



中国科学院大学

University of Chinese Academy of Sciences

博士学位论文

基于语言知识迁移的端到端语音识别方法研究

作者姓名: _____ 白 焯 _____

指导教师: _____ 陶建华 研究员 _____

_____ 中国科学院自动化研究所 _____

学位类别: _____ 工学博士 _____

学科专业: _____ 模式识别与智能系统 _____

培养单位: _____ 中国科学院自动化研究所 _____

2021 年 6 月

Transferring Text-Only Knowledge to
End-to-End Speech Recognition

A dissertation submitted to the
University of Chinese Academy of Sciences
in partial fulfillment of the requirement
for the degree of
Doctor of Philosophy
in Pattern Recognition and Intelligent Systems

By

BAI Ye

Supervisor: Professor TAO Jianhua

Institute of Automation, Chinese Academy of Sciences

June, 2021

中国科学院大学 学位论文原创性声明

本人郑重声明：所呈交的学位论文是本人在导师的指导下独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明或致谢。本人完全意识到本声明的法律结果由本人承担。

作者签名：

日 期：

中国科学院大学 学位论文授权使用声明

本人完全了解并同意遵守中国科学院大学有关保存和使用学位论文的规定，即中国科学院大学有权保留送交学位论文的副本，允许该论文被查阅，可以按照学术研究公开原则和保护知识产权的原则公布该论文的全部或部分內容，可以采用影印、缩印或其他复制手段保存、汇编本学位论文。

涉密及延迟公开的学位论文在解密或延迟期后适用本声明。

作者签名：

日 期：

导师签名：

日 期：

摘要

大规模无标注文本语料中蕴含着丰富的语言知识。提炼出无标注文本语料中的知识来提升分类、匹配、序列标注等语言信息处理任务的性能已经被证实是一种行之有效的手段。然而，对采用神经网络一体化建模的语音识别、机器翻译等文本生成模型，无标注文本语料的优势并没有完全显现出来。这是由于实用的文本生成模型往往是条件化的(如根据语音、图像等生成文本)，需要成对数据训练，所以其难以直接利用无标注纯文本数据。已有的一些利用方法存在识别阶段增加额外模型导致开销大、无法利用已训练好的语言模型导致不灵活等问题。如何有效地令基于神经网络一体化建模的文本生成模型利用大规模无标注文本语料中的语言知识，同时避免开销大和不灵活这两个问题，还缺乏深入地研究。

本文从“如何利用纯文本数据提升端到端语音识别性能”这一具体的实际问题出发，以迁移学习为主线方法，面向从大规模无标注文本语料中迁移知识到端到端语音识别模型，在“上文语言知识迁移”、“全局上下文语言知识迁移”、“跨模态全局上下文语言知识迁移”三个递进的层面上，完成了四项创新工作。

1. 提出一种文本知识利用方法。针对已有方法存在识别阶段增加额外模型导致开销大、无法利用已训练好的语言模型导致不灵活的问题，本文提出了一种基于老师-学生学习的文本知识利用方法 LST，利用大规模无标注文本语料中的语言知识，来提升端到端语音识别的性能：首先利用语言模型将大规模纯文本中的语言知识表示起来，然后利用老师-学生学习将此语言知识迁移到端到端语音识别系统中。与其它方法相比，该方法不增加预测阶段的计算代价，比较高效；同时，该方法可以利用其它开放获取的已经训练好的语言模型而不需要自行训练，方便灵活。本文还分析比较了该方法与另一种典型的文本知识利用方法浅融合，发现平滑模型估计的分数空间是这两种方法提升识别性能的重要性质。同时，该方法不仅可以应用在语音识别任务，还可以应用在其它所有条件化的文本生成任务中。

2. 提出一种全局上下文语言模型。针对端到端编码器-解码器模型没有利用文本中上下文知识的问题，本文提出一种全局上下文语言模型“因果完形填空器”，然后利用 LST 方法将此全局上下文语言知识迁移到端到端编码器-解码器模型

中，使得编码器-解码器模型也可以利用全局上下文语言知识。相比其它的利用双向语言知识的方法，该方法不增加识别阶段复杂性，还可以灵活地利用无标注纯文本。

3. 证实利用语音中包含的语言知识而不进行显式语言建模也可以有效进行语音识别。针对已有端到端编码器-解码器模型的自回归模式束搜索阶段耗时较大的问题，基于观察到的语音与文本的语言知识同构现象，本文提出一种端到端非自回归语音识别模型 LASO。该方法没有显式地自回归语言建模，所以可以并行地实现同时预测所有的词。实验表明，所提模型在两种规模的公开中文语音数据集上都可以表现出与自回归模型可比的性能，但处理速度是自回归模型的近 50 倍。这些结果表明，不进行显式地自回归语言建模，而是利用语音中的语言知识，也可以进行高效的语音识别。

4. 提出一种跨模态全局语言知识迁移方法，有效提升了单模态语音识别模型性能。根据文本与语音的语言知识同构性，本文提出将大规模预训练语言模型中的语言知识跨模态地迁移到非自回归语音识别模型 LASO 中。实验证明，所提方法可以提升纯语音模态建模的端到端语音识别模型的效果。同时结果表明，利用不同模态的语言知识同构性进行知识迁移，可以有效地提升不同模态模型的性能。

关键词：端到端语音识别，语言知识迁移，全局上下文语言模型，非自回归语音识别，跨模态知识迁移

Abstract

Large-scale text-only data contains rich knowledge. It has been confirmed that distilling knowledge in text-only data can improve performance in many natural language processing tasks, such as classification, matching, and sequence labeling. However, the advantages of text-only data have not been shown in deep learning based end-to-end text generation models. Because practical text generation is conditional, these models need paired data to train. It is non-trivial to directly use text-only data to train these models. Previous work will add extra modules during recognition or cannot use pre-trained language models. Thus, it is worth investigating methods, which are flexible to use pre-trained language models, to use text-only data for improving end-to-end text generation models without extra computation during inference.

This thesis focuses on a practical problem: how to use text-only data to improve end-to-end speech recognition. Taking transfer learning, we would like to transfer knowledge from text-only data to end-to-end speech recognition. We discuss three aspects: transferring knowledge of left context, transferring knowledge of whole-sentence context, and cross-modal knowledge transfer. The four contributions are as follows.

1. *Propose a method to using knowledge in text-only data.* We propose a teacher-student learning based method called LST. It uses knowledge in text-only data to improve end-to-end speech recognition. It first uses language models to represent knowledge in text-only data. Then, the knowledge is transferred to speech recognition models. Compared with other methods, LST is more efficient since it does not add computation during the test stage. In addition, it is flexible for LST to use language models pre-trained by others. This thesis also analyzes and compares LST and another method shallow fusion. We found that smoothing the score space of a model is an important factor to improve performance. The proposed LST can not only be used for speech recognition but also all the other conditional text generation.

2. *Propose a whole-sentence language model.* The encoder-decoder model does not use the "future" context during text generation. To address this issue, this thesis

proposes a whole-sentence language model called causal cloze completer. We use the proposed LST to transfer the whole-sentence knowledge to end-to-end speech recognition. Compared with other methods which use bidirectional information in a sentence, the proposed method does not add extra computation at the test stage. And it can use text-only data flexibly.

3. *Confirm that the language semantics in speech can be used to speech recognition without explicit language modeling.* Based on the observed isomorphism between speech and text, this thesis proposes an end-to-end non-autoregressive speech recognition model called LASO. LASO is non-autoregressive so that it can generate all tokens in parallel. The experiments show that the proposed LASO achieves comparable performance on two public Chinese speech datasets. The processing speedup is about 50 times, compared with the autoregressive baseline. These results show that it is feasible to speech recognition without explicit language modeling.

4. *Propose a cross-modal knowledge transfer method to improve the performance of a unimodal speech recognition model.* Based on the isomorphism between speech and text, this thesis proposes to transfer knowledge from large-scale pre-trained language models to the proposed non-autoregressive model. The experiments show that the proposed method can improve the performance of the unimodal end-to-end speech recognition model. The results reveal that using the isomorphism between the speech and the text and transferring the knowledge from the text-based model can improve the performance of the speech model.

Keywords: end-to-end speech recognition, language knowledge transfer, whole-sentence language modeling, non-autoregressive speech recognition, cross-modal knowledge transfer

目 录

| | |
|---------------------------|----|
| 第 1 章 绪论 | 1 |
| 1.1 引言 | 1 |
| 1.2 研究背景与意义 | 2 |
| 1.3 研究思路与本文工作 | 6 |
| 第 2 章 背景介绍及相关工作 | 9 |
| 2.1 引言 | 9 |
| 2.2 基于注意力机制的编码器-解码器模型 | 9 |
| 2.2.1 问题的表示 | 9 |
| 2.2.2 模型结构 | 10 |
| 2.2.3 注意力机制 | 10 |
| 2.2.4 训练与预测 | 12 |
| 2.2.5 基于图结构的前馈神经网络模型 | 16 |
| 2.3 端到端语音识别系统中纯文本知识利用研究现状 | 23 |
| 2.3.1 基于融合的方法 | 23 |
| 2.3.2 基于合成数据的方法 | 24 |
| 2.3.3 有待研究的问题 | 25 |
| 第 3 章 上文语言知识迁移 | 27 |
| 3.1 引言 | 27 |
| 3.2 老师-学生学习训练方法 | 28 |
| 3.3 本文方法: LST 训练 | 29 |
| 3.3.1 语言模型 | 29 |
| 3.3.2 语言模型作为老师模型 | 30 |
| 3.4 实验 | 32 |
| 3.4.1 实验数据 | 32 |
| 3.4.2 实验设置 | 35 |
| 3.4.3 实验结果 | 36 |
| 3.4.4 分析与讨论 | 42 |
| 3.5 小结 | 45 |

| | |
|-----------------------------------|----|
| 第 4 章 全局上下文语言知识迁移 | 47 |
| 4.1 引言 | 47 |
| 4.2 基于完形填空的上下文全局语言建模 | 48 |
| 4.2.1 完形填空 | 48 |
| 4.2.2 因果完形填空器 | 49 |
| 4.2.3 将全局上下文知识迁移到端到端语音识别 | 51 |
| 4.3 相关工作 | 51 |
| 4.4 实验 | 52 |
| 4.4.1 实验数据 | 52 |
| 4.4.2 实验设置 | 52 |
| 4.4.3 实验结果 | 53 |
| 4.5 小结 | 58 |
| 第 5 章 跨模态全局上下文语言知识迁移 | 59 |
| 5.1 引言 | 59 |
| 5.2 语音识别中的语言语义 | 60 |
| 5.3 基于图结构前馈神经网络的非自回归语音识别模型 | 62 |
| 5.3.1 语音识别作为逐位置的分类问题 | 62 |
| 5.3.2 模型 | 62 |
| 5.3.3 训练 | 64 |
| 5.3.4 识别 | 65 |
| 5.4 跨模态语言知识迁移 | 65 |
| 5.5 相关工作 | 67 |
| 5.6 实验 | 68 |
| 5.6.1 实验数据 | 68 |
| 5.6.2 实验设置 | 69 |
| 5.6.3 评价准则 | 70 |
| 5.6.4 实验结果 | 71 |
| 5.6.5 分析与讨论 | 76 |
| 5.7 小结 | 83 |
| 第 6 章 总结与展望 | 85 |
| 6.1 本文工作总结 | 85 |
| 6.2 未来工作展望 | 86 |
| 附录 A 梅尔滤波器组特征的提取 | 87 |

| | |
|-------------------------------|-----|
| 附录 B 端到端语音识别模型的一些训练技巧 | 89 |
| B.1 谱增强 | 89 |
| B.2 模型平均 | 90 |
| B.3 最小词错误率训练 | 90 |
| 参考文献 | 93 |
| 作者简历及攻读学位期间发表的学术论文与研究成果 | 107 |
| 致谢 | 111 |

