通过多码本向量量化知识蒸馏提升全格式端到端语音识别

Jian You 思科系统 上海，中国
jianyou@cisco.com

Xiangfeng Li 思科系统 上海，中国
xiangfel@cisco.com

Erwan Zerhouni 思科系统 苏黎世，瑞士
ezerhoun@cisco.com

摘要—**传统自动语音识别（ASR）模型通常会生成经过规范化、缺少标点符号和大小写信息的文本，因此需要借助后处理模型提升可读性。然而，由于采用级联系统设计，这种方法会增加系统复杂度和延迟。为应对这一挑战，研究者日益倾向于开发能够直接预测标点符号和大小写的端到端（E2E）自动语音识别模型，但该领域仍未得到充分探索。本文提出一种增强型全格式端到端自动语音识别模型，通过多码本向量量化（MVQ）实现知识蒸馏（KD）。实验结果表明，无论是否包含标点符号和大小写信息，我们的模型在词错误率（WER）方面均显著优于以往方法，在标点符号错误率（PER）方面也有显著提升。在 LibriSpeech-PC 的 test-clean 和 test-other 子集上进行的评估显示，我们的模型取得了当前最优结果。**

关键词——自动语音识别、标点符号、大小写、知识蒸馏

I. 引言

在传统的自动语音识别（ASR）模型中，输出文本通常采用小写且不带标点符号，这会显著降低文本的可读性和用户体验。缺乏适当的格式不仅妨碍文本的直接使用，还会对摘要、命名实体识别（NER）和神经机器翻译（NMT）等下游应用的性能产生不利影响，因为这些应用依赖格式良好的文本。因此，传统ASR系统无法生成全格式输出，削弱了其在更复杂自然语言处理任务中的实用性。这一缺陷使得必须使用额外的后处理模型来对文本进行格式化。

众多研究致力于对未格式化文本进行标点符号和大小写恢复。早期研究主要采用卷积神经网络（CNN）和循环神经网络（RNN）进行标点恢复 [1, 2, 3]。随着Transformer模型的出现及其日益普及，[4], 与 RNN 相比，该模型以更优异的性能以及捕捉更长序列中上下文信息的能力而著称，近年来许多研究已采用 Transformer 来完成这一任务 [5, 6, 7]。然而，在传统自动语音识别系统中集成用于预测标点符号和大小写的后处理模块，仍面临诸多挑战。首先，这些模块通常忽略韵律等声学特征，而这些特征对于准确进行正字法判断至关重要，例如区分陈述句和疑问句。仅依赖词汇特征会增加系统对错误和误解的易感性。此外，将多个独立模型串联起来进行格式化不仅需要大量维护工作，还会引入延迟，从而使系统效率更加复杂。

鉴于级联系统固有的局限性，近期研究日益关注将格式化功能整合到传统ASR系统中。这种创新性的端到端方法能够直接从声学特征映射生成全格式文本词元序列。端到端格式化语音识别最初在[8], 中进行了探索，其中系统能够直接从语音输入输出区分大小写的文本。随后，[9] 和 [10] 开发了能够以端到端方式直接从语音生成带标点文本的系统。在[11], 中，研究人员采用堆叠式Transformer编码器层，并使用配对的语音与带标点文本数据，通过连接时序分类（CTC）损失进行训练。类似地，[12] 采用Transformer模型，同时引入了专门用于标点符号的独立解码器。尽管名为Whisper [13] 的现成Transformer-Transformer架构证明了ASR模型可以训练为生成格式化文本，但其无法在流式模式下运行，限制了其在实时场景中的适用性。[14] 和[15] 均使用带有标点符号和大小写的文本，对基于Conformer的[16] 端到端模型进行训练，并在公开语料库上取得了良好性能。在 [17] 和 [18], 的研究中，预训练大语言模型（LLM）被整合到端到端ASR系统中，以实现全格式转录输出。

