4 嵌入解码器的 MHAT，其内部语言模型分别包含 860 万和 1460 万个参数，总参数量分别为 1.69 亿和 1.82 亿。在表 2 和表 1 中，采用 V^2 嵌入、V^4 嵌入和长短期记忆网络解码器的 MHAT，其 JEIT 相较于基线 MHAT 在 Maps、Play、Web 和 YT 上分别实现了 1.6%–4.1%、1.6%–4.9% 和 4.3%–8.8% 的相对词错误率下降。对于这 3 种解码器，JEIT 在罕见词测试集上均未导致词错误率上升。这表明 JEIT 对各种类型和规模的标签解码器都有益。随着内部语言模型规模增大，以及标签解码器的条件历史延长，JEIT 的效果也会增强。

标签 Dec	参数 (M)	实验对比	罕见词			
			地图	播放	Web	YT
V^2 嵌入	8.6	基线 6.2	14.5	37.9	21.9	24.9
		JEIT 6.3	13.9	**36.6**	21.3	24.5
V^4 嵌入	14.6	基线 6.3	14.4	37.5	22.1	25.0
		JEIT **6.2**	**13.7**	36.9	**21.1**	**24.3**

表2。采用 V^2 嵌入和 V^4 嵌入解码器的 MHAT 的词错误率（%）。两个 MHAT 的级联编码器和空白解码器与表1中采用 LSTM 标签解码器的 MHAT 相同

4.6. 将 JEIT 与其他文本注入方法结合

我们按照 [23]中的设置，使用基于音素的非配对文本训练 JOIST MHAT。文本编码器的输出馈入因果编码器的第 3 个 Conformer 层。非配对端到端损失的权重 α 为 0.25。我们采用 1.5 的内部语言模型损失权重 β 进行 JEIT 与 JOIST 联合训练（CJJT）。在语言模型融合过程中，我们减去内部语言模型分数。

Exp	参数 (M)	VS	罕见词			
			Maps	Play	Web	YT
Base	210	6.2	14.1	37.4	21.4	24.6
JEIT		6.2	13.2	35.5	19.4	22.6
JOIST		6.2	12.4	34.7	18.5	22.1
JEIT +JOIST（CJJT）		6.2	12.0	33.8	17.9	21.3
基线 +语言模型	280	6.0	11.8	34.4	17.9	23.2
JEIT +语言模型		6.0	11.7	34.0	17.0	22.6
CJJT +语言模型		**6.0**	10.6	31.9	15.6	20.8
CJJT +MWER + 语言模型		6.1	**10.1**	**30.7**	**14.7**	**19.3**

表 3。结合 JEIT 与各种文本注入方法及 MWER 时，采用 LSTM 标签解码器的 MHAT 的词错误率（%）。

CJJT 始终优于 JOIST 和 JEIT，这表明向编码器和解码器注入文本具有互补性，且其收益可以叠加。值得注意的是，尽管模型参数少了 7000 万，CJJT 在罕见词测试集上的词错误率仍与采用基础 MHAT 的语言模型融合相近，甚至更低。采用 JEIT/CJJT MHAT 的语言模型融合还可额外获得 2.3%–12.8% 的相对收益，因此尽早进行 JEIT 或 CJJT 对语言模型融合极为有益。采用 CJJT 的语言模型融合优于采用 JEIT 的语言模型融合，这表明 JEIT、JOIST 和语言模型融合彼此互补。最后，我们进行 CJJT、最小词错误率（MWER）训练 [43] 并进行语言模型融合，在所有系统中取得了最佳词错误率，相较于基线的词错误率相对降低了 17.9%–31.3%。

