arXiv:2301.02736v1 [eess.AS] 6 Jan 2023

在上下文端到端自动语音识别中使用外部离策略语音到文本映射

David M. Chan[⋆†] *Shalini Ghosh*[†] *Ariya Rastrow*[†] *Bj¨orn Hoffmeiste r*[†]

[⋆] 加利福尼亚大学伯克利分校 [†] 亚马逊 Alexa AI

摘要

尽管自动语音识别（ASR）模型的泛化性能有所提升，但针对下游任务专门优化自动语音识别模型仍是一项艰巨任务，主要原因是可用数据减少（因此需要收集更多数据），以及数据分布快速变化（因此需要更频繁地进行模型微调）。本研究探讨了利用外部知识的潜力，尤其是通过文本转语音方法生成的离策略键值库，以灵活地在训练后适应新的数据分布。我们的方法利用从文本转语音中提取的音频嵌入和语义文本嵌入，通过基于近似 k 近邻（KNN）的注意力融合步骤，为自动语音识别提供偏置。在 LibiriSpeech 以及内部语音助手/搜索数据集上的实验表明，与微调基线相比，所提出的方法最多可将领域自适应时间缩短 1K GPU 小时，同时使词错误率最多提升 3%。这表明，该方法有望用于在具有挑战性的零样本和少样本场景中适配生产环境中的自动语音识别系统。

索引词— 语音识别、迁移学习、微调、适应、上下文

1 引言

自动语音识别（ASR）面临的最具挑战性问题之一，是如何针对下游应用对大规模模型，尤其是语音编码器进行专门化。这些应用往往（a）标注训练样本较少，且（b）语音数据的分布变化迅速。解决这一问题的传统方法是频繁收集新数据，用于重新训练和专门化模型，并借助领域提示 [1], 增量学习 [2], 知识蒸馏 [3], 手写语法 [4], 或度量学习 [5, 6] 等工具，以降低为下游应用重新训练模型所带来的影响。遗憾的是，对于产品列表这类快速变化的数据，或需要针对每位客户进行专门化的应用而言，这些方法虽然有效，却要么速度本质上较慢，要么计算成本高到难以实现。

如需联系，请将问题发送至davidchan@berkeley.edu。此项工作是在亚马逊 Alexa AI 实习期间完成的。

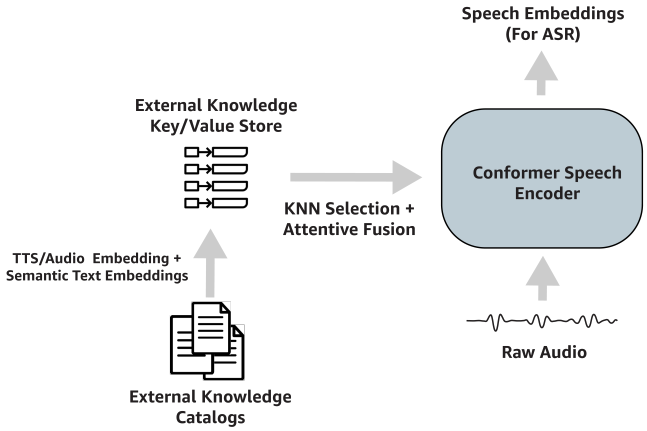

图 1：我们的方法概览：利用文本转语音映射实现上下文自动语音识别。我们利用文本目录中的数据生成音频和文本表示，从而建立从音频键到文本值的映射。为了在自动语音识别中利用这些映射，我们在微调（或训练）阶段于语音编码器中实现了 K 近邻注意力。

在这项工作中，我们提出了一种方法，利用外部文本目录——其中包含多达 1,000 万个专业词汇或短语的大型列表——来提升模型在微调过程中的性能，并在将已微调模型专门化以适应新数据集时进一步提升其性能。我们方法的主要亮点如下：首先，我们构建一个键值外部知识库，将目录中每个文本条目（通常包含 100 万至 1,000 万个示例）的音频表示映射为该文本的语义表示。接下来，我们训练一个模型，使其通过关注检索到的键值对来利用该外部知识库；这些键值对通过近似 k 近邻检索获得。依赖外部、固定且离策略的键值库，意味着可以在专门化过程中更新该知识库；每个新模型只需更新短语列表，无须额外微调。

受 Borgeaud 等人 [7] 和 Wu 等人 [8], 的启发，我们采用上下文嵌入方法，重点关注自动语音识别，利用文本转语音生成的音频数据和语义文本嵌入，为 Conformer 模型的语音编码器提供偏置。据我们所知，此前尚未有人使用文本转语音对文本上下文进行编码