尽管已有一些关于全格式端到端自动语音识别系统的研究，但这一领域仍未得到充分探索。本研究提出了一种增强型全格式端到端自动语音识别模型，该模型基于通过多码本向量量化实现的知识蒸馏。具体而言，我们采用类 Transformer 转导器系统，在带格式文本上使用循环神经网络转导器（RNN-T）[19] 损失进行训练。我们选择近期发布的 Zipformer [20] 作为声学编码器，因为它速度快、性能出色。受先前研究[11] 的启发，该研究根据中间层的输出和不含标点符号的词元序列计算辅助 CTC 损失，我们采用知识蒸馏框架[21]，使声学编码器的中间层能够从自监督预训练教师模型的另一个中间层的嵌入中学习。据我们所知，尽管知识蒸馏已广泛应用于传统自动语音识别系统[22, 23, 24], ，我们的研究首次尝试以端到端方式通过知识蒸馏增强全格式自动语音识别。引入知识蒸馏的目的是使声学编码器的中间层能够更有效地促进准确的文本预测。标点预测一方面

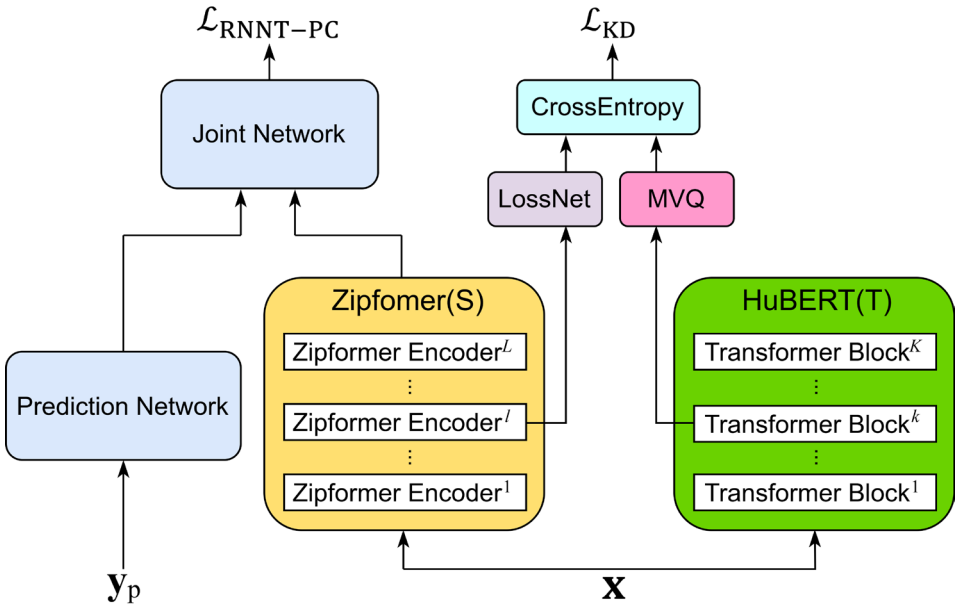


图 1。所提出模型的架构。学生模型是 RNN-T 中的声学编码器 Zipformer。教师模型是经过微调的 HuBERT 版本。 $\mathbf{x}$ 是输入语音序列，$\mathbf{y}_{\text{p}}$ 是此前预测的非空白标签序列。

受韵律特征影响；另一方面，由于 RNN-T 解码器会根据先前生成的非空白标签序列来确定每个新输出，因此标点预测也取决于先前预测文本所提供的上下文。因此，先前预测的文本越准确，模型准确预测标点符号的能力就越强。此外，大小写和标点预测能够相互促进 [7], ，标点预测的改进也能提升大小写预测的准确性。在 LibriSpeech-PC [15] 数据集上进行的实验表明，所提出的模型性能更优，验证了其有效性。

II. 所提出的方法

A. 基于 *RNN-T* 的带标点和大小写自动语音识别

在我们的方法中，我们采用称为 RNN-T 的神经转导器系统，这是因为它简单且在流式模式下具有出色的性能。端到端自动语音识别本质上可以表述为序列到序列问题。我们将输入语音话语表示为 $\mathbf{x} = \{\mathbf{x}_1 \dots \mathbf{x}_T\}$，其中 T 为帧数。包含标点符号和混合大小写词元的 WordPiece 词表表示为 Y。相应的、以**标**点符号和**大**小写格式表示的标签序列记为 $\mathbf{y}_{\text{pc}} = \{y_1 \dots y_U\}$，其中 U 为标签序列的长度，且 $y_u \in Y$。