图形列表

| | |
|---|----|
| 1.1 语音识别发展的一个简要历史 | 3 |
| 1.2 第二代技术范式和第三代技术范式的对比。 | 4 |
| 1.3 本文研究思路 | 6 |
| 2.1 端到端语音识别示意图 | 9 |
| 2.2 基于注意力机制的编码器-解码器模型 | 10 |
| 2.3 注意力机制 | 12 |
| 2.4 教师强制训练 | 13 |
| 2.5 一步束搜索的示意图。这里假设束的宽度为 3。左边表示旧的束，右边表示扩展出的新束，蓝色的圆点表示旧累积分数，黄色圆点表示根据旧束中状态，用语音识别模型计算出的每一个词分数，绿色圆点表示新的候选的累积分数。颜色深浅表示分数大小。 | 14 |
| 2.6 基于图结构的前馈神经网络。(a) 全连接图，(b) 自注意力的输入和输出，每一个节点为一个节点表示向量。 | 16 |
| 2.7 语言模型的图结构表示。该图表示了序列的一部分前缀。(a) 语言模型的图结构表示，(b) 语言模型的消息传播模式。 | 17 |
| 2.8 transformer 结构示意图 | 19 |
| 2.9 自注意力机制的位置编码。(a) 无位置编码时，交换两个字的融合结果比较；(b) 带位置编码时，交换两个字的融合结果比较。 | 20 |
| 2.10 基于融合的方法 | 23 |
| 2.11 基于合成数据的方法 | 24 |
| 3.1 本文提出的基于老师-学生学习的 LST 方法 | 27 |
| 3.2 循环神经网络语言模型 | 29 |
| 3.3 以循环神经网络为老师模型的 LST 方法 | 30 |
| 3.4 软标签与硬标签的比较。示意图中假设语言模型输入上文“不要着”。相比于硬标签，软标签可以提供不同词概率相对大小的信息。 | 31 |
| 3.5 识别结果分数的直方图。前四个图模型输出的是概率。对于后两个图中的浅融合，模型输出的分数不可以看做概率。 | 44 |
| 4.1 “完形填空”的示意图 | 48 |
| 4.2 因果完形填空器的结构示意图 | 49 |
| 4.3 因果完形填空器的消息流 | 50 |
| 5.1 语音中蕴含的语言知识示意图 | 59 |

| | |
|--|----|
| 5.2 语言语义的无向图表示 | 61 |
| 5.3 LASO 的模型结构 | 63 |
| 5.4 跨模态语言知识迁移 | 66 |
| 5.5 LASO 编码器最后一层自注意力分数的可视化结果 | 78 |
| 5.6 LASO 解码器最后一层自注意力分数的可视化结果 | 79 |
| 5.7 LASO 位置相关总结器最后一层注意力分数的可视化结果 | 80 |
| 5.8 训练集长度分布直方图和测试集句子长度与字错误率关系的散点图。 散点的面积大小和字错误率数值大小成正关系。(a) AISHELL-1 上的 结果, (b) AISHELL-2 上的结果。 | 82 |
| A.1 梅尔尺度三角滤波器组示意图 | 87 |
| B.1 谱增强技术示意图 | 89 |

表格列表

| | |
|---------------------------------------|----|
| 3.1 语音数据情况 | 33 |
| 3.2 文本数据情况 | 34 |
| 3.3 不同数据训练的 3 元语法语言模型在开发集标注文本上的困惑度 .. | 35 |
| 3.4 AISHELL-1: 困惑度测试 | 37 |
| 3.5 AISHELL-1: 超参数选择 | 38 |
| 3.6 AISHELL-1: 测试集上的字错误率 | 39 |
| 3.7 AISHELL-1: LST 与浅融合的组合 | 40 |
| 3.8 AISHELL-2: 困惑度测试 | 41 |
| 3.9 AISHELL-2: 测试集上的字错误率 | 41 |
| 3.10 AISHELL-2: LST 与浅融合的组合 | 42 |
| 4.1 AISHELL-1: 完形填空正确率 | 54 |
| 4.2 AISHELL-1: 超参数选择 | 54 |
| 4.3 AISHELL-1: 测试集上的字错误率 | 55 |
| 4.4 AISHELL-2: 完形填空正确率 | 57 |
| 4.5 AISHELL-2: 测试集上的字错误率 | 57 |
| 5.1 数据集长度信息统计 | 68 |
| 5.2 模型结构配置符号 | 69 |
| 5.3 AISHELL-1: 不同模型结构配置下的字错误率 | 72 |
| 5.4 AISHELL-1: 和基线模型比较 | 74 |
| 5.5 AISHELL-2: 与基线系统比较 | 75 |

符号列表

| 符号 | 含义 |
|--|----------------------------------|
| y | 特指一个符号 (汉字, 词, 英文单词, 字母等) 序列 |
| y_j | 特指符号序列中第 j 个符号 |
| $y_{<j}$ | 特指符号序列中第 j 个符号的前缀 (不包括 y_j) |
| i, I, λ | 字母和希腊字母表示一个标量, 其中大写字母表示总数 |
| $\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{h}$ | 小写粗体字母表示一个行向量 |
| $\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{H}$ | 大写字母表示一个矩阵 |
| $[\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3]$ | $[\cdot]$ 表示一组向量拼接成的矩阵 |
| \exp, \log | 正体字表示某些数学函数 |
| \mathbb{R}, \mathbb{D} | 黑板粗体表示集合 |

第 1 章 绪论

1.1 引言

深度神经网络的表示学习能力极大地提升了模式识别任务的水平 [1]。近年来,通过无监督学习任务¹获得的能提取好的表示²的深度学习模型在计算机视觉 [2–5],自然语言处理 [6–8],语音处理 [9, 10] 等任务中获得了巨大的成功。特别是在自然语言处理中,首先利用大规模无标注的文本语料预训练 (pre-training) 出强大的文本表示模型,再在数量较少的标注数据上进行微调 (fine-tuning) 的预训练-微调模式给文本信息处理带来了巨大的改变。这种方式有两大好处。首先,预训练模型可以有效利用超大规模数据,提取出语言知识提升下游任务性能:商用预训练模型可以利用百吉比特乃至太比特规模的互联网文本。其次,利用少量标注数据进行微调,预训练模型就可以获得优异的性能,大大降低了标注数据成本。所以,预训练-微调已经成为了语言信息处理的新范式 [8]。

然而,目前预训练-微调模式在分类和匹配任务(如文本分类、阅读理解、问答匹配等)中带来的巨大优势,在语音识别 (speech recognition)、机器翻译 (machine translation)、自动摘要 (automatic summarization)、图像描述 (image caption) 等文本生成 (text generation) 任务中并没有同期显现出来。这主要是因为实用的生成任务,往往是类似于语音识别、机器翻译、图像描述等条件化的文本生成任务。这就造成这些任务的模型往往需要输入成对数据 (paired data),例如语音识别中的语音-文本对,机器翻译中的源语言-目标语言对,看图说话中的图片-文本对等。这些成对数据本身就是标注数据,获取的成本较高。目前有一些工作尝试在这些文本生成任务的条件编码部分利用预训练模型。比如,在语音识别领域,利用无标注语音无监督地训练语音编码模型 [9, 10];在机器翻译领域,利用无标注文本无监督地训练源语言编码模型 [11]。然而,如何有效地在文本生成侧利用大规模无标注文本中提炼出的语言知识,还缺乏深入的研究。

¹这里的无监督指的是“不使用人类提供的标注”。近年来部分文献采用“自监督”来表示“标注从样本本身得来而非人工添加”这一情况。然而“自监督学习”这一名词还未定型。故本文使用“无监督”这一更为经典的名词。

²如何评判一个“表示”的好坏现在还未有定论。目前学界一般是利用下游任务的性能来衡量所提取表示的好坏。

本文针对“如何有效地将大规模无标注文本中提炼出的语言知识应用在文本生成任务”这一问题进行研究。具体来说，本文针对语音识别这一典型的文本生成任务，设计了一种基于迁移学习的语言知识集成方法，从“上文语言知识迁移”、“全局上下文语言知识迁移”、“跨模态全局上下文语言知识迁移”三个递进的层次讨论如何高效利用大规模无标注文本中提炼出的语言知识。作为一种一般性的方法，本文研究的方法同时也可以推广到所有文本生成任务中。

本章的后续部分组织如下。1.2 节介绍研究的背景，并引出本文研究的问题和意义。1.3 节介绍本文的研究思路和工作。

1.2 研究背景与意义

自动语音识别是指利用机器将语音转换成对应的文字。长久以来，制造可以与人交流的机器一直是人类的梦想。可以与人直接对话的机器也常常以一个幻想角色的身份出现在科幻电影中³。作为让机器理解人类语言的第一步，自动语音识别一直是科学家致力于攻克的一个问题。早在计算机还未流行起来的 1952 年，贝尔实验室的 Davis 等 [12] 就使用电路搭建了语音数字识别器。在之后的近 70 年里，语音识别系统一直沿着词汇量越来越大，准确率越来越高，识别速度越来越快，使用场景越来越复杂的方向不停演进 [13–20]。

从目前的视角看，自动语音识别经历了三代技术范式的发展。

- 第一代技术范式为模板匹配方法。采用对应于某些发音或文字的典型模式作为模板，测试时输入的语音与哪一个模板最匹配，就将其识别为对应的发音或文字 [12, 21–29]。这一时期的代表性技术是动态时间规整 (dynamic time warping, DTW)[29]。然而，基于模板匹配的技术存在词汇量小，难以识别连续语音，语音变化应对能力不强等问题，难以在真实的大词汇量连续语音识别场景中进行应用。

- 第二代技术范式为噪声信道 (noisy channel) 下的概率模型方法。在该范式下，语音识别被定义为一个噪声信道下的最大后验决策问题：给定声学特征序列，求对应概率最大的文本序列 [30]。声学特征和语言特征被声学模型和语言模型分别建模。隐马尔可夫模型 (hidden Markov models, HMMs) 被用来建模语音帧之间的动态转移以及声学特征和语音学单元（音素，上下文相关三音素，音节