进行过相关探索。

我们的主要贡献有三点：

1. 据我们所知，我们提出了首种利用大规模文本数据对语音编码器进行上下文偏置的方法。

2. 我们证明，将我们的方法与近似K近邻查找相结合，可降低自动语音识别模型的词错误率，尤其是在编码目录与目标领域相匹配的情况下。

3. 我们证明，在需要快速响应分布变化（例如，快速更新体育赛事目录或个人目录）的情况下，我们的方法无需重新训练模型即可提供准确的解决方案。

2 相关工作

利用额外上下文提升解码器和联合模型的性能，尤其是通过自动语音识别转换器中的深度融合技术，已得到相当充分的研究 [9, 10, 11, 12, 13, 14, 15, 16, 17]。这些方法大多由两个部分组成：(1) 为话语检索局部上下文线索的方法；(2) 将这些上下文线索与转换器堆栈中的联合模型或解码器融合的方法。不同方法的区别在于这两个部分的实现方式。我们的工作有几个关键的不同之处。我们主要关注语音编码器的深度偏置，而非将上下文网络与语音解码器进行浅层融合，或对语言模型生成的候选项重新排序。虽然从理论上讲，Sathyendra 等人 [18] 提出的方法可以用于深度融合，但相关工作并未探索这一方向。事实上，正如我们在表2中所展示的，深度融合是影响上下文学习技术质量的重要区别因素。此外，我们着重研究大规模上下文（$> 10K$ 个上下文条目），以探索其在领域专门化中的应用；而现有偏置方法通常侧重于个性化，处理的上下文最多为 $1K$ 个条目（且许多情况下仅处理 < 100 个条目）。同样，虽然理论上上述方法可以应用于更大规模的上下文，但我们发现，这需要弥补显著的实现差距，包括解决扩展性问题及效率问题（我们将在第4节中讨论）。

尽管解码器和联合模型的后期融合与偏置已得到充分研究，但对语音编码器本身进行偏置，尤其是采用早期/深度融合方法，仍是一个研究不足的领域。与我们提出的模型最接近的工作由 Chen 等人提出 [19], , 他们利用注意力机制处理一组局部的基于LSTM的字素/音素嵌入，以增强音频编码器。他们发现，每个话语仅使用 40 个上下文文本实体对编码器进行偏置，就能使专用测试数据集上的性能最高提升 75%。类似地，Sathyendra 等人 [18] 和 Chang 等人 [20]表明，将小规模（<100）上下文通过基于注意力的过程与语音模型和语言模型融合后，词错误率会降低。我们的方法有所不同，主要是为领域专门化而设计，而现有的偏置方法则侧重于个性化。这最明显地体现在目录规模上——在以往的工作中，每个话语的上下文最多包含 100 个话语，而我们使用的目录最多包含 1000 万个样本。因此，我们的模型旨在补偿一般领域偏移，而非对自动语音识别性能进行局部个性化改进。此外，我们的工作支持完全离策略的专门化。在现有工作中，每次推理过程都会重新编码上下文，因此缓存中间结果的机会有限。我们的上下文完全离线计算；目录更新不会影响模型训练，因此，生成专用目录可以与模型开发并行进行。

认为自动语音识别中的 Transformer 会从扩展上下文中受益，并非没有道理。在自然语言处理领域，Vaswani等人 [21] 表明，更长的记忆注意力上下文能够提升 Transformer 的性能；包括 Dai等人 [22], 和 Child等人 [23] 在内的一系列方法，则致力于增加每个自然语言序列可用上下文的长度。长上下文面临的主要问题是效率，因为 Transformer 的计算量会随上下文长度呈二次方缩放。为缩短处理时间，Dai等人 [22] 和 Child等人 [23] 等方法仅通过完整上下文中的一个子集计算梯度，而非遍历整个上下文，以节省内存和计算资源。Wu 等人 [8], 近期将这种方法具体化：他们证明，通过大型内存库缓存外部键值对来扩展标准文本 Transformer 的上下文，能够显著降低标准语言建模任务的困惑度。Wu 等人 [8] 采用 K近邻方法检索最相关的上下文元素，并且仅通过这些元素进行反向传播。尽管这种查找可能会引发一定的离策略漂移，但该方法十分有效，能够显著提升性能，尤其是在复制任务中；这类任务要求模型指向特定的先前元素，而较小上下文的模型可能无法访问这些元素。