我们将 Y_b 定义为 $Y \cup \{\langle \text{b} \rangle\}$，其中 $\langle \text{b} \rangle$ 表示空白标签。将 Y_b^* 记为输出空间 Y_b上的所有序列的集合，并将元素 $\mathbf{a} \in Y_b^*$ 记为对齐序列。后验概率表示为：

$$P(\mathbf{y}_{\text{pc}}|\mathbf{x}) = \sum_{\mathbf{a} \in \mathcal{B}^{-1}(\mathbf{y}_{\text{pc}})} P(\mathbf{a}|\mathbf{x}) \tag{1}$$

其中，$\text{B} : Y_b^* \to Y^*$ 是一个从对齐序列 $\mathbf{a}$中移除空白符号的函数。对齐概率 $P(\mathbf{a}|\mathbf{x})$ 的公式基于以下假设：时刻 t 的输出在条件上依赖于先前非空白预测序列：

$$P(\mathbf{a}|\mathbf{x}) = \prod_{\tau} P(a_\tau|\mathbf{p}, \mathbf{x}) \tag{2}$$

其中，a_τ 表示位置 τ处的输出，$\mathbf{p}$ 表示预测网络的非空白输出，而 $P(a_\tau|\mathbf{p}, \mathbf{x})$ 表示在位置 τ观察到 a_τ 的概率。因此，RNN-T 损失函数为后验概率的负对数，如公式 (1) 所示：

$$\mathcal{L}_{\text{RNNT-PC}} = -\log P(\mathbf{y}_{\text{pc}}|\mathbf{x}) \tag{3}$$

B. 基于 *MVQ* 的知识蒸馏

知识蒸馏（KD）是提升基于 RNN-T 的自动语音识别系统性能的一种常用方法 [22, 25]。与传统知识蒸馏中教师标签通常为分布格 [25, 26] 或浮点型嵌入 [27, 28], 不同，MVQ-KD 将教师模型中间层的嵌入压缩为码本索引（CI）。随后训练学生模型预测其码本索引。设 $\mathbf{T}_t^k$ 为从教师模型第 k 层提取的嵌入，$\mathbf{S}_t^l$ 为学生模型第 l 层在第 t 帧的嵌入。令 $\boldsymbol{i}_t = (i_{t,1}, \dots, i_{t,N})$ 为 $\mathbf{T}_t^k$压缩后的码本索引，其中 N 是码本数量，$i_{t,n}$ 表示选中了第 n 个码本中的哪个条目。知识蒸馏损失函数为：

$$\mathcal{L}_{\text{KD}} = \sum_{t=1}^{T} \sum_{n=1}^{N} \text{CrossEntropy}\left(\boldsymbol{i}_t, \text{LossNet}(\mathbf{S}_t^l)\right) \tag{4}$$

其中，LossNet 是一个神经网络，通过带有 softmax 激活的线性层将学生嵌入转换为概率。在每个时间戳 t，MVQ-KD 都会执行 N 次独立分类。我们采用 8 位整数表示 $\boldsymbol{i}_t$ 的每个条目。例如，当 N =8 时，序列 (24, 0, 1, 27, 255, 12, 17, 10) 可作为码本索引（CI）的示例。

鉴于本研究中教师嵌入的维度为 512，若使用 N=16，则可实现 128 的压缩率。通过 MVQ-KD，可以预先计算教师标签，并以较高的压缩率保存到磁盘，从而显著缓解传统 KD 中的计算和存储问题。学生模型和教师模型的详细配置见第 III.B 节。

C. 与辅助损失融合

我们采用公式（3）作为主要损失函数，该函数足以用于训练端到端格式的自动语音识别模型。不同于 [11], 中采用由不含标点符号的词元序列和中间层输出所产生的辅助 CTC 损失来促进模型训练的方法，我们采用公式（4）中定义的知识蒸馏损失作为辅助损失函数。通过对这两个损失函数进行加权组合，实现与辅助损失的融合：

$$\mathcal{L} = \mathcal{L}_{\text{RNNT-PC}} + \alpha \mathcal{L}_{\text{KD}} \tag{5}$$