5. 结论

我们提出 JEIT，通过单阶段联合训练将未配对文本注入内部语言模型。JEIT 简化了两阶段内部语言模型适应，并消除了 KLD 正则化，在罕见词测试集上相较基线实现了最高 10.2% 的相对词错误率降低。JEIT 后，MHAT 的表现优于 HAT，且在内部语言模型适应期间比 HAT 鲁棒得多。向编码器和解码器注入文本具有互补性，将二者结合（CJJT）可实现最高 16.4% 的相对增益。语言模型融合可使所有文本注入方法进一步获得最高 12.8% 的相对提升。

6. 参考文献

[1] C.-C. Chiu 等人，“采用序列到序列模型的最先进语音识别”，见 ICASSP 论文集，2018 年。[2] M. Jain、K. Schubert、J. Mahadeokar 等人，“采用改进束搜索的低延迟受控 RNN-T 自动语音识别”，arXiv 预印本 arXiv:1911.01629，2019 年。[3] T. Sainath、Y. He、B. Li 等人，“在质量和延迟方面超越服务器端传统模型的流式端侧端到端模型”，见 ICASSP 论文集，2020 年。[4] J. Li、R. Zhao、Z. Meng 等人，“开发性能超越高性能混合模型且具备定制能力的 RNN-T 模型”，Interspeech，2020 年。[5] A. Zeyer、A. Merboldt、R. Schl¨uter 和 H. Ney，“一种用于改进神经转导器的新训练流程”，arXiv 预印本 arXiv:2005.09319，2020 年。[6] T. Sainath、Y. He、A. Narayanan 等人，“一种高效的流式非循环端侧端到端模型，改进了罕见词建模”，2021 年。[7] A. Hannun 等人，“Deep Speech：扩展端到端语音识别”，arXiv 预印本 arXiv:1412.5567，2014 年。[8] C. Gulcehre、O. Firat、K. Xu 等人，“关于在神经机器翻译中使用单语语料库”，CoRR，2015 年。[9] E. McDermott、H. Sak 和 E. Variani，“端到端自动语音识别中语言模型融合的密度比方法”，见 ASRU 论文集。IEEE，2019 年。[10] E. Variani、D. Rybach 等人，“混合自回归转导器（HAT）”，arXiv 预印本 arXiv:2003.07705，2020 年。[11] Z. Meng、S. Parthasarathy、E. Sun 等人，“用于领域自适应端到端语音识别的内部语言模型估计”，见 SLT 论文集。IEEE，2021 年。[12] A. Zeyer、A. Merboldt、W. Michel 等人，“采用内部语言模型先验校正的 LibriSpeech 转导器模型”，arXiv 预印本 arXiv:2104.03006，2021 年。[13] N. Kanda、X. Lu 和 H. Kawai，“用于 CTC声学模型的基于最大后验概率的解码”，Interspeech，2016 年。[14] Z. Meng、Y. Wu、N. Kanda 等人，“结合语言模型融合的端到端语音识别最小词错误率训练”，Interspeech 论文集，2021 年。[15] D. Zhao、T. Sainath、D. Rybach 等人，“浅层融合端到端上下文偏置”，Interspeech，2019 年，第 1418–1422 页。[16] A. Rosenberg、Y. Zhang 等人，“采用增强合成语音的语音识别”，见 ASRU 论文集。IEEE，2019 年。[17] Y. Deng、R. Zhao、Z. Meng 等人，“利用结合神经文本转语音的半监督训练改进 RNN-T 的领域扩展”，Interspeech，2021 年，第 751–755 页。[18] X. Zheng、Y. Liu、D. Gunceler 等人，“利用合成音频改善端到端自动语音识别系统对未登录词的识别”，见 ICASSP 论文集。IEEE，2021 年。[19] A. Bapna、Y.-A. Chung、N. Wu 等人，“SLAM：通过语音文本联合预训练实现语音与语言建模的统一编码器”，arXiv 预印本 arXiv:2110.10329，2021 年。[20] Y. Tang、H. Gong 等人，“用于语音翻译和识别的统一语音文本预训练”，见 ACL 论文集，2022 年。[21] S. Thomas、H. J. Kuo、B. Kingsbury 和 G. Saon，“减少构建口语理解系统对语音训练数据需求的探索”，见 ICASSP 论文集，2022 年。