当上下文来自外部，甚至与当前话语正交时，扩展上下文是否仍有帮助？似乎有一些证据支持这一观点：在标准自动语音识别流程之外，已有研究表明，利用大规模文本数据生成的外部记忆增强的模型，有望胜过规模相近但不具备外部知识的模型。Borgeaud 等人 [7] 最近证明，借助自然语言句子数据库进行外部知识检索，可以使从语言建模到问答等广泛的纯语言任务效率提升高达 25 倍。同样，知识增强学习已被证明广泛

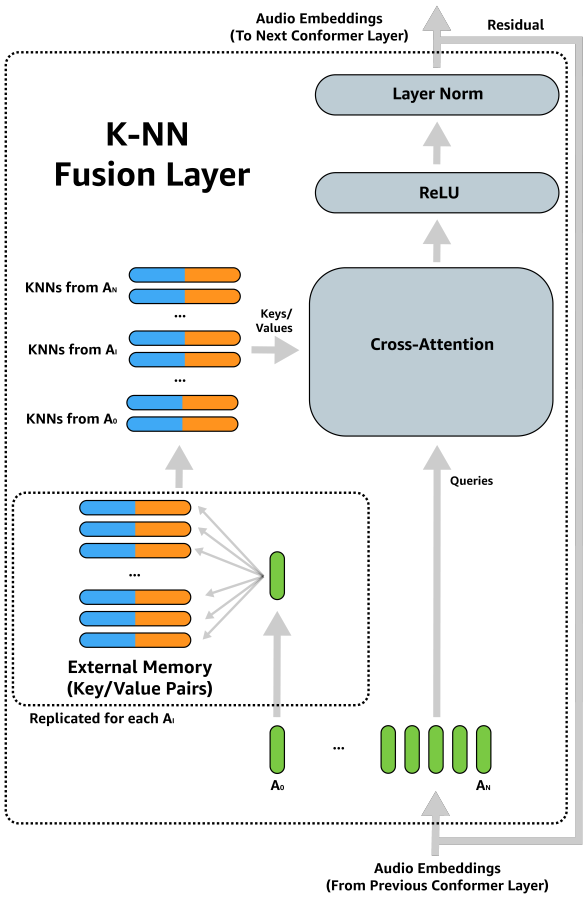

图 2：K近邻融合层概览。对于每个音频帧嵌入，我们使用目录中的音频键提取近似K近邻。这些近邻构成标准交叉注意力层的上下文键值存储 [21], , 其中查询是输入的音频帧嵌入。

适用于问答 [24, 25], 图像描述 [26] 及其他任务 [27, 28, 29, 30, 31, 32]。

3 方法

我们的方法概览见图 1。我们的方法包含两个关键组件：(1) 一种在可听语音与目录的文本表示之间生成键值映射的方法，我们称之为“外部记忆”；(2) 一种将“外部记忆”与现有语音编码器融合的基于注意力的模块。外部记忆能够离线、以离策略方式更新至关重要，因为这样可以低成本地修改记忆，而无需承担重新训练的成本。

3.1 生成外部记忆

外部记忆生成过程的概览如图 3所示。我们的方法生成的外部记忆由音频嵌入键/文本嵌入

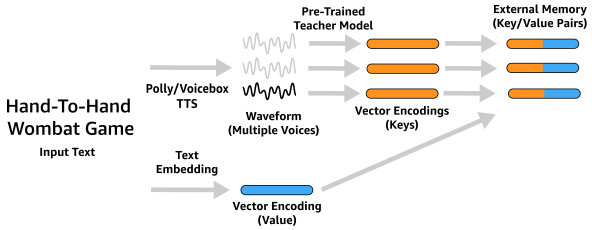

图 3：文本目录编码过程概览。对于每个目录条目，我们生成基于TTS的音频编码，作为键值对中的“键”向量。值是该条目的语义文本嵌入。键值对被组装成外部记忆，见图 2

.