³1968 年库布里克导演的著名科幻电影《2001：太空漫游》就有可以与人自由对话的智慧电脑 HAL。

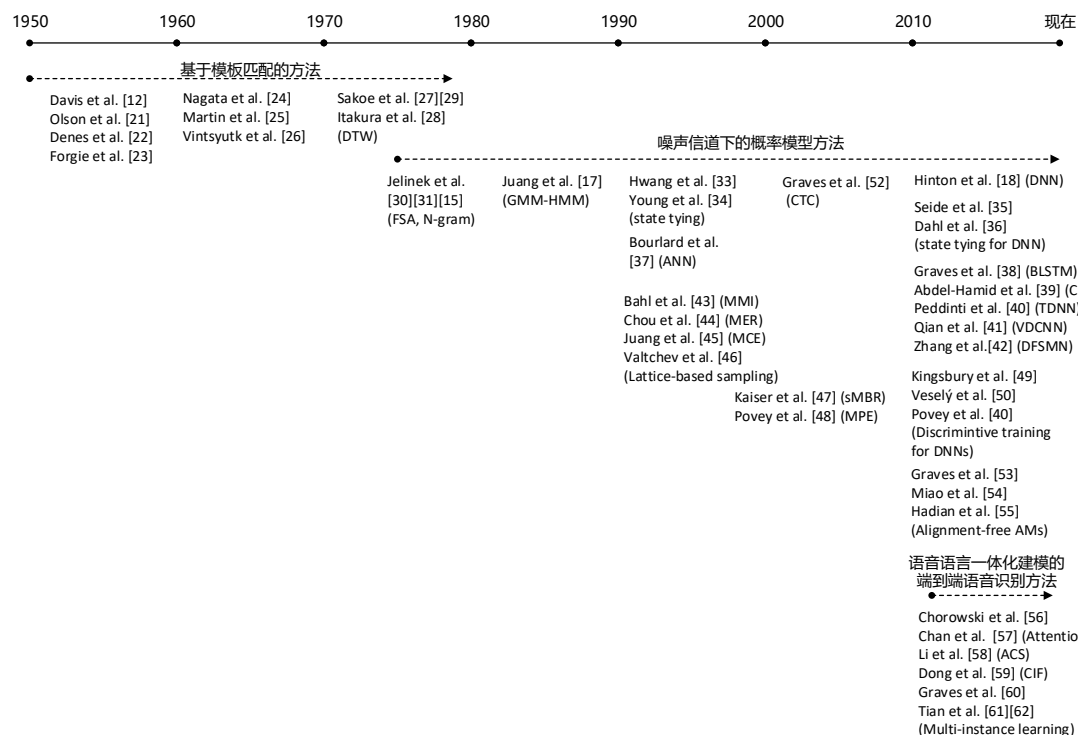


图 1.1 语音识别发展的一个简要历史

Figure 1.1 A brief history of speech recognition.

等) 之间的关系 [31, 32]。词之间的关系通过统计语言模型建模 [15]。在建模单元方面, 基于决策树的状态绑定技术解决了上下文相关三音素数据的稀疏问题, 使隐马尔可夫模型可以表示更细微的发音单元, 并显著地提升了语音识别的效果 [33–36]。这项技术已经成为了语音识别系统的标准技术。在隐马尔可夫模型观测概率建模方面, 主要分为基于高斯混合模型 (Gaussian mixture models, GMMs) 的生成式模型建模和基于人工神经网络 (artificial neural networks, ANNs) 的判别式模型建模。从 20 世纪 90 年代初期开始一直到 21 世纪初期, 高斯混合模型一直占据统治地位 [17]。人工神经网络在 1990 年被引入用来建模隐马尔可夫模型的观测概率分布 [37], 然而早期的神经网络并没有获得相比高斯混合模型更好的效果。进入到 21 世纪 10 年代, 随着数据量、计算资源、优化技巧等瓶颈问题被慢慢解决, 相比早期人工神经网络规模大很多的深度神经网络 (deep neural networks, DNNs) 被成功地用于语音识别领域 [18], 并引领了深度学习 (deep learning, DL) 的发展浪潮。结合优化改进的神经网络结构 [38–42] 和序列级区分性训练技术 (sequence-level discriminative training, SDT)[43–51], 语音识别系统的性能大大提升, 并可以成功地在实际场景和产品系统中使用。为了解决神经网络训练需要帧

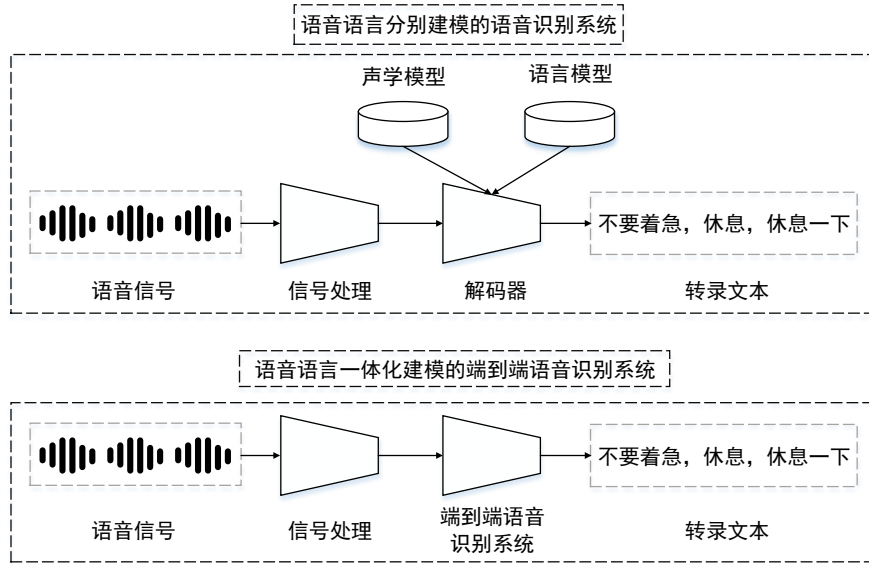


图 1.2 第二代技术范式和第三代技术范式的对比。

Figure 1.2 the second generation vs. the third generation

级别标注而导致训练流程复杂的问题，一系列序列级损失函数被提出来直接训练神经网络 [52–55]，大大简化了语音识别系统的构建流程。

- 第三代技术范式为语音语言一体化建模的端到端语音识别方法⁴。得益于误差可以传导到所有部件的特性，神经网络可以采用统一的优化准则优化所有部件，所以神经网络可以进行声学语言一体化端到端建模。目前来看，端到端语音识别系统可以归结为两类：基于软对齐机制的编码器-解码器模型 [56–59] 和基于多示例学习 (multi-instance learning, MIL) 神经网络转换器 (transducers) 模型 [60–62]。基于软对齐机制的编码器-解码器模型中，编码器编码声学特征序列，解码器编码文本特征，通过可学习的软对齐机制学习编码器和解码器之间的对齐关系，抽取最匹配的声学特征预测下一个词。基于多示例学习的神经网络转换器模型一般包括一个声学编码模块和文本编码模块，枚举出所有可能的语音-文本对齐路径进行优化，测试时可以自动组合出最匹配的对齐路径进行预测。相比于第二代技术范式声学语言分离建模，端到端语音识别避免了多模型可能存在的目标准则不一致导致的误差累积问题。同时，端到端语音识别不需要发音词典等专家知识，也不需要利用预先准备好的识别系统进行帧级别标注，整体构建流程简单，大大降低了开发语音识别系统的门槛。

⁴ “端到端”这个术语目前还未定型，一般有两种解释：1) 不需要一个预先准备好的语音识别系统进行帧级别标注，而直接训练神经网络；2) 语音语言一体化建模。本文采用第二种解释。

相比于第二代技术范式中语音语言分离建模的语音识别系统，第三代技术范式端到端语音识别主要具有三个特点。

1. 语音部分和语言部分一体化建模，采用统一的损失函数进行优化，避免了分开建模时各模型之间存在的不匹配以及整个系统的误差累积；

2. 建模单元灵活，无需专家知识构建发音词典，系统构建过程简单，无需复杂的构建流水线；

3. 整个语音识别系统都采用神经网络构成，相比静态解码网络，系统总体积大大减小。

正是由于这三个优点，端到端语音识别技术吸引了学术界和工业界的关注。

然而，一体化建模技术也引入了数据使用上的灵活性问题。具体来说，由于语音语言一体化建模，端到端模型一般利用语音-文本成对数据训练构建。所以相比于语音语言分离建模的语音识别系统可以通过直接训练语言模型来利用大规模纯文本数据，端到端语音识别模型并不能直接利用纯文本数据。

纯文本数据相比语音-文本成对数据成本低廉，容易获取⁵。更为重要的是，大规模纯文本数据中蕴含着丰富的语言知识⁶，可以极大地提升语言信息处理模型的性能。如 BERT [7] 利用了 13 吉比特纯文本进行训练，大幅度地提升机器阅读理解等几个自然语言理解任务的性能指标。后续，XLNet [63] 利用了大约 126 吉比特纯文本进行训练，进一步地提升了性能。而由 OpenAI 公司开发的 GPT-3[8] 则利用了 570 吉比特纯文本数据(原始数据规模为 45 太比特)，并达到了报道中令人惊异的“零样本学习”的能力，显示了利用超大规模纯文本数据训练的潜在价值，甚至有媒体认为“GPT-3 是通向通用智能的方向”⁷。虽然媒体的报道并不能说明科学问题，但也说明利用大规模文本数据的 GPT-3 显示出的性能带给大众的震撼。所以，大规模纯文本数据的利用具有科学价值和实用价值。

如果能像语音语言分别建模的模型一样，让端到端语音识别系统将纯文本数据可以利用起来，就可以降低端到端语音识别系统构建的数据成本，甚至进一步地提升语音识别的性能。特别是对于某些语音-文本成对数据难以获取的情形，如某些低资源语言等，纯文本数据的利用就更有意义。所以，研究如何利用纯文

⁵2021 年客服语音数据标注成本约为一小时 400 至 500 元人民币。

⁶这里的语言知识指的是统计意义上的词，字，或子词共现信息。

⁷Claypoole, Theodore (July 30, 2020). "New AI Tool GPT-3 Ascends to New Peaks, But Proves How Far We Still Need to Travel". The National Law Review.

本数据中的语言知识提升端到端语音识别性能的方法具有实际应用价值。

同时，由于语音和文本属于不同的模态，利用纯文本数据实际上涉及到跨模态知识利用的问题。对于复杂任务，人类往往不局限于单一模态，而是跨模态统筹处理所有可以利用的信息 [64]。赋予机器像人一样充分挖掘不同模态知识的能力，是令其处理现实环境复杂信息的重要步骤。所以研究如何利用纯文本数据中的语言知识提升端到端语音识别性能的方法具有理论价值和科学意义。

1.3 研究思路与本文工作

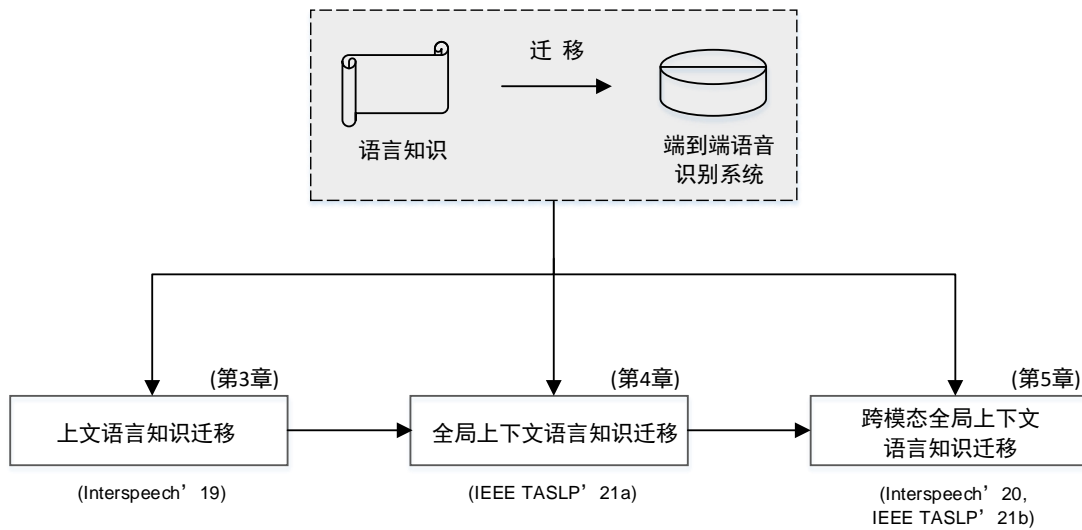


图 1.3 本文研究思路

Figure 1.3 the research route of this thesis

本文从“如何利用纯文本数据提升端到端语音识别性能”这一实际问题出发，基于迁移学习方法，面向从纯文本数据中迁移知识到端到端语音识别模型问题，从三个递进的方面进行探讨，如图 1.3 所示。