其中，α 是用于控制蒸馏损失权重的超参数。

与知识蒸馏损失的融合通过使学生模型能够学习教师模型中间层的丰富信息来增强学生模型。所提出模型的架构如图 1 所示。在推理过程中，按照第 2.2 节所述，LossNet 会被移除，因此不会增加推理时间开销。

III. 实验设置

A. 数据集

我们使用 LibriSpeech-PC [15] 数据集进行训练，该数据集通过恢复标点符号和大小写，增强了 LibriSpeech 语料库 [30] 的原始转录文本。在恢复过程中，如果样本包含非标准 Unicode 字符、较大的文本偏差或时长不足一秒，则会将其排除。恢复完成后，LibriSpeech 语料库中 960 小时音频数据的 90% 以上得以保留。评估在 test-clean 和 test-other 子集上进行。

LibriSpeech-PC 包含种类繁多的标点符号和单词大小写形式。在本实验中，我们对转录文本执行以下预处理步骤：1）仅关注句号、逗号和问号；由于其他标点符号在数据集中的出现频率较低，因此将其排除；2）用空格分隔单词和标点符号；3）将多个连续空格合并为一个空格。

B. 实现细节

采用在整个 LibriSpeech 数据集上微调的超大型 HuBERT [29], 作为教师模型[1]。中型 Zipformer [20] 被用作学生模型。超大型 HuBERT 模型共包含 48 层。根据 [21], 中的选择，我们提取第 36 个 Transformer 块的嵌入来生成码本索引（CI），因为该层信息丰富，同时也较容易学习。对于学生模型，按照 [20], 中的经典设置，其六个堆栈的层数分别为 {2, 2, 3, 4, 3, 2}。我们在学生模型的第 4 个堆栈（具体为第 11 层）上进行知识蒸馏，以使其相对位置与教师模型中的位置相近。按照 [21]中的建议，码本数量设为 16。使用 MVQ 生成整个训练集的码本索引，并将其存储在磁盘上，以便训练时读取。

我们考虑两种完整格式化的端到端自动语音识别模型：**Whisper-large** 模型[2] 和我们的 **Zipformer-KD** 模型。所提出的模型使用 icefall[3] 框架开发。我们采用 icefall LibriSpeech 配方[4] 中的默认配置来设置 Zipformer。输入声学特征采用 80 维梅尔滤波器组特征。我们使用 SentencePiece 单元语言模型分词器 [31]，其词汇表包含 500 个词元，反分词后的输出包括大写和小写字符，以及逗号、句号和问号。默认情况下，我们将公式（5）中的 α 设为 0.1。模型在四块 32GB NVIDIA Tesla V100 GPU 上训练 30 个训练轮次，最终模型通过对最后十个最佳检查点进行平均得到。解码时采用大小为 4 的束搜索。

C. 训练外部语言模型

在解码过程中，可以使用外部语言模型（LM）来提升端到端自动语音识别系统的性能。在本研究中，我们使用的外部语言模型由一个包含 1024 个单元的三层长短期记忆（LSTM）网络组成。该模型与转导器模型使用相同的词汇表，并在 LibriSpeech-PC 转录文本上训练了 10 个训练轮次。解码时使用最后一个训练轮次的检查点。转导器模型和语言模型的输出通过对数线性插值进行融合，这种方法也称为浅层融合 [32]。

D. 评估指标

为了衡量带有标点符号和大小写的预测文本的准确性，我们采用两类评估指标。第一类是基于 F1 的指标，包括精确率、召回率和 F1 分数，这些指标常用于标点符号和大小写研究 [33]。然而，在自动语音识别应用中，为所有样本计算 F1 分数并不可行，因为模型的输出可能包含自动语音识别错误。在本研究中，我们仅针对词错误率为零的样本计算基于 F1 的指标。第二类是基于词错误率的指标，包括不带标点符号且不区分大小写的经典词错误率（WER）、区分大小写但不带标点符号的词错误率（WER C）、同时包含标点符号和大小写的词错误率（WER PC），以及标点符号错误率（音素错误率，PER）[15]。

IV. 实验结果

A. 与最先进模型的比较

基于 F1 的指标结果如表 1 所示。在包含 2417 个样本的 test-clean 子集中，端到端 Conformer [15] 有 66% 的样本实现了零词错误率，而 Zipformer-KD 将这一比例提升至 76%，展现出更强的无错误输出能力。在标点符号方面，端到端 Conformer 的 F1 分数为 87.40，Zipformer-KD 则提升至 88.57，表明其标点