值对构成，这些值对来自纯文本目录。为了生成音频嵌入键，我们使用文本转语音（TTS）生成音频数据的波形表示，然后使用预训练语音编码器模型对这些波形表示进行嵌入。为了生成文本嵌入值，我们采用现成的语义文本嵌入方法，包括独热编码、GLoVE [33] 和 BERT 风格嵌入 [34]方法。

3.1.1 文本转语音

在我们的工作中，我们采用两个文本转语音模块来生成音频嵌入所需的音频：Amazon Polly 文本转语音服务，以及 Alexa-AI 的内部文本转语音（TTS）库。该库经过优化，可为 ASR 模型训练和测试生成数据。由于它能够生成多种声音，我们将其称为 Multivoice-TTS。对于 Amazon Polly 和 Multivoice-TTS，我们各使用十种声音，主要来自英语（美国）和英语（英国）地区设置。在每个话语前后均插入 0.1 秒的静音。

3.1.2 音频嵌入

尽管外部目录的音频嵌入可以通过多种方式构建，与 Wu 等人类似， [8], 我们希望音频嵌入尽可能接近符合策略的自注意力嵌入。因此，我们将基线模型（未经微调）某个中间层的自注意力表示的均值作为音频嵌入。

3.1.3 文本嵌入

在我们的工作中，我们探索了几种生成文本嵌入的方法，这些嵌入构成记忆键值对中的值。对于小型目录，我们探索了在训练过程中构建的学习得到的独热嵌入。虽然此类嵌入可以带来更好的性能（因为它们是针对每项任务显式构建的），但它们无法扩展——因为不能离线计算（因此也无法在测试时插入）。为了生成可扩展的文本嵌入，我们探索了两种语义文本嵌入方法：基于词语共现概率构建的 GLoVE 嵌入 [33], , 以及 BERT-

https://aws.amazon.com/polly/

风格嵌入 [34], , 它们是通过大型统计模型学习得到的。GLoVE 嵌入为 300 维，使用公开可用的向量计算得出；而我们的 BERT 风格嵌入则使用`all-MiniLM-L6-v2`模型，通过`sentence-transformers`软件包计算得出 [35]。

3.2 外部记忆融合

外部记忆融合过程概述如图 2所示。我们提出的方法中的语音编码器基于 Conformer 编码器 [36], , 并增设了 K 近邻（ KNN ）融合层。在每个 KNN 融合层中，对于话语 a_i 的每个音频帧嵌入 A，我们都会从外部记忆 $E = (k_i, v_i)\ 1 \leq i \leq |E|$ 中查询一组 m 最近邻:

$$\mathcal{N}_{a_i} = \underset{N \subset E, |N|=m}{\arg\min} \sum_{(k_j, v_j) \in N} ||k_j - a_i||_2^2 \qquad (1)$$

然后，我们将该层的上下文构造为 $\mathcal{C} = \cup_{a_i \in A} \mathcal{N}_{a_i}$。根据 $\mathcal{C}$，我们可以构造两个矩阵 $K_c \in \mathbb{R}^{m|A|,\ d_{\text{key}}}$ 和 $V_c \in \mathbb{R}^{m|A|,\ d_{\text{value}}}$，分别由键和值组成。随后，K-NN 融合层的输出为:

$$F(A, E) = A + \text{LN}\left(\text{ReLU}\left(\text{softmax}\left(\frac{(AW_q)K_c'}{\sqrt{d}}\right)(V_c W_v)\right)\right) \qquad (2)$$

其中，LN 表示层归一化。遗憾的是，由于我们处理的是大型目录，计算公式 1的开销可能非常大。因此，我们不计算精确的最近邻，而是采用计算效率高得多的近似最近邻。为了从大规模目录中高效提取近似最近邻，我们利用 FAISS [37]库生成经过优化乘积量化变换的键（64 维） [38],，并使用一个包含 2048 个质心、采用乘积量化快速扫描编码的分层可导航小世界（HNSW）索引进行搜索 [39]。即使目录包含超过 700 万组键值对，这种方法也只会使前向传播延迟增加 15%。

3.3 实验设计

3.3.1 ASR 基础模型

虽然在实践中，我们的方法可以应用于许多不同的语音编码器，但我们使用 Conformer 编码器 [36]。对于解码器，我们采用一个隐藏维度为 320 的单层 LSTM 解码器，不使用显式的预训练语言模型。虽然我们探索了多种编码器规模，但对于 Librispeech，我们主要遵循 Gulati 等人 [36] 的做法，使用一个 16 层、隐藏维度为 144 的编码器（参数量为 10.3M）。对于内部 Alexa-AI 数据，我们使用一个含有 208.37M 个参数的 Conformer 模型。所有模型均采用 ReLU 激活、批归一化以及丢弃率为 0.1 的丢弃。对于 ASR 分词，我们使用 SentencePiece 模型 [40] ，词汇量大小分别为 640（ Librispeech）和 4096（内部数据）。

表 1：采用小型模型（10.3M 参数）时，在 Librispeech 数据上的词错误率；模型采用了不同的文本转语音、文本嵌入和每帧神经网络数（K）配置。MV-TTS 指 Multivoice-TTS。