本文首先在第2章介绍端到端语音识别的背景，比较其它端到端语音识别系统利用纯文本知识的方法，引出本文的研究路线和方法，然后详细地探讨这三个层次的问题。

第3章介绍所提的语言知识迁移方法 LST：以神经网络语言模型将文本中语言知识表示起来，再利用老师-学生学习迁移到端到端语音识别系统。在该章中，我们比较了两种自回归的神经网络语言模型作为老师模型时的效果——循环神经网络语言模型和基于图结构前馈神经网络的语言模型。该章在中文公开数据

集进行了实验比较，从语言模型的种类、纯文本数据的来源和规模、与其它集成语言知识方法的结合等角度进行评价和分析。最后，我们对不同模型的识别结果的分数空间进行部分地可视化，分析了所提知识迁移方法的性质，结果表明，LST 方法作为一种训练阶段的集成方法，可以有效地平滑识别结果的分数空间。

第4章针对自回归语音识别系统没有利用全局文本信息的问题，提出了一种基于自注意力机制的全局上下文语言模型“因果完形填空器”，再基于所提 LST 方法将提炼的全局上下文知识迁移到端到端语音识别系统。该章节弥补了第3章的老师语言模型没有充分利用全局上下文语言知识的问题。实验表明，所提方法可以进一步地提升语音识别的正确率。

第5章利用观察到的语音和文本的语言知识同构现象，提出跨模态地迁移此语言知识提升语音识别效果。该章首先提出了一个利用语音中的语言语义而不显式地语言建模的非自回归端到端语音识别模型 LASO，然后将工业级大规模语言模型 BERT 中的语言知识迁移到 LASO 中。该章节在前两章的基础上，进一步地使端到端语音识别系统抛开文本模态，直接利用语音中的语言语义进行语音识别，并采用跨模态全局语言知识迁移方法提升语音识别的效果。实验表明，所提 LASO 模型在识别准确率高的情况下，识别速度相比自回归模型提升 50 倍，而利用 BERT 中的语言知识则进一步地提升了识别准确率。这些结果表明，利用语音和文本的语言知识同构，跨模态迁移语言知识确实可以提升语音识别的性能。

最后，第6章总结论文的研究工作，并展望未来的研究方向。

第2章 背景介绍及相关工作

2.1 引言

本章介绍端到端语音识别系统的基本背景和端到端语音识别系统中利用纯文本知识的相关工作。

端到端语音识别系统利用一个神经网络直接根据语音特征序列生成文本。整个神经网络采用统一的损失函数进行优化。本文主要关注基于注意力机制的编码器-解码器模型。第2.2节介绍语音识别问题的形式化表示，基于注意力机制的编码器-解码器的模型结构，训练方法和预测方法，以及本文主要用到的基于图结构的前馈神经网络模型。第2.3节介绍利用文本知识提升端到端语音识别性能的相关工作，并分析它们的特点，最后引出有待研究的问题。

2.2 基于注意力机制的编码器-解码器模型

2.2.1 问题的表示

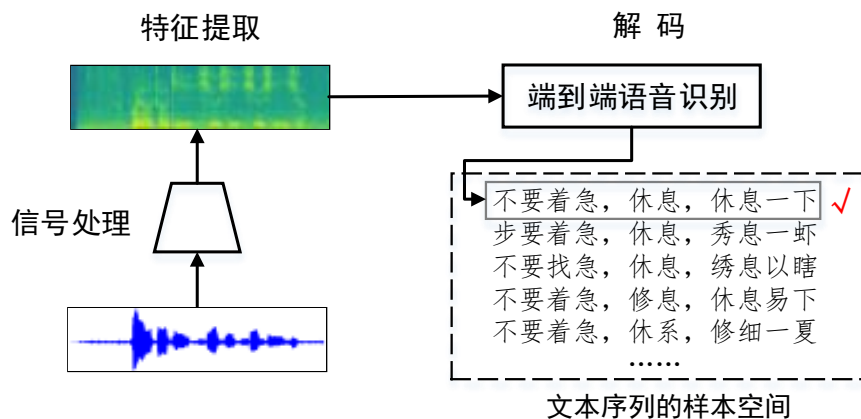


图 2.1 端到端语音识别示意图

Figure 2.1 an illustration of end-to-end speech recognition

语音识别是指将语音转换为其对应的文本。语音波形经过分帧、短时傅里叶变换、梅尔尺度滤波器组变换等，转变为一个向量序列 $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_T]$ 。这段音频对应文本为长度为 J 的词¹序列 y 。语音识别就是求给定语音特征序列 \mathbf{X} 下，

¹本文中，统一用“词” (token) 来表示语音识别系统的单个标注单元，其可以为字母、汉字、词、子词等。

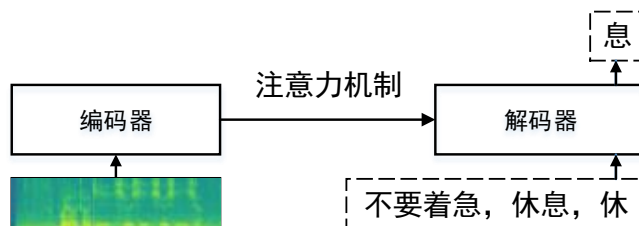


图 2.2 基于注意力机制的编码器-解码器模型

Figure 2.2 the attention-based encoder-decoder model

概率最大的词序列,

$$y^* = \arg \max_y P(y|\mathbf{X}). \quad (2.1)$$

端到端语音识别模型通过一个神经网络来估计概率 $P(y|\mathbf{X})$ 。而获得最大概率的序列 y^* 的过程称为解码。

2.2.2 模型结构

基于注意力机制的编码器-解码器模型将概率通过乘法法则进行拆解。

$$P(y|\mathbf{X}) = P(y_1|\mathbf{X}) \prod_{j=2}^J P(y_j|y_{<j}, \mathbf{X}). \quad (2.2)$$

y_1 一般表示句子的开始, 为 $\langle s \rangle$ 符号, $P(y_1|\mathbf{X}) = 1$ 。 $P(y_j|y_{<j}, \mathbf{X})$ 通过神经网络模型来估计。

具体地, 如图 2.2, 基于注意力机制的编码器-解码器模型分为 3 个部分。编码器将声学特征进行编码, 获得高层表示序列。解码器是一个条件化的语言模型, 计算 $P(y_j|y_{<j}, \mathbf{X})$ 。注意力机制是编码器和解码器的桥梁, 从声学特征的高层表示序列中提取最匹配解码器的表示。

2.2.3 注意力机制

注意力机制是基于注意力机制的编码器-解码器模型的核心部件。它根据解码器状态和编码器状态, 将编码器输出的特征表示序列融合为一个隐表示向量, 以供解码器根据文本和此隐表示向量预测下一个词。

具体到语音识别, 如图 2.2所示, 解码器根据已经预测出的词和声学特征序列, 去预测下一个词(息)。然而, 编码器输出的是序列, 解码器不易直接处理(因为序列长度很长而且可变), 所以需要将编码器的输出序列融合为一个易于处理

的隐表示向量。比较直接的融合方法是 1) 对编码器的输出序列取平均;2) 对于循环神经网络, 取最后一个输出的状态表示。然而这样做的话, 编码器预测不同的词的时候, 输入的都是同样的编码器融合向量, 这显然不利于区分不同的词的发音。

编码器输出的序列, 大体上表示了不同的语音帧对应的特征。如图 2.2 中所示的例子, “息” 对应的是哪一个语音帧对应的特征呢? 如果我们能自动地选出序列中合适的表示, 则会有利于解码器根据不同的声学表示 (即发音相关的信息) 来预测不同的词。

注意力机制 [65, 66] 就是一种利用加权平均的方式来将编码器的输出序列进行融合的机制。其权重是根据解码器的状态和编码器输出序列进行计算的, 所以是根据解码器不同的状态动态变化的, 适用于挑选适合当前解码器需要预测的词的编码器输出。

一般地, 如图 2.3 所示, 注意力机制的输入可以分为查询 (query) 向量和键值对 (key-value pair)。注意力机制可以被描述为如下查询过程。

1. 利用函数 a , 对查询向量与每一个键向量, 计算注意力分数。
2. 利用 softmax 函数对注意力分数进行归一化, 生成一组权重。
3. 利用生成的权重, 对值向量进行加权求和, 得到融合结果。

这一过程模仿了从数据库中进行查询的过程。具体到基于注意力机制的编码器-解码器模型, 编码器的输出序列可以看做是存储的一系列键值对, 解码器当前的状态是查询向量, 解码器根据当前的状态从存储的键值对中提取了最匹配的融合结果。图 2.3 中, 注意力权重的颜色深浅表示了值的大小, 颜色最深的值表示查询向量最 “注意” 到对应的键向量。需要指出的是, 键向量和值向量可以是相同的, 依然可以表示查询过程。

函数 $\alpha = a(\mathbf{q}, \mathbf{k})$ 计算一个注意力分数。可以采用不同的方式来进行计算, 如多层感知机 (multilayer perceptron, MLP), 内积, 和双线性形式:

$$\begin{array}{ll}
 \text{多层感知机} & a(\mathbf{q}, \mathbf{k}) = \tanh(\mathbf{q}\mathbf{W}_1 + \mathbf{k}\mathbf{W}_2)\mathbf{v}^T, \\
 \text{内积} & a(\mathbf{q}, \mathbf{k}) = \mathbf{q}\mathbf{k}^T, \\
 \text{双线性形式} & a(\mathbf{q}, \mathbf{k}) = \mathbf{q}\mathbf{W}\mathbf{k}^T,
 \end{array} \tag{2.3}$$