[1] https://dl.fbaipublicfiles.com/hubert/ hubert_xtralarge_ll60k_finetune_ls960.pt

[2] https://huggingface.co/openai/whisper-large [3] https://github.com/k2-fsa/icefall [4] https://github.com/k2-fsa/icefall/blob/master / egs/librispeech/ASR/zipformer/zipformer.py

表 1。 词错误率为零的样本数量，以及这些样本在标点和大小写方面的 F1 分数、精确率和召回率。

模型	子集（样本总数）	词错误率=0 样本	标点符号 F1	精确率	召回率	大小写 F1	精确率	召回率
端到端 Conformer [15]	test-clean (2417)	1605(66%)	87.40	86.11	88.90	94.85	95.53	94.20
Zipformer-KD（我们的方法）		1849(76%)	88.57	88.10	89.05	95.76	95.80	95.73
端到端 Conformer [15]	test-other (2856)	1439(50%)	88.27	86.87	89.93	95.50	96.37	94.67
Zipformer-KD（我们的方法）		1704（59%）	89.03	88.23	89.84	96.13	96.73	95.53

表 2. LibriSpeech-PC test-clean 和 test-other 集上的基于词错误率的指标结果。

模型	Num. 参数。	test-clean（%） PER	词错误率 WER C		WER PC	test-other（%） PER	词错误率 WER C		WER PC
Whisper-base [15]	74M	36.97	4.50	6.82	11.23	40.27	11.23	13.57	17.82
端到端 Conformer [15]	115M	29.48	2.83	4.72	8.11	27.37	5.65	7.57	10.48
Whisper-large	1550 M	30.52	3.02	5.15	8.60	30.55	5.99	8.02	11.29
Zipformer-KD（我们的方法）	65M	29.74	2.20	4.20	7.58	28.18	5.23	7.27	10.21
端到端 Conformer+语言模型 [15]	115M	29.24	2.22	4.05	7.66	27.67	4.67	6.55	9.80
Zipformer-KD+语言模型（我们的方法）	65+25M	**27.24**	**1.87**	**3.59**	**6.65**	**26.55**	**4.56**	**6.28**	**9.05**

预测性能更优。在大小写方面，Zipformer-KD 的 F1 分数为 95.76，高于端到端 Conformer 的 94.85，表现出持续的性能提升。 在包含 2856 个样本的 test-other 子集中，Zipformer-KD 的表现更出色，有 59% 的样本实现了零词错误率，而端到端 Conformer 为 50%。此外，Zipformer-KD 的标点 F1 分数升至 89.03，超过端到端 Conformer 的 88.27。这些结果表明，Zipformer-KD 在降低错误率和提升准确率方面始终略胜一筹，因此在各种声学环境中提升格式化文本准确率方面可能更有效。为确保模型之间的比较公平，必须仅考虑零词错误率样本子集的交集。尽管如此，端到端 Conformer [15] 并未提供其零词错误率样本的明确数据。因此，我们只能分别基于各模型对应的零词错误率样本子集来呈现结果。

表 2 展示了 LibriSpeech-PC 的 test-clean 和 test-other 子集上基于词错误率的指标结果。在不使用外部语言模型的情况下，拥有 6500 万参数的 Zipformer-KD 在 WER PC 上表现出竞争力，在 test-clean 子集和 test-other 子集上的得分分别为 7.58% 和 10.21%。这表明它在准确处理标点符号和大小写等文本细节方面具有出色的能力并且优于两个 Whisper 变体。在音素错误率方面，Zipformer-KD 在 test-clean 子集和 test-other 子集上的得分分别为 29.74% 和 28.18%，与端到端 Conformer [15] 等更大型模型相比也具有竞争力。值得注意的是，尽管未使用语言模型，Zipformer-KD 在 test-clean 子集上的词错误率和 WER PC 均低于端到端 Conformer+LM [15] 。集成外部语言模型后，Zipformer-KD+LM 模型的性能进一步提升，整体表现最佳。它在 test-clean 子集和 test-other 子集上的 WER PC 得分分别为 6.65% 和 9.05%，与此前最先进模型相比，相对提升显著，分别约为 13% 和 8%。实现这一提升的同时，模型参数规模更小。此外，模型在 test-clean 子集和 test-other 子集上的音素错误率分别降至 27.24% 和 26.55%，进一步彰显了其准确性和效率。