目录	TTS	Text	K	测试集清洁子集	其他测试集
基线				5.77	13.34
训练	Polly	独热编码	4	5.75 (0.34%)	13.30 (0.29%)
	Polly	独热编码	8	5.72 (0.86%)	13.19 (1.10%)
	Polly	独热编码	16	5.71 (1.03%)	13.15 (1.42%)
	Polly	BERT	8	5.74 (0.52%)	13.26 (0.60%)
	MV-TTS	独热编码	8	5.52 (4.33%)	12.96 (2.84%)
	MV-TTS	BERT	8	5.68 (1.63%)	13.05 (2.18%)
Test	Polly	GLoVE	8	6.33 (-8.84%)	14.56 (-9.15%)
	Polly	BERT	8	5.71 (1.03%)	13.24 (0.75%)
	MV-TTS	GLoVE	8	6.15 (-6.17%)	14.32 (-6.84%)
	MV-TTS	BERT	8	**5.34** (8.05%)	**12.84** (3.86%)

表2：使用 Multivoice-TTS、BERT嵌入和每帧8个神经网络的模型，在不同层加入目录数据后，Librispeech 测试集词错误率的相对改进。

数据集	1	3	12	16	3,12	all
清洁子集	1.02%	3.65%	6.65%	2.63%	7.79%	8.05%
其他	0.71%	2.88%	2.97%	1.08%	3.41%	3.86%

3.3.2 目录数据源

在我们的工作中，我们探索了多种不同的目录数据源。对于 Librispeech，我们使用训练数据集或测试数据集中出现的 2500 个最罕见词元构建了一个模拟目录。为训练数据和测试数据分别构建独立目录，使我们能够探索模型在测试时面对目录分布偏移时的表现。我们的内部 Alexa 目录聚焦于媒体领域的助手查询，包含 15K 个电影名称。

3.3.3 训练详情

对于 Librispeech，该模型使用 TensorFlow 实现，并使用 24 个 Nvidia V-100 GPU 训练 120 轮次，批次大小为 2048，采用 Adam优化器，学习率为 $3e^{-4}$。对于 Alexa-AI数据集，该模型使用 104 个 Nvidia V-100 GPU 微调 30 轮次，批次大小为 832，采用 Adam优化器，并使用预热/保持学习率调度，预热步数为 10,000，最大学习率为 $5e^{-3}$。

4 结果与讨论

在这项工作中，我们展示了两个数据集的结果——Librispeech [41], , 一个包含 960 小时相对清洁、带标注的自动语音识别数据的数据集，以及一个侧重于以媒体为中心的查询的内部 Alexa 数据集。

4.1 Librispeech

我们在 Librispeech 上的主要结果如表 1所示。总体而言，使用额外数据增强模型

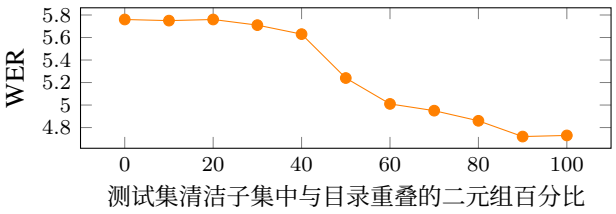

图4：Librispeech 测试集清洁子集在不同测试目录上的词错误率。随着测试目录中与测试数据集重叠的二元组百分比增加，目录增强模型的性能也随之提升。

表3：采用不同模型参数、结合多说话人文本转语音、BERT 和每帧 8 个神经网络时，相较于基线微调，Librispeech 测试集的词错误率相对改善幅度。

数据集	5M	10M	50M	100M	300M
清洁子集	28.9%	8.05%	4.28%	1.66%	0.08%
其他	19.3%	3.86%	2.65%	-0.07%	0.01%

表 4：Alexa-AI 性能。T-C：目录生成时间。T-FT：微调时间。Multivoice-TTS、BERT 和每帧 8 个神经网络。

模型/测试数据	T-C (分钟)	T-FT (GPU 小时)	相对文本转语音词错误率
B_{FT}/训练-文本转语音	0	2048	7.1%
B_{cat}/训练-文本转语音	33	1600	6.8%
B_{FT}/测试-文本转语音	-	-	0.52%
B_{cat}/测试-文本转语音	-	-	4.12%
$B_{\text{FT}+T}$/测试-文本转语音	0	1024	19.66%
$B_{\text{cat}+T}$/测试-文本转语音	28	0	21.27%