其中, \mathbf{W}_1 、 \mathbf{W}_2 、 \mathbf{W} 和 \mathbf{v} 都是可训练的参数。所得结果 α 是一个标量。

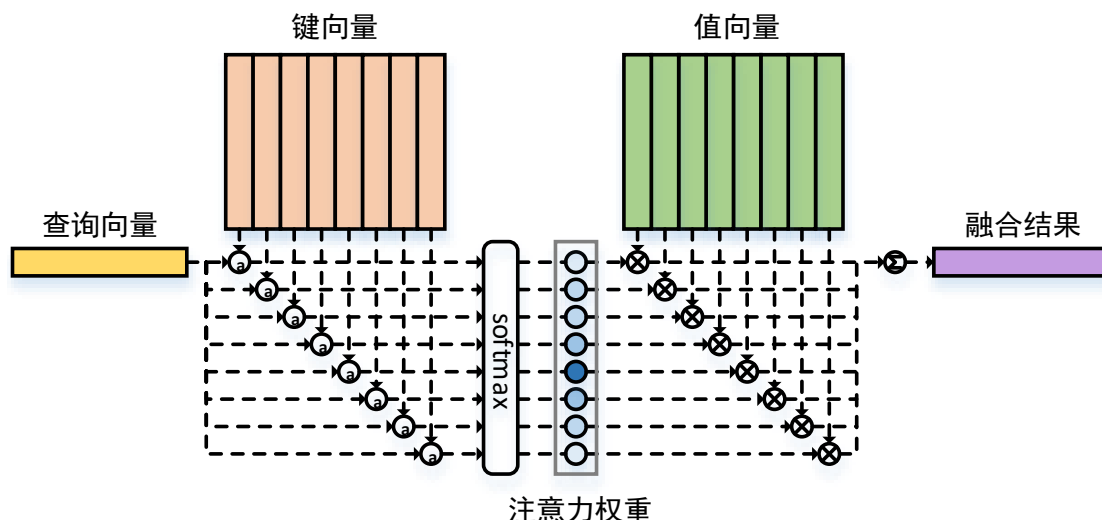


图 2.3 注意力机制

Figure 2.3 the attention mechanism

$\text{softmax}(\alpha)$ 是 $\arg \max$ 函数的松弛形式，其令输入向量 α 中最大的元素趋近于 1，其它元素趋近于 0，向量各元素都为正且和为 1，构成一个概率的形式。各元素的计算如下：

$$\frac{\exp(\alpha_i)}{\sum_{j=1}^K \exp(\alpha_j)}, \quad (2.4)$$

其中， α_i 为一个元素， K 为向量的元素总个数，对应于图 2.3 键向量或值向量的个数。

可以看出， softmax 作为一种松弛形式的 $\arg \max$ 函数，使得注意力机制可以全面地处理整个序列，且更容易被训练。基于 softmax 的注意力又称为软注意力 (soft attention)。其它的还有只关注序列中一个位置的基于 $\arg \max$ 的硬注意力 (hard attention)[67]，和不需要计算所有键向量而只计算前面一部分就可以触发一次查询的单调注意力 (monotonic attention)[68]。

多头注意力机制 (multihead attention) 先分别将查询向量、键向量、值向量分别投影到几个子空间 (subspace)，再运行注意力机制，最后再将结果拼接融合。多头注意力相当于融合了不同的注意模式，可以提升注意力机制的性能。在 2.2.5 小节将详细介绍一种多头注意力的实现。

2.2.4 训练与预测

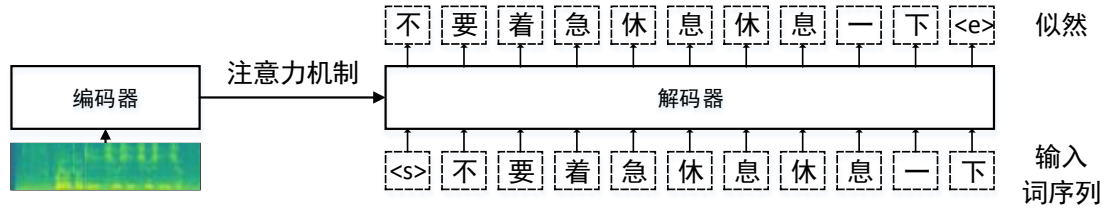


图 2.4 教师强制训练

Figure 2.4 teacher-forcing training

2.2.4.1 训练

基于注意力机制的编码器-解码器的基本训练方法为极大似然估计 (maximum likelihood estimation, MLE)。具体上, 假设编码器-解码器模型表示如下对数似然函数:

$$\log P(y_j | y_{<j}, \mathbf{X}) = f(y_{<j}, \mathbf{X}, \theta), \quad (2.5)$$

其中, θ 表示神经网络的参数。极大似然估计最小化如下负对数似然函数²:

$$\begin{aligned} -\log P(y|\mathbf{X}) &= -\sum_{j=2}^J \log P(y_j | y_{<j}, \mathbf{X}) \\ &= -\sum_{j=2}^J f(y_{<j}, \mathbf{X}, \theta), \end{aligned} \quad (2.6)$$

其中, 每一个对数概率都是由编码器-解码器模型 f 计算得到的。

实践中, 训练样本中的词序列是整个输入进解码器, 计算词序列的似然, 而不是一个一个依次计算, 如图 2.4 所示。这种训练方式称为教师强制 (teacher-forcing)。这样训练最大化地利用了神经网络工具包的并行计算能力, 最为高效³。

² $P(y_1|\mathbf{X}) = 1$, 所以略去, 见式 2.2。

³然而这种训练方式一定程度上会带来训练和预测阶段模型的不匹配, 即预测阶段是一个词一个词生成的, 中间有可能有错误, 但训练时却全部由正确答案训练。这种不匹配称为曝光偏差 (exposure bias)。曝光偏差在基于循环神经网络的模型中表现比较明显, 在基于 transformer 的模型中不大明显。

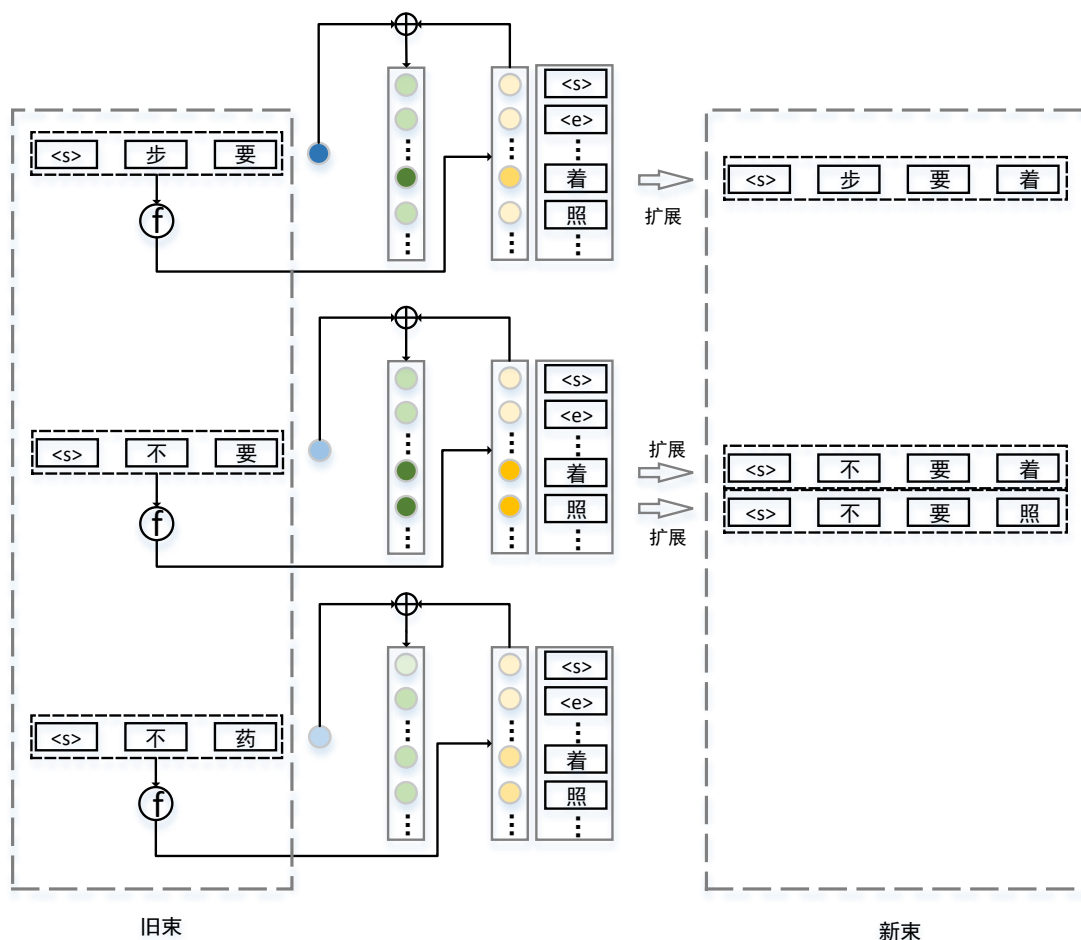


图 2.5 一步束搜索的示意图。这里假设束的宽度为 3。左边表示旧的束，右边表示扩展出的新束，蓝色的圆点表示旧累积分数，黄色圆点表示根据旧束中状态，用语音识别模型计算出的每一个词的分数的，绿色圆点表示新的候选的累积分数。颜色深浅表示分数大小。

Figure 2.5 An illustration of one step of beam search. We assume that the beam width is 3. The left part denotes the previous beam. The right part denotes extended new beam. The blue dots are previous accumulated scores. The yellow dots are scores of each token in the vocabulary. These are computed by the end-to-end ASR model. The green dots are accumulated scores of the candidates. The shade of the color indicates the scales of the scores.

2.2.4.2 预测

预测是指根据模型计算得到的对数概率，从所有可能的词序列中找出概率最大的词序列。根据语音识别的传统，此过程又称为解码 (decode)。所有可能的词序列是很多的，假设词表 \mathcal{V} 大小为 $|\mathcal{V}|$ ，序列长度最大为 L ，那么词序列有 $|\mathcal{V}|^L$ 种可能。采用动态规划的方法，自左向右地进行搜索，计算复杂度为 $O(|\mathcal{V}|^2 L)$ 。然而，在实际应用中，直接采用动态规划的方法依然耗时过多。引入束搜索 (beam search) 技术在动态规划迭代的过程中动态地剪枝，大大降低了计算量 [69]。

束搜索以广度优先 (breadth-first) 的方式扩展搜索树，对于搜索树的每一层，计算每一个节点的累积分数 (accumulated scores)，然后根据累积分数剪枝，在保留下来的节点的基础上，进一步扩展下一层。具体到语音识别的预测，如图 2.5 所示。左边部分为当前保留下来的束 (beam)。模型分别对束里 3 个状态⁴计算出词表上的对数概率分布，与原来累积分数求和，生成累积分数的候选，构成搜索树下一层节点。然后，再根据某些准则进行剪枝，确定这一步束搜索的最后结果，即图中所示的“新束”。

束搜索的剪枝策略是启发式的 (heuristic)。最简单的策略是直接保留分数最高的 N 个节点。另一种策略是剪去与累积分数最大节点相差大于某个阈值的节点，此阈值也可以自适应调整⁵。基于注意力机制的编码器-解码器模型一般就采用最简单的策略，只保留分数最高的几个候选，就可以获得很好的性能。

当某一个候选序列扩展到句子结束符号 $\langle e \rangle$ 时，就单独保存。当所有候选序列都结束，迭代到最大步数，搜索就可以停止了。

最终搜索结果是最后保存下来的候选结果里，累积分数最高者。实际上，还可以进一步地对候选结果用其它的语言模型重打分 (rescore)，然后重新筛选出分数最高者，提升识别的性能。

⁴这里的“状态”是广义上的对词序列前缀的表示。对于循环神经网络的实现，状态表示一个隐变量向量。对于 transformer 网络，则表示前缀文本。

⁵这种根据分数来进行剪枝的策略一般用在混合模型中。详情请参考语音识别工具箱 KALDI 的解码器文档以及 ProcessEmitting 函数。 <http://kaldi-asr.org/doc/decoders.html>

<https://github.com/kaldi-asr/kaldi/commit/f273ec5b6fe2687cff3967bf497c6eed06dd80e3#diff-dae0571bdf4acd3070fb02879508ddffff1a8dd217b872ce6d5e0ac26fa40121R173>