[22] Z. Chen、Y. Zhang、A. Rosenberg 等，《MAESTRO：通过模态匹配获得的匹配语音文本表示》，载于 Interspeech 论文集，2022 年。[23] T. Sainath、R. Prabhavalkar 等，《JOIST：用于自动语音识别的联合语音与文本流式模型》，SLT 论文集，2022 年。[24] Z. Meng、N. Kanda、Y. Gaur 等，《面向领域自适应端到端语音识别的内部语言模型训练》，载于 ICASSP 论文集，IEEE，2021 年。[25] J. Pylkk¨onen、A. Ukkonen、J. Kilpikoski 等，《RNN-Transducer 预测网络的快速纯文本领域自适应》，arXiv 预印本 arXiv:2104.11127，2021 年。[26] Z. Meng、Y. Gaur、N. Kanda 等，《利用纯文本数据对端到端语音识别进行内部语言模型自适应》，载于 Interspeech，2022 年。[27] X. Chen、Z. Meng 等，《用于高效语言模型自适应的因式分解神经转导器》，载于 ICASSP 论文集，2022 年。[28] Z. Meng、T. Chen、R. Prabhavalkar 等，《模块化混合自回归转导器》，载于 SLT 论文集，IEEE，2022 年。[29] C. Gao、G. Cheng、R. Yang 等，《使用非配对文本数据对端到端自动语音识别模型的 Transformer 解码器进行预训练》，载于 ICASSP 论文集，2021 年。[30] S. Kullback 和 R. A Leibler，《论信息与充分性》，《数学统计年鉴》，1951 年。[31] H. Liao、E. McDermott 等，《利用半监督训练数据为 YouTube 视频转录进行大规模深度神经网络声学建模》，载于 ASRU 论文集，IEEE，2013 年。[32] 《Google 的人工智能：我们的原则》，https: // ai.google/principles/，访问日期：2022-07-20。[33] C. Kim、A. Misra、K. Chin 等，《在虚拟房间中生成大规模模拟话语，以训练用于 Google Home 远场语音识别的深度神经网络》，2017 年。[34] J. Li、D. Yu、J.-T. Huang 等，《在 CD-DNN-HMM 中利用混合带宽训练数据改进宽带语音识别》，载于 SLT 论文集，IEEE，2012 年。[35] D. S Park、W. Chan、Y. Zhang 等，《SpecAugment：一种用于自动语音识别的简单数据增强方法》，Interspeech 论文集，2019 年。[36] X. Gonzalvo、S. Tazari、C. Chan 等，《Google 实时 HMM 驱动的单元选择合成器的最新进展》，2016 年。[37] S. Ding、W. Wang 等，《适用于动态模型规模的统一级联编码器自动语音识别模型》，Interspeech 论文集，2022 年。[38] A. Narayanan 等，《用于统一流式与非流式自动语音识别的级联编码器》，载于 ICASSP 论文集，IEEE，2021 年。[39] A. Narayanan 等，《使用流式端到端模型识别长篇语音》，载于 ASRU 论文集，IEEE，2019 年。[40] M. Schuster 和 K. Nakajima，《日语和韩语语音搜索》，载于 ICASSP 论文集，IEEE，2012 年，第 5149–5152 页。[41] R. Botros、T. N Sainath、R. David 等，《绑定与缩减 RNN-T 解码器》，Interspeech 2021 论文集，2021 年。[42] M. Ghodsi、X. Liu、J. Apfel 等，《采用无状态预测网络的 RNN-Transducer》，载于 ICASSP 论文集，2020 年。[43] R. Prabhavalkar、T. Sainath、Y. Wu 等，《基于注意力的序列到序列模型的最小词错误率训练》，载于 ICASSP 论文集，IEEE，2018 年。