比不使用外部数据的模型表现更强。对于 Librispeech，使用训练目录进行训练、使用测试目录进行测试时，我们获得了出色的迁移表现，优于训练和测试均使用训练目录的情况，这表明还实现了额外的零样本专门化。虽然独热向量的表现优于 BERT 向量，但我们必须针对每个目录训练这些向量，因此无法进行测试时专门化。在所有情况下，BERT 的表现均优于 GLoVE（使用 GLoVE 会导致测试时专门化的表现下降）。图4 表明，我们的方法能够捕捉并利用目录中的领域数据。在这项实验中，模型使用包含 30 万个训练集独有二元组的目录进行训练；我们展示了该模型在十个测试目录上的表现，每个测试目录都包含 3 万个二元组，数据取自测试集或开发集。随着测试数据中可在测试目录中找到的二元组比例增加，模型表现也随之提升——这表明，在零样本学习设置下，我们的方法能够有效利用测试目录中的信息。

消融实验：表2 探讨了将外部知识增强置于模型不同层时模型的性能。尽管让所有层都能获取外部知识是最有效的方法，但我们发现这种方法会导致延迟过高，因为它会使模型一次前向传播的延迟增加约 85%。使用单层仅会使延迟增加约 15%，而使用两层则会使延迟增加约 23%。表3探讨了随着模型参数数量增加，该方法在 Librispeech 上的性能。随着参数数量增加，外部记忆带来的收益会降低。

4.2 Alexa

为了进一步验证我们的方法，我们还通过真实场景模拟，考察模型对测试数据的泛化能力。我们从一个基线模型开始 B (参见：第 3 节)，并训练了两个衍生模型：B_{FT}，在 Alexa 的 TTS 目录（$\mathcal{C}$，第 3 节）以及额外的 12 万小时去标识化 Alexa 数据上进行微调，$\mathcal{D}$，以及 B_{cat}，该模型采用我们的方法，在 $\mathcal{D}$上进行微调，并使用目录 $\mathcal{C}$。结果（表4，第 1/2 行）表明，即使 GPU 小时数显著减少，我们的方法仍能取得相近的词错误率。将模型迁移到测试数据集（不更新目录）时，从表4 （第 3/4 行）可以看出，我们训练的模型性能更好，这表明与仅通过微调训练的模型相比，该模型学会了更好地泛化。

最后，我们更新微调模型和目录模型，将测试数据纳入其中。测试数据通过额外的 GPU 训练整合到微调模型中，而在目录模型中，测试数据则通过目录生成和拼接的方式纳入。表 4 （第 5/6 行）进一步表明，即使不进行额外的 *GPU* 训练，我们的方法（$B_{\text{cat}+T}$）也能达到与

微调（$B_{\text{FT}+T}$）方法。

5 结论

本文介绍了首个利用纯文本目录数据，对语音编码器表示进行大规模上下文化的方法。尽管本文是构建上下文化语音编码器的良好开端，但研究目录的嵌入、利用字素/音素嵌入等问题，仍是值得探索的未来研究方向。该方法提供了一种自然的方式，利用外部记忆来应对开发集/测试集中的未登录词所导致的分布偏移，确保训练数据中的罕见词得到识别，处理个性化需求，并使用发音而非文本转语音——我们希望在真实世界的数据上评估该方法的这些特性。