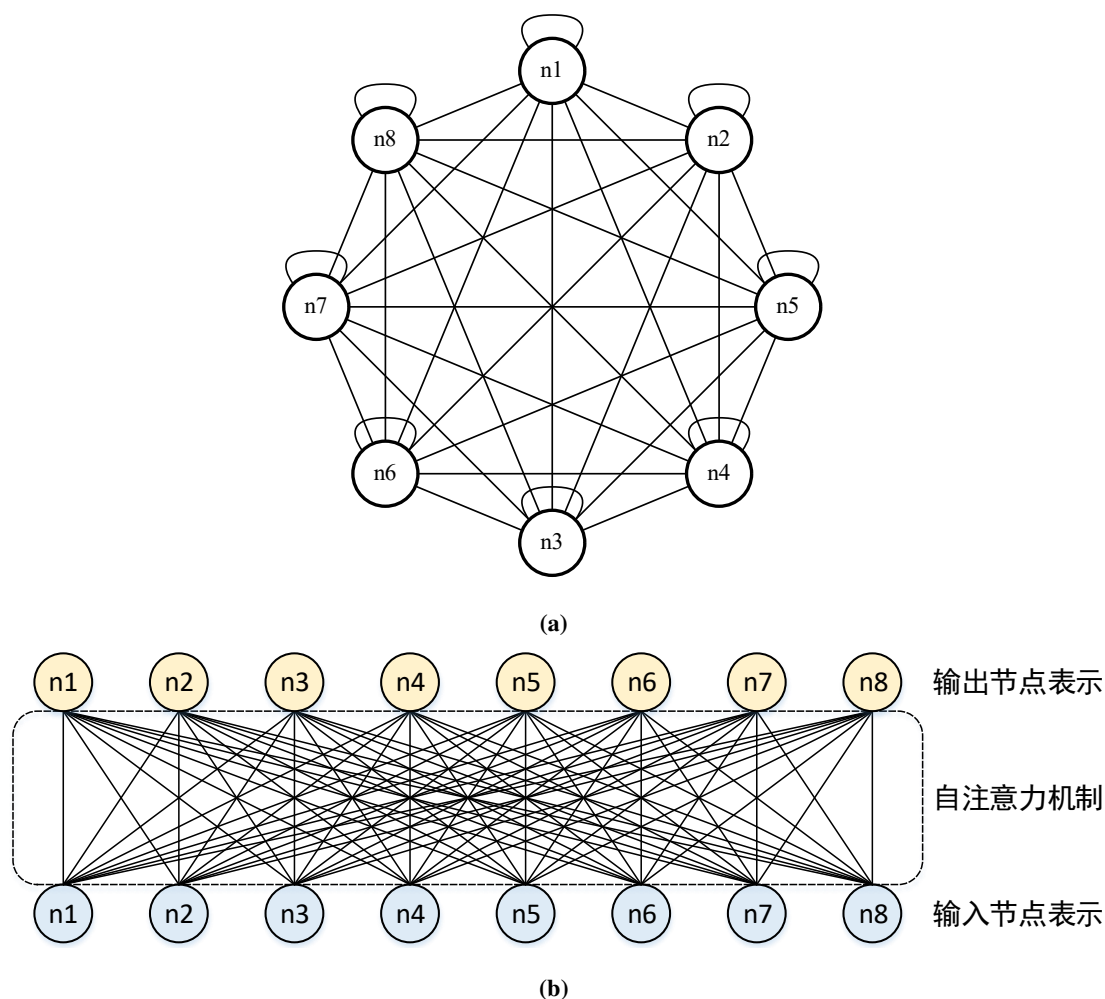


图 2.6 基于图结构的前馈神经网络。(a) 全连接图, (b) 自注意力的输入和输出, 每一个节点为一个节点表示向量。

Figure 2.6 A graph-based feedforward neural network. (a) A fully-connected graph, (b) the inputs and outputs of self-attention, each node denotes a representation vector for a node.

2.2.5 基于图结构的前馈神经网络模型

本文所用到的端到端语音识别系统全部为基于图结构的前馈神经网络模型 [70, 71]。这种模型被称为 transformer⁶, 其基础是自注意力机制。自注意力机制是一种查询向量、键向量和值向量都来自于同一输入向量序列的注意力机制。

2.2.5.1 序列中的图结构

假设用图 (graph) 来表示序列之间的关系。如图 2.6(a) 所示, 每一个节点表示序列中的一个向量。例如一个声学特征序列 $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_8]$, 在图中 n1 表示

⁶目前该模型还没有确切的翻译, 这里直接使用英文。

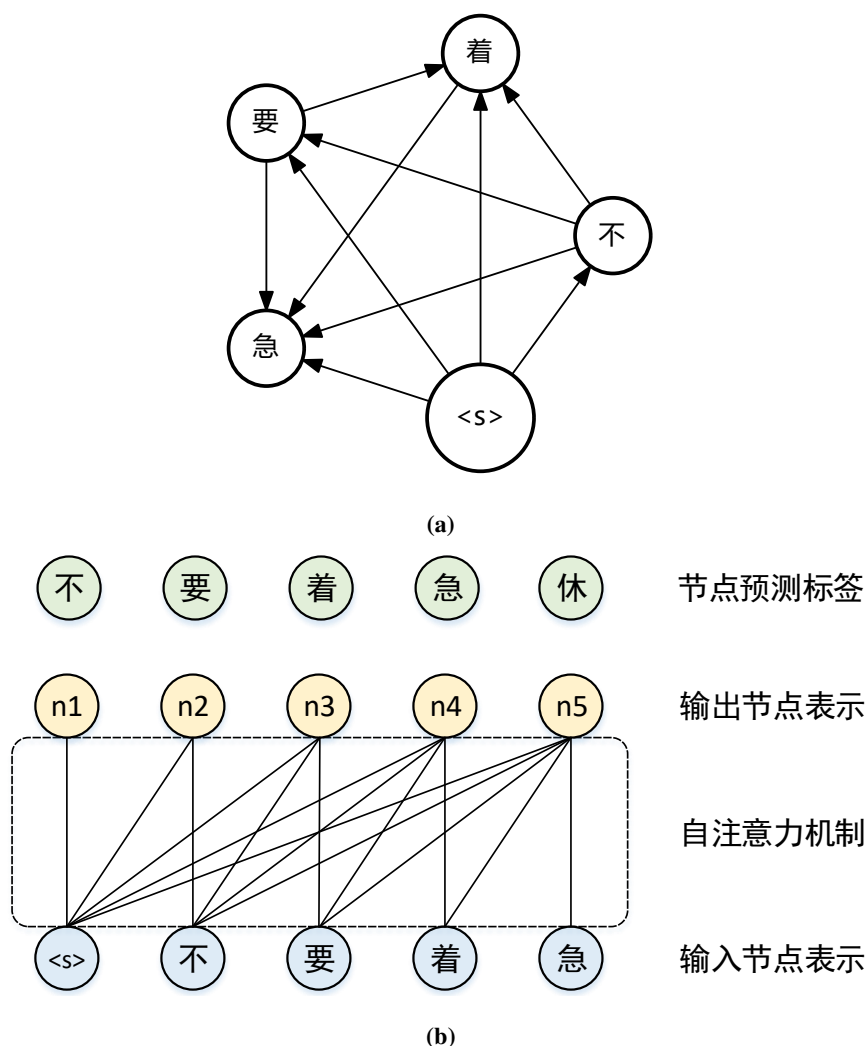


图 2.7 语言模型的图结构表示。该图表示了序列的一部分前缀。(a) 语言模型的图结构表示，(b) 语言模型的消息传播模式。

Figure 2.7 A graph of language model. This figure shows a partial of prefix of the example. (a) a graph of a language model, (b) the message-passing pattern of the language model.

x_1 , n_2 表示 x_2 , 以此类推。每一个特征都和所有特征 (包括其本身) 都有联系, 所以构成一个带自环 (self-loop) 的全连接图 (fully-connected graph)。自注意力机制利用消息传播 (message-passing) 建模这种图结构 [72]。具体地, 每一个节点的查询向量和所有节点的键向量计算注意力分数, 再将所有节点的值向量利用加权和来聚合 (aggregation), 这样, 所有节点的消息就传播到了这个节点, 进入到下一层, 如图 2.6(b) 所示, 节点为“消息”, 边上的权重通过自注意力机制计算。利用自注意力机制, 可以直接对序列中各个向量的关系建模。编码器就是在建模这种全连接图。

对于解码器，同样地利用图来表示序列中向量之间的关系，如图 2.7(a) 所示，图表示了例子序列的前缀 “<s> 不要着急” 的关系。可以看出，其与图 2.6 中编码器不同在于，序列中后一个词无法看见前一个词，所以整个图构成了一个有向图，而非类似编码器中的无向图。词序列中每一个词都是依据前缀中所有词的传递的“消息”来预测下一个词，如图 2.7(b) 所示。

2.2.5.2 基于自注意力机制的实现

下面介绍由自注意力机制实现上一小节所描述的图结构建模。

整体结构。如图 2.8 所示，transformer 全部由注意力机制构成。编码器部分首先利用卷积模块对声学特征序列降采样，一方面可以捕捉声学特征的局部性，一方面可以减小序列长度，使注意力机制容易学习。降采样之后的序列与位置编码叠加，然后被输入 transformer 编码器。transformer 编码器全部为自注意力机制，即查询向量，键向量，值向量都为同一输入序列。解码器部分首先提取文本中每一个词的嵌入 (embedding)⁷，然后将嵌入与序列位置编码叠加，输入 transformer 解码器。解码器包括两种注意力机制：第一种自注意力机制为带掩蔽的注意力机制，控制每一个词只能“注意到”它前面的词，构成如图 2.7 所示的图结构；第二种互注意力机制 (cross-attention) “注意”编码器的输出，获取匹配的发音，其查询向量为和词序列有关的表示序列，键向量和值向量为和发音有关的编码器输出。解码器的最后为仿射变换和 softmax 函数，其计算出词表上的概率分布。

部件实现。注意力机制为多头尺度放缩内积注意力。

$$\text{attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{D_k}} + \mathbf{M}\right)\mathbf{V}, \quad (2.7)$$

$$\mathbf{H}_i = \text{attention}(\mathbf{Q}\mathbf{W}_i^q, \mathbf{K}\mathbf{W}_i^k, \mathbf{V}\mathbf{W}_i^v), i = 1, \dots, N_H \quad (2.8)$$

$$\text{mha}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = [\mathbf{H}_1; \dots; \mathbf{H}_H]\mathbf{W}^o. \quad (2.9)$$

其中， \mathbf{Q} 、 \mathbf{K} 和 \mathbf{V} 分别为由查询向量、键向量和值向量构成的矩阵， D_k 为键向量的维度， \mathbf{M} 为掩蔽矩阵， \mathbf{H}_i 表示一个头的输出。这里的矩阵都为行主序 (row-major order) 的，即一行表示序列中的一个向量。 \mathbf{W}_i^q 、 \mathbf{W}_i^k 、 \mathbf{W}_i^v 和 \mathbf{W}^o 为参数矩阵。