B. 消融研究

我们针对非流式模式和流式模式下的知识蒸馏（KD）进行了消融研究。如表 3 所示，在非流式模式下，不使用 KD 会导致所有评估指标均有所下降：在 test-clean 和 test-other 子集上，WER PC 分别增加了 0.24% 和 0.49%，PER 分别增加了 0.49% 和 1.33%

表 3。 在非流式和流式模式下，对 LibriSpeech-PC 数据集进行知识蒸馏的消融研究。报告了音素错误率、词错误率、WER PC（括号内）和训练成本。训练成本以 GPU 小时总数衡量。

模型	test-clean（%） PER	词错误率（ PC）	test-other（%） PER	词错误率（ PC）	GPU 小时
非流式模型					
不使用知识蒸馏的 Zipformer	30.23	2.42 (7.82)	29.51	5.55 (10.70)	152
使用知识蒸馏的 Zipformer	29.74	2.20 (7.58)	28.18	5.23 (10.21)	164
流式模型					
未使用知识蒸馏的 Zipformer	30.01	3.13 (9.00)	29.05	7.85 (12.90)	188
Zipformer（采用知识蒸馏）	28.74	2.85 (8.53)	28.81	7.55 (12.75)	200

表 4. 超参数调优的消融研究 α。

超参数	WER PC (%)	
	test-clean	test-other
$\alpha = 0.05$	7.62	10.21
$\alpha = 0.1$ (**最终**)	**7.58**	**10.21**
$\alpha = 0.2$	7.67	10.22

。相反，采用 KD 后，这些评估指标均显著改善，表明了使用 KD 的优势。在流式模式下，KD 的影响更加显著：在 test-clean 和 test-other 子集上，WER PC 分别降低了 0.47% 和 0.15%，PER 分别降低了 1.27% 和 0.24%。这些结果表明，KD 能有效且稳健地提升全格式自动语音识别的性能。由于教师模型是基于 LibriSpeech 数据集微调的 HuBERT 模型，而该数据集中的转录文本均为规范化文本，因此引入 KD 可以引导学生模型的中间表示更有利于准确预测规范化文本。值得注意的是，输出中的标点符号对先前预测文本的正确性非常敏感。在 RNN-T 解码器中，后续词元的预测依赖于先前输出的词元；因此，先前预测文本的准确性能够显著提升标点预测的性能。此外，大写词元通常与标点符号同时出现；例如，句号后的词总是大写的。因此，标点预测的改善也有助于更准确地预测大写词元。由此可见，采用 KD 能显著提升全格式自动语音识别的性能。

在训练成本方面，引入知识蒸馏仅导致 GPU 小时总数略有增加——非流式模式和流式模式均增加了 12 小时。由于教师标签（即码本索引）在训练前便已准备就绪，因此无需在训练过程中即时生成教师标签这一计算密集型操作。因此，有限的额外开销主要来自从本地存储读取码本索引，以及计算公式 (4) 中所述的知识蒸馏损失所需的额外步骤。

表 4 展示了对公式 (5) 中超参数 α 的调优结果。我们在三个不同取值 {0.05, 0.1, 0.2} 上调节蒸馏损失权重 α，同时保持所有其他训练参数不变。根据表 4，在 α 设置为 0.1 时，test-clean 和 test-other 子集上的性能均达到最优。

C. 示例输出

表 5 展示了使用知识蒸馏（KD）和不使用知识蒸馏时的模型输出示例。两个模型在预测标点符号和大小写方面均表现出色。然而，未使用 KD 训练的模型难以准确识别低频词，而采用 KD 后则能够正确识别这些低频词。