6 参考文献

[1] S. Dingliwa、A. Shenoy、S. Bodapati、A. Gandhe、R. T. Gadde 和 K. Kirchhoff，“领域提示：迈向自动语音识别系统中兼顾内存与计算效率的领域自适应”，载于*Interspeech 2022*，2022。[2] D. Baby、P. D’Alterio 和 V. Mendelev，“基于循环神经网络转导器的语音识别模型的增量学习”，载于*Interspeech 2022*，2022。[3] K. Zhao、H. D. Nguyen、A. Jain、N. Susanj、A. Mouchtaris、L. Gupta 和 M. Zhao，“通过模块替换进行知识蒸馏：用于循环神经网络转导器自动语音识别”，载于*Interspeech 2022*，2022。[4] A. Gandhe、A. Rastrow 和 B. Hoffmeister，“面向口语对话系统的可扩展语言模型自适应”，载于*2018 IEEE* 口语技术研讨会 *(SLT)*。IEEE，2018，第 907–912 页。[5] T. P´apai、S. Ghosh 和 H. Kautz，“在训练马尔可夫逻辑网络时结合主观概率与数据”，第 7523 卷，2012 年 09 月，90–105 页。[6] S. Mahadevan、B. Mishra 和 S. Ghosh，“利用流形上的度量学习进行领域自适应的统一框架”，*CoRR*，第 abs/1804.10834 卷，2018。[7] S. Borgeaud、A. Mensch、J. Hoffmann、T. Cai、E. Rutherford、K. Millican、G. B. Van Den Driessche、J.-B. Lespiau、B. Damoc、A. Clark 等，“通过从数万亿词元中检索来改进语言模型”，载于国际机器学习会议。PMLR，2022，第 2206–2240 页。[8] Y. Wu、M. N. Rabe、D. Hutchins 和 C. Szegedy，“记忆型 Transformer”，*arXiv* 预印本 *arXiv:2203.08913*，2022。[9] S. Novotney、S. Mukherjee、Z. Ahmed 和 A. Stolcke，“线索向量：基于多种上下文信号的语言模型模块化训练”，*arXiv* 预印本 *arXiv:2203 08774*，2022。。[10] A. Shenoy、S. Bodapati 和 K. Kirchhoff，“目标导向型对话代理中用于语音识别的语言模型上下文偏置”，*arXiv* 预印本 *arXiv:2103.10325*，2021。[11] D. Zhao、T. N. Sainath、D. Rybach、P. Rondon、D. Bhatia、B. Li，和 R. Pang，“浅层融合端到端上下文偏置。”载于*I nterspeech*，2019年，第 1418–1422 页。

[12] B. Liu, L. Lu, “利用循环神经网络进行对话上下文语言建模”，载于 *2017 IEEE* 国际声学、语音与信号处理会议（*ICASSP*）。IEEE，2017，第 5715–5719 页。

[13] A. Jaech 和 M. Ostendorf，《用于查询自动补全的个性化语言模型》，*arXiv* 预印本 *arXiv:1804 09661*，2018。.[14] S. Kim 和 F. Metze，《对话上下文感知端到端语音识别》，载于 *2018 IEEE* 口语技术研讨会（*SLT*）。IEEE，2018 年，第 434–440 页。

[15] R. Liu, S. Yavuz, M. Li, “用于文档建模的分层循环神经网络”，载于*2015*年自然语言处理实证方法会议论文集，2015年，第899–907页。

[16] I. Williams, A. Kannan, P. S. Aleksic, D. Rybach 和 T. N. Sainath，“在端到端神经网络系统中使用束搜索进行上下文语音识别。”发表于 *Interspeech*，2018 年，第 2227–2231 页。

[17] T. Munkhdalai, K. C. Sim, A. Chandorkar, F. Gao, M. Chua、T. Strohman 和 F. Beaufays，《利用神经联想记忆实现设备端个性化语音识别的快速上下文适配》，载于*ICASSP 2022-2022 IEEE* 国际声学、语音与信号处理会议（*ICASSP*）。IEEE，2022 年，第 6632–6636 页。

[18] K. M. Sathyendra, T. Muniyappa, F.-J. Chang, J. Liu, J. Su, G. P. Strimel, A. Mouchtaris, and S. Kunzmann，“神经转导器中的个性化语音识别上下文适配器”，载于*ICASSP 2022-2022 IEEE*国际声学、语音与信号处理会议 *(ICASSP)*。IEEE，2022年，第8537–8541页。

[19] Z. Chen, M. Jain, Y. Wang, M. Fuegen, “用于上下文端到端自动语音识别的联合字素和音素嵌入》。载于*Interspeech*，2019年，第 3490–3494 页。

[20] F.-J. Chang、J. Liu、M. Radfar、A. Mouchtaris、M. Omologo、A. Rastrow 和 S. Kunzmann，“用于语音识别的上下文感知型 Transformer 转导器”，载于 *2021*年 *IEEE* 自动语音识别与理解研讨会 *(ASRU)*。IEEE，2021年，第 503–510 页。[21] A. Vaswani、N. Shazeer、N. Parmar、J. Uszkoreit、L. Jones、A. N. Gomez、L. Kaiser 和 I. Polosukhin，“注意力就是你所需要的一切”，神经信息处理系统进展，第 30 卷，2017年。[22] Z. Dai、Z. Yang、Y. Yang、J. Carbonell、Q. V. Le 和 R. Salakhut- dinov，“Transformer-XL：超越固定长度上下文的注意力语言模型”，*arXiv* 预印本 *arXiv:1901.02860*，2019年。[23] R. Child、S. Gray、A. Radford 和 I. Sutskever，“使用稀疏 Transformer 生成长序列”，*arXiv* 预印本 *arXiv:1904.10509*，2019年。[24] K. Marino、M. Rastegari、A. Farhadi 和 R. Mottaghi，“OK-VQA：需要外部知识的视觉问答基准测试”，载于 *IEEE/CVF* 计算机视觉与模式识别会议论文集，2019年，第 3195–3204 页。