⁷一个词的“嵌入”是指在向量空间中指定一个向量来代表这一个词。嵌入向量在模型训练的过程中更新，最终获得的向量和这个词具有某种意义上语义的同构。

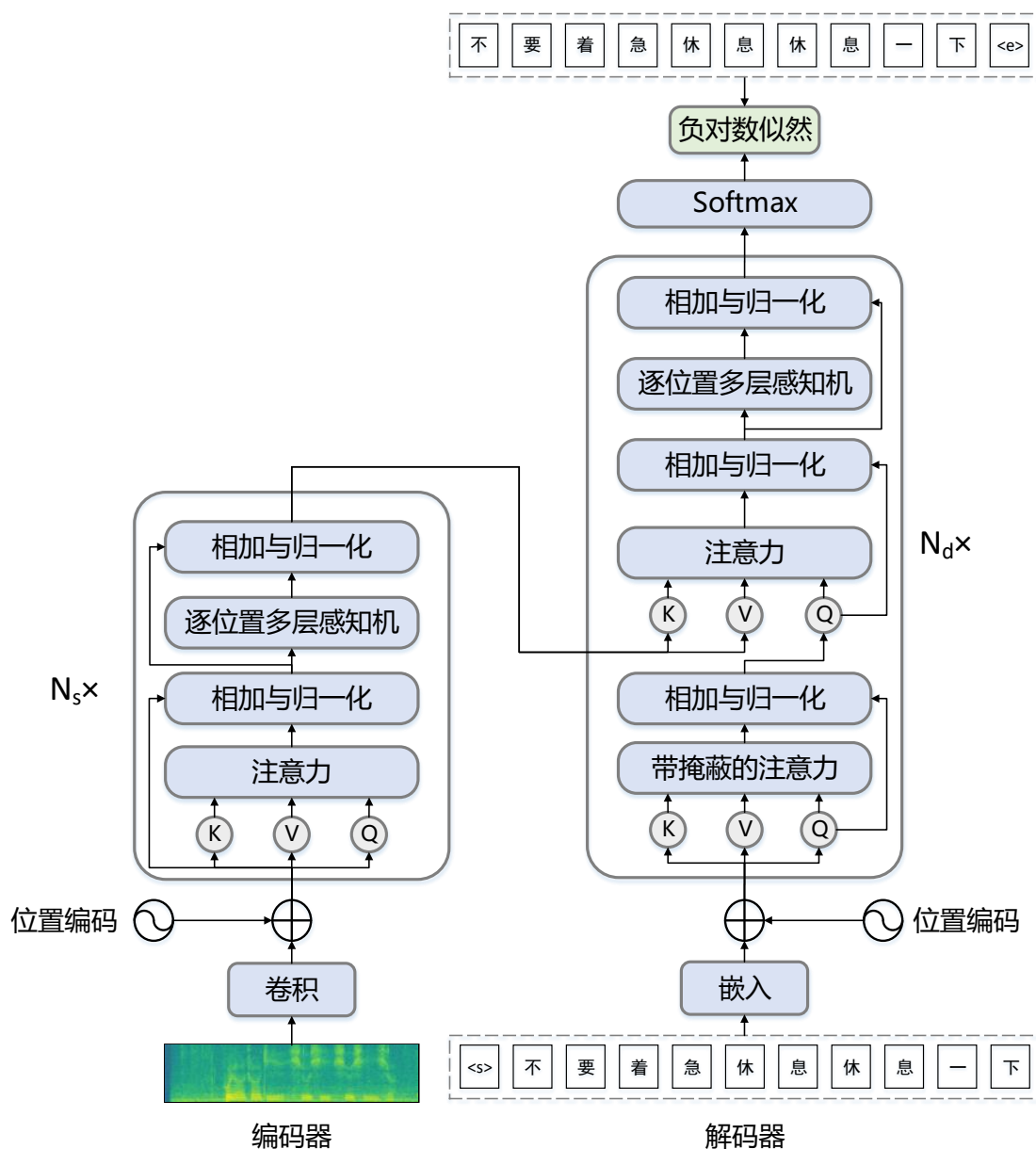


图 2.8 transformer 结构示意图

Figure 2.8 the architecture of transformer

掩蔽矩阵 \mathbf{M} 为控制消息传播路线的工具：

$$\mathbf{M}_{i,j} = \begin{cases} -\infty, & \text{注意力分数掩蔽为 0} \\ 0. & \text{其它} \end{cases} \quad (2.10)$$

这样，在经过 softmax 函数以后，对应位置的注意力分数就会变成零，对应节点的消息就无法传播到下一层了。对于解码器，需要让每一个词都不能“看见”后边的词，所以掩蔽矩阵为对角线之上的元素全部为 $-\infty$ 的上三角矩阵。

transformer 中另一个重要部件是“逐位置的多层感知机”。它对序列中每一

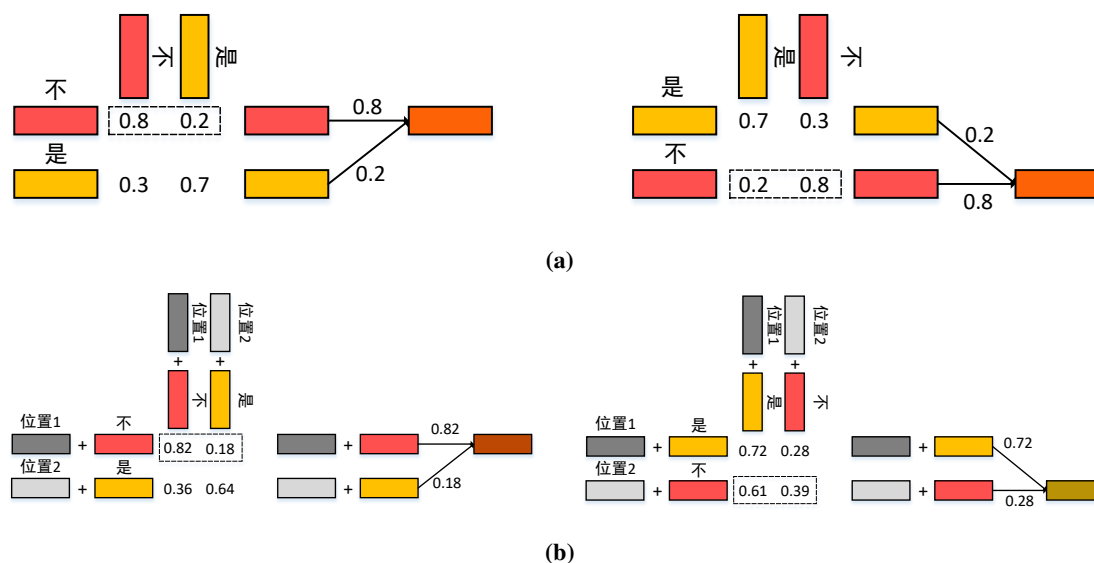


图 2.9 自注意力机制的位置编码。(a) 无位置编码时，交换两个字的融合结果比较；(b) 带位置编码时，交换两个字的融合结果比较。

Figure 2.9 The positional encodings of self-attention. (a) a comparison of exchanging two tokens without positional encodings; (b) a comparison of exchanging two tokens with positional encodings.

个向量都用一个多层感知机进行变换：

$$\text{MLP}(\mathbf{x}) = g(\mathbf{x}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2, \quad (2.11)$$

其中， \mathbf{W}_1 、 \mathbf{W}_2 、 \mathbf{b}_1 和 \mathbf{b}_2 为参数， g 为某种激活函数。常用的激活函数有 ReLU[73–75]、swish[76] 和 GLU[77] 等。在本文中，主要使用 GLU 激活函数，其形式为⁸：

$$\text{sigmoid}(\mathbf{x}) = \frac{1}{1 + \exp(-\mathbf{x})}, \quad (2.12)$$

$$g([\mathbf{x}_1; \mathbf{x}_2]) = \mathbf{x}_1 \otimes \text{sigmoid}(\mathbf{x}_2), \quad (2.13)$$

\otimes 表示逐元素相乘。可以看出，GLU 会使输入向量的维数缩小一半，所以为了保持一致，式 2.11 中 \mathbf{W}_1 需要把维度扩大一倍。

使用基于内积的自注意力机制进行序列建模的一个问题是它会忽略序列中的顺序 (order)。这是由于基于内积的自注意力的基本操作 softmax、求和都是符合交换律 (commutative property) 或者置换不变的 (permutation invariant)⁹。举例来说，一个句子中“不”的位置对整个句子的语义作用很大。但是，如图 2.9(a) 所

⁸本文中，指数函数 \exp 内的向量表示逐元素取指数。

⁹多头注意力机制中经过了不同权重的仿射变换，内积相关的操作不一定可交换。

示，“不”和“是”的位置交换以后，“不”位置对应的融合向量是一样的。解决这一问题的一个办法是使用位置编码，如图 2.9(b) 所示，将和位置相关的编码加到字对应的表示上，两个字的位置交换以后，对应加上的位置编码也是不同的，导致最后融合的结果是不同的，这样建模了序列中的顺序。位置编码可以是可训练的。另一种常用的是利用正余弦函数 [70]:

$$\begin{aligned} \mathbf{P}_{i,2j} &= \sin\left(\frac{i}{10000^{\frac{2j}{D}}}\right), \\ \mathbf{P}_{i,2j+1} &= \cos\left(\frac{i}{10000^{\frac{2j}{D}}}\right), \end{aligned} \quad (2.14)$$

其中， \mathbf{P}_i 表示序列中第 i 个位置的编码， $\mathbf{P}_{i,j}$ 表示其中第 j 个元素。这种直接在序列中叠加位置编码的方法称为绝对位置编码。另外还有一种在注意力机制的计算中引入位置编码的方法称为相对位置编码 (relative position encodings)[78]。采用相对位置编码更有利于处理模型未见过的长度的序列。

残差连接 (residual connection)[79] 为消息建立直接连接的通路，让底层消息可以直接传输到高层，让训练时梯度容易反向传播。其形式为:

$$\mathbf{x} + g(\mathbf{x}), \quad (2.15)$$

其中， g 为非线性变换。也就是说，残差连接将输入与非线性变换的输出加和，制造了一条通路。

层归一化 (layer normalization)[80] 对一个序列中的向量归一化，可以稳定训练¹⁰。具体上，层归一化按照如下方式计算¹¹:

$$\mu = \frac{1}{N_{\text{elem}}} \sum_{i=1}^{N_{\text{elem}}} h_i, \quad (2.16)$$

$$\sigma = \sqrt{\frac{1}{N_{\text{elem}}} \sum_{i=1}^{N_{\text{elem}}} (h_i - \mu)^2}, \quad (2.17)$$

$$\tilde{\mathbf{h}} = \frac{\mathbf{s} \otimes (\mathbf{h} - \mu)}{\sigma} + \mathbf{b}, \quad (2.18)$$

其中， h_i 为向量 \mathbf{h} 的一个元素， N_{elem} 为元素个数， μ 和 σ 分别为算出的均值和标准差。放缩因子 \mathbf{s} 和偏置 \mathbf{b} 为可训练的参数。归一化的位置可以根据情况进行调整，[81, 82] 指出将归一化放在自注意力前面有助于稳定训练。

¹⁰批归一化 (batch normalization)、层归一化 (layer normalization)、组归一化 (group normalization) 等归一化方法为什么有效的内部工作机理，现在还未有定论，是一个开放问题。