V. 结论

在本研究中，我们提出了一种增强型全格式端到端自动语音识别模型，通过多码本向量量化进行知识蒸馏。我们构建了一个传感器系统，采用高性能的 Zipformer 作为声学编码器，并使用格式化文本进行训练。我们将自监督预训练教师模型的中间层嵌入压缩为码本索引（CI）。学生模型，尤其是声学编码器，经过训练以预测其码本索引。LibriSpeech-PC 数据集上的实验结果表明，所提出的模型性能更优。此外，消融研究验证了知识蒸馏在提升全格式自动语音识别性能方面的有效性。

表 5. 有知识蒸馏和无知识蒸馏时模型输出的示例。

Gold	谁是 Humpy Dumpy？问道，那老鼠。
未使用知识蒸馏的 Zipformer	谁是UncyDumpy？问道 老鼠。
使用知识蒸馏的 Zipformer	谁是Hum py Dumpy？问道 老鼠。

参考文献

[1] S. Kim, 《采用逐层多头注意力的深度循环神经网络与标点恢复》，载于*Proc. ICASSP*，2019年，第7280-7284页。[2] Ł. Augustyniak, P. Szymański, M. Morzy, P. Żelasko, A. Szymczak, J. Mizgajski, Y. Carmiel, N. Dehak, 《自发对话中的标点预测：能否通过改造词嵌入来减少自动语音识别错误？》，载于*Proc. INTERSPEECH*，2020年，第4906-4910页。[3] M. Sunkara, S. Ronanki, D. Bekal, S. Bodapati, K. Kirchhoff, 《用于对话语音标点预测的多模态半监督学习框架》，载于*Proc. INTERSPEECH*，2020年，第4911-4915页。[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, 《注意力就是一切》，载于*NeurIPS*，2017年，第5998-6008页。[5] R. Pappagari, P. Żelasko, A. Mikołajczyk, P. Pęzik, and N. Dehak, 《低资源场景下对话语音真实大小写恢复与标点符号的联合预测》，载于*Proc. ASRU*，2021年，第1185- 1191页。[6] M. Hentschel, E. Tsunoo, and T. Okuda, 《通过多任务学习和知识蒸馏提升标点恢复的鲁棒性与速度》，载于*Proc. ICASSP*，2021年，第7773-7777页。[7] J. Švec, J. Lehečka, L. Šmídl and P. Ircing, 《基于 Transformer 的自动标点预测与自动语音识别输出单词大小写重建》，载于文本、语音与对话国际会议，2021年，第86-94页。[8] D. Caseiro, P. Rondon, and P. Aleksic, 《使用大小写掩码的混合大小写上下文自动语音识别》，载于*Proc. INTERSPEECH*，2020年，第686- 690页。[9] M. Mimura, S. Sakai, and T. Kawahara, 《面向议会会议的端到端语音到清洁转录文本模型》，载于亚太信号与信息处理协会年度峰会暨会议，2021年，第465-470页 [10] Y. Guan, 《采用自动插入标点的端到端自动语音识别系统》，*arXiv*预印本 *arXiv:2012.02012*，2020年[11] J. Nozaki, T. Kawahara, K. Ishizuka, T. Hashimoto, 《端到端语音到带标点文本识别》，载于*Proc. INTERSPEECH*，2022年，第1811-1815页。[12] Z. Zhou, T. Tan, and Y. Qian, 《用于流式端侧语音识别的标点预测》，载于*Proc. ICASSP*，2022年，第7277- 7281页。[13] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, 《通过大规模弱监督实现鲁棒语音识别》，载于国际机器学习会议，2023年，第28492-28518页。[14] P. K. O'Neill, V. Lavrukhin, S. Majumdar, V. Noroozi, Y. Zhang, O. Kuchaiev, J. Balam, Y. Dovzhenko, K. Freyberg, M. D. Shulman, B. Ginsburg, S. Watanabe, G. Kucsko, 《SPGISpeech：5,000小时的