[25] X. Pan、K. Sun、D. Yu、J. Chen、H. Ji、C. Cardie 和 D. Yu，“利用外部知识改进问答”，arXiv 预印本 arXiv:1902.00993，2019年。[26] Q. Wu、C. Shen、P. Wang、A. Dick 和 A. Van Den Hengel，“基于属性和外部知识的图像描述与视觉问答”，IEEE模式分析与机器智能汇刊，第 40 卷，第 6 期，第 1367–1381 页，2017年。[27] Z. Tang、S. Gu、J. Bao、D. Chen 和 F. Wen，“改进的向量量化扩散模型”，arXiv 预印本 arXiv:2205 16007，2022年。。[28] U. Khandelwal、A. Fan、D. Jurafsky、L. Zettlemoyer 和 M. Lewis，“最近邻机器翻译”，arXiv 预印本 ar Xiv:2010.00710，2020年。[29] A. Goyal、A. Friesen、A. Banino、T. Weber、N. R. Ke、A. P. Badia、A. Guez、M. Mirza、P. C. Humphreys、K. Konyushova 等人，“检索增强强化学习”，载于《国际机器学习会议》。PMLR，2022年。[30] X. Xie、J. Niu、X. Liu、Z. Chen、S. Tang 和 S. Yu，“将领域知识融入医学图像分析深度学习的综述”，《医学图像分析》，第 69 卷，101985，2021年。[31] A. J. Kumar、C. Morales、M.-E. Vidal、C. Schmidt 和 S. Auer，“在自动语音识别中使用知识图谱对 N最佳列表进行重评分”，arXiv 预印本 arXiv:1705.08018，2017年。[32] T. Tran、V. Le、H. Le 和 T. M. Le，“从深度学习到深度推理”，载于第 27 届 ACM SIGKDD知识发现与数据挖掘大会论文集，2021年，第 4076–4077 页。[33] J. Pennington、R. Socher 和 C. D. Manning，“GLoVE：用于词语表示的全局向量”，载于《2014年自然语言处理经验方法会议 (EMNLP) 论文集》，2014年，第 1532–1543 页。[34] J. Devlin、M.-W. Chang、K. Lee 和 K. Toutanova，“BERT：用于语言理解的深度双向 Transformer预训练”，arXiv 预印本 arXiv:1810.04805，2018年。[35] N. Thakur、N. Reimers、J. Daxenberger 和 I. Gurevych，“增强型SBERT：用于改进成对句子评分任务双编码器的数据增强方法”，载于《2021年计算语言学协会北美分会：人类语言技术会议论文集》。线上：计算语言学协会，2021年6月，第 296–310 页。 [线上]。可访问：https://www.aclweb.org/anthology/2021.naacl-main.28 [36] A. Gulati、J. Qin、C.-C. Chiu、N. Parmar、Y. Zhang、J. Yu、W. Han、S. Wang、Z. Zhang、Y. Wu 等人，“ Conformer：用于语音识别的卷积增强型Transformer”，arXiv 预印本 arXiv:2005.08100，2020年。[37] J. Johnson、M. Douze 和 H. J´egou，“利用图形处理器进行十亿规模的相似性搜索”，IEEE大数据汇刊，第 7 卷，第 3 期，第 535–547 页，2019年。[38] T. Ge、K. He、Q. Ke 和 J. Sun，“优化的乘积量化”，IEEE模式分析与机器智能汇刊，第 36 卷，第 4 期，第 744–755 页，2013年。[39] Y. A. Malkov 和 D. A. Yashunin，《利用分层可导航小世界图进行高效、稳健的近似最近邻搜索》，*IEEE*模式分析与机器智能汇刊，第 42 卷，第 4 期，第 824–836 页，2018 年。[40] T. Kudo 和 J. Richardson，《Sentencepiece：一种用于神经文本处理的简单、语言无关的子词分词器和去分词器》，*arXiv* 预印本 *arXiv:1808.06226*，2018 年。[41] V. Panayotov、G. Chen、D. Povey 和 S. Khudanpur，《LibriSpeech：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*国际声学、语音与信号处理会议（*ICASSP*）。IEEE，2015 年，第 5206–5210 页。