金融音频转录文本与全格式端到端语音识别》，载于*Proc. INTERSPEECH*，2021年，第1434-1438页。[15] A. Meister, M. Novikov, N. Karpov, E. Bakhturina, V. Lavrukhin, and B. Ginsburg, 《LibriSpeech-PC：端到端自动语音识别模型标点和大小写能力评估基准》，载于*Proc. ASRU*，2023年，第1-7页。[16] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, R. Pang, 《Conformer：用于语音识别的卷积增强型 Transformer》，载于*Proc. INTERSPEECH*，2020年，第5036-5040页。[17] X. Yang, W. Kang, Z. Yao, Y. Yang, L. Guo, F. Kuang, L. Lin, and D. Povey, 《用于可控风格上下文自动语音识别的 PromptASR》，载于*Proc. ICASSP*，2024年，第10536-10540页。[18] S. Ling, Y. Hu, S. Qian, G. Ye, Y. Qian, Y. Gong, E. Lin, and M. Zeng, 《利用语音调整大语言模型以实现全格式端到端语音识别》，载于*Proc. ICASSP*，2024年，第11046-11050页。[19] A. Graves, 《采用循环神经网络的序列转导》，*arXiv*预印本 *arXiv:1211.3711*，2012年。[20] Z. Yao, L. Guo, X. Yang, W. Kang, F. Kuang, Y. Yang, Z. Jin, L. Lin, and D. Povey, 《Zipformer：更快、更出色的自动语音识别编码器》，载于第十二届国际学习表征会议，2023年。[21] L. Guo, X. Yang, Q. Wang, Y. Kong, Z. Yao, F. Cui, F. Kuang et al., 《预测用于知识蒸馏的多码本向量量化索引》，载于*Proc. ICASSP*，2023年，第1-5页。[22] K. Zhao, H. D. Nguyen, A. Jain, N. Susanj, A. Mouchtaris, L. Gupta, and M. Zhao, 《通过模块替换进行知识蒸馏，用于基于循环神经网络转导器的自动语音识别》，载于*Proc. INTERSPEECH*，2022年，第4436-4440页。[23] S. Tian, K. Deng, Z. Li, L. Ye, G. Cheng, T. Li, and Y. Yan, 《通过一致的声学表征学习实现基于连接时序分类的语音识别知识蒸馏》，载于*Proc. INTERSPEECH*，2022年，第2633-2637页。

[24] J. Lee, 和 H. Seo, 《面向高效语音识别的仅解码器大语言模型在线知识蒸馏》，载于 *INTERSPEECH*论文集，2024年，第2890-2894页。[25] X. Yang、Q. Li 和 P. C. Woodland, 《利用大型自监督预训练模型蒸馏神经转导器》，载于 *ICASSP*论文集，2022年，第8527-8531页。[26] S. Panchapagesan、D. S. Park、C.-C. Chiu、Y. Shangguan、Q. Liang 和 A. Gruenstein, 《面向RNN转导器模型的高效知识蒸馏》，载于 *ICASSP*论文集，2021年，第5639-5643页。[27] V. Manohar、P. Ghahremani、D. Povey 和 S. Khudanpur, 《一种用于序列训练自动语音识别模型无监督领域自适应的教师-学生学习方法》，载于 *2018*年*IEEE*口语语言技术研讨会，2018年，第250-257页。[28] R. V. Swaminathan、B. King、G. P. Strimel、J. Droppo、A. Mouchtaris, 《CoDERT：通过协同学习蒸馏用于转导器语音识别的编码器表示》，载于 *INTERSPEECH*论文集，2021年，第4543-4547页。[29] W.-N. Hsu、B. Bolte、Y.-H. H. Tsai、K. Lakhotia、R. Salakhutdinov 和 A. Mohamed, 《HuBERT：通过隐藏单元掩码预测进行自监督语音表征学习》，*IEEE/ACM*音频、语音与语言处理汇刊，第29卷，第3451-3460页，2021年。[30] V. Panayotov、G. Chen、D. Povey 和 S. Khudanpur, 《LibriSpeech：基于公有领域有声读物的ASR语料库》，载于 *ICASSP*论文集，2015年，第5206-5210页。[31] T. Kudo, 《子词正则化：利用多个子词候选改进神经网络翻译模型》，载于 计算语言学协会第*56*届年会论文集，第1卷，第66-75页，2018年。[32] D. Bahdanau、J. Chorowski、D. Serdyuk、P. Brakel 和 Y. Bengio, 《端到端注意力机制大词汇量语音识别》，载于 *ICASSP*论文集，2016年，第4945-4949页。[33] V. Păiş 和 D. Tufiş, 《大小写与标点恢复：综述》，《人工智能评论》，第55卷，第1681-1722页，2022年。