¹¹这里向量减去一个标量的意思是向量中每一个元素都减去这一个标量。

在最后，如图 2.8，通过一个仿射变换将解码器输出的向量映射到词表维度，再通过 softmax 函数计算词表上的概率分布：

$$\frac{\exp(o_k)}{\sum_{j=1}^{|\mathcal{V}|} \exp(o_j)}, \quad (2.19)$$

其中， o_j 表示仿射变换输出向量 \mathbf{o} 的第 j 个元素， k 表示词表中第 k 个词。

模型特点。相对于循环神经网络的实现，基于自注意力机制实现的编码器-解码器模型具有以下优势。

1. 自注意力机制的实现为几个矩阵相乘。相比于循环神经网络需要一步一步迭代计算，自注意力机制容易被并行计算。

2. 对于解码器，自注意力机制对解码中间产生的错误相比循环神经网络更鲁棒。这是因为，对于循环神经网络，其输入为编码前缀的隐变量和当前时刻的输入。一个错误的输入极大地影响了结果。而自注意力机制，历史前缀都可以给当前时刻提供消息，所以更鲁棒¹²。

基于自注意力机制的模型也确实取得了良好的识别表现。

然而，基于自注意力机制的模型也存在一些缺点。一个主要的缺点就是，在计算自注意力的时候，需要进行一个和矩阵长度有关的大规模矩阵乘法。对于非常长的序列，会有内存溢出等现象。有一些工作通过引入保存外部记忆的方法存储历史信息 [83]。也有一些工作提供了高效计算注意力机制的方法 [84]。

¹²在我们的实验中，发现最小词错误率训练 (minimum error rate, MWER) 对于基于循环神经网络的编码器-解码器模型效果提升明显，但是对于 transformer 则效果不大。我们分析这就是由于 transformer 模型对错误更鲁棒导致的。

2.3 端到端语音识别系统中纯文本知识利用研究现状

2.2节介绍了基于注意力机制的编码器-解码器模型的系统架构以及训练和预测方法。可以看出，编码器-解码器模型需要利用语音-文本成对数据进行训练。

本节介绍针对使编码器-解码器模型可以利用纯文本数据的相关研究，分析它们的特点，并引出有待解决的问题。使编码器-解码器模型可以利用纯文本数据的相关方法可以分为两类：基于融合的方法和基于合成数据的方法。

2.3.1 基于融合的方法

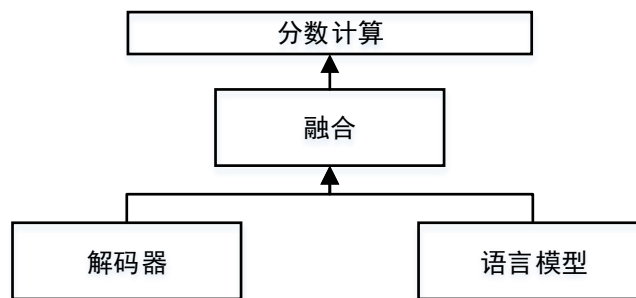


图 2.10 基于融合的方法

Figure 2.10 fusion-based methods

基于融合的方法首先在纯文本数据上训练语言模型 (统计语言模型或神经网络语言模型)，然后在预测阶段，将语音识别模型和语言模型的信息融合起来进行决策 [85–92]。如图 2.10。

根据融合方法的不同，融合法可以分为浅融合和深融合。

浅融合是指将解码器输出的分数和语言模型输出的分数加权融合 [85, 86, 88]。语言模型可以是统计语言模型，也可以是神经网络语言模型。融合的公式为：

$$\log P(y|\mathbf{X}) = \log P_{\text{ASR}}(y|\mathbf{X}) + \gamma \log P_{\text{LM}}(y), \quad (2.20)$$

其中， γ 为加权系数。一般 γ 是一个超参数，需要通过实验确定。 γ 可以用密度估计法来计算 [92, 93]。浅融合可以灵活地利用各种训练好的语言模型，较为方便。然而， λ 的确定需要一定技巧和调整。

深融合则是利用一个多层感知机对解码器隐藏层状态和语言模型隐藏层状态进行融合 [85, 87, 90, 91]。由于多层感知机中有可训练的参数，所以深融合法

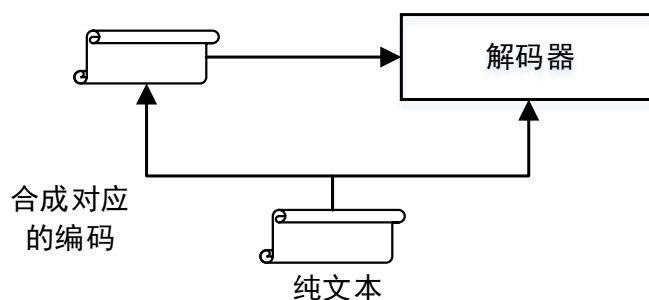


图 2.11 基于合成数据的方法

Figure 2.11 synthetic data based methods

需要进一步利用语音-文本成对数据训练多层感知机。这在一定程度上限制了深度融合的使用。冷融合法提出先训练语言模型，然后固定语言模型，辅助语音识别模型的训练。同时，冷融合法不利用语言模型隐藏状态，而是利用语言模型提供的概率分数，这使得冷融合法可以利用统计语言模型，而非只能利用神经网络语言模型 [87]。

根据一些经验比较 [88, 89]，在实际应用中，浅融合和深度融合中的冷融合效果比较好，一般的深度融合效果相对较差。融合法，特别是浅融合法的优势在于，它可以灵活地利用提前训练好的语言模型。特别是对于一些需要领域自适应的场合，通过浅融合法，利用目标领域的语言模型，简单地调整参数即可实现自适应。然而，根据实践中的经验，由于端到端模型是语音语言一体化建模，外部语言模型对端到端模型的语言建模部分并不容易很好地纠正，所以实践中性能提升比较有限。融合法的另一个问题是，它在预测阶段给系统增加了一个额外的语言模型，增加了系统预测的计算代价。

2.3.2 基于合成数据的方法

基于合成数据的方法根据纯文本数据生成对应的编码器编码，这样获得了成对数据，然后就可以训练整个模型 [94–102]，如图 2.11。自然地，虽然这种方法训练到了整个端到端语音识别模型，但由于编码器编码是根据文本合成而非从自然场景获得，并不具备典型的语音特性(如丰富的语速变化、丰富的说话人音色等)，这种方法并不能很好地提高编码器部分的性能。基于合成数据的方法的目的还是提升整个模型的语言建模部分，即解码器的性能。

最直接的合成数据的方法是利用语音合成 (text-to-speech, TTS) 系统合成语

音 [94, 97, 98, 101]。虽然这种方法可以利用纯文本，提升语音识别的性能，但是合成语音数据首先需要一个较好的语音合成系统，并且大规模合成语音资源开销较大。

另一种合成数据的思路是，不合成语音，而合成某种向量表示，可以让解码器训练。基于这种思路的方法可以分成四种。第一种是利用文本生成音素表示序列代表发音，再将音素表示序列输入到某个编码器，再将编码输入到解码器 [96]。第二种是利用一个神经网络模型学习文本到编码器输出的变换，然后用这个神经网络对大规模文本生成编码器输出 [95]。这两种方法虽然都可以提升语音识别性能，但是训练过程比较复杂，需要额外加一些神经网络模块。针对这个问题，第三种方法直接采用一个可训练的向量表示“没有语音”，当利用纯文本数据训练解码器时，给解码器输入这个向量，表示光训练语言建模能力 [99, 100]。第四种方法则将解码器进行分拆，让底层为一个纯粹的语言模型，使其可以通过纯文本训练 [102]。

合成数据的方法在利用纯文本数据的同时，不额外增加预测阶段的计算代价。然而这种方法在训练阶段利用纯文本数据，不适合做快速的领域自适应。而且，这种方法必须利用原始文本进行训练，不适用于利用现在流行的大规模预训练语言模型。

2.3.3 有待研究的问题

前面两小节介绍了目前主流的两类利用纯文本数据提升端到端语音识别的方法。然而，这两种方法存在以下问题。

基于融合的方法需要在预测阶段使用语言模型，这导致预测阶段的复杂度增加。同时，在预测阶段使用语言模型，导致其只能采用单向的自回归语言模型。但当前流行的大规模预训练语言模型有很多都是双向的，如 BERT [7]。这两点限制了该方法的使用。

基于合成数据的方法不增加预测阶段的计算代价。然而这一类方法也需要自行使用纯文本数据进行训练，且训练模式须与解码器配套，为单向自回归模式。目前流行的大规模预训练语言模型往往利用了十分庞大的纯文本数据训练，对于语音识别问题再自行利用文本训练相对来说较为耗时费力。而且现在流行的 BERT 等 [7] 双向语言模型表现了强大的语言建模能力，合成数据的方法却无法直接使用。

针对上面提到的问题，本文希望提出一种方法：

1. 利用纯文本数据中的语言知识提升端到端语音识别模型的性能，但不增加预测阶段计算代价；
2. 可以方便地利用预训练语言模型。

为此，本文提出一种基于迁移学习的方法，将预训练语言模型中的知识迁移到端到端语音识别系统。后续第3、4、5章详细探讨本文所提方法。

第3章 上文语言知识迁移

3.1 引言

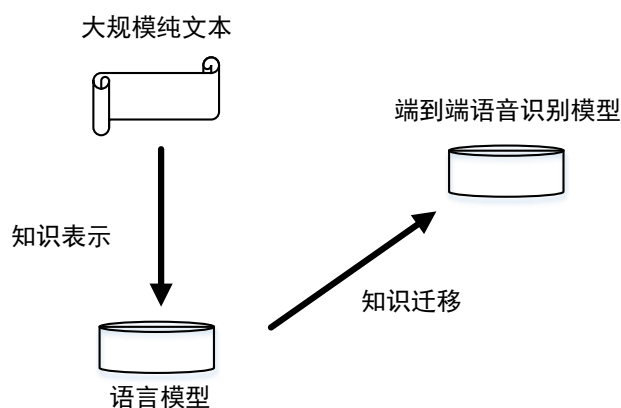


图 3.1 本文提出的基于老师-学生学习的 LST 方法

Figure 3.1 the proposed teacher-student learning based LST method

根据第2章的介绍可知，端到端语音识别系统难以直接利用纯文本数据训练。另一方面，语言模型是利用大规模纯文本数据训练得到的。自然地，我们可以想到将存储在语言模型中的知识迁移到端到端语音识别模型，来提升语音识别的效果。迁移学习 (transfer learning) 是指利用源领域任务的知识来提升目标领域任务的性能 [103]。在此设定下，语言模型即是源领域任务，端到端语音识别为目标领域任务。

本章提出一种基于老师-学生学习 (teacher-student learning) 的方法将语言模型中的语言知识迁移到语音识别模型，称为 LST (Learn Spelling from Teachers)[104]。简而言之，该方法首先利用语言模型将大规模纯文本知识表示起来，然后利用迁移学习的方法将知识迁移到端到端语音识别模型，如图 3.1。它是一种训练阶段知识集成方法，不增加测试阶段的系统开销；它对语言模型没有特殊要求，可以灵活地利用各种语言模型。在本章中主要利用了经典的单向语言模型，所以迁移的是上文知识。本章中所提的方法是一种一般化地利用大规模纯文本数据知识的方法，可以应用在机器翻译、图像描述 [105] 等各类文本生成任务中。

第3.2节介绍老师-学生学习的背景和方法，第3.3节具体介绍 LST 方法，第3.4节介绍实验结果，最后第3.5节小结本章。

3.2 老师-学生学习训练方法

老师-学生学习是指利用一个“老师模型”提供监督信号 (supervision), 让“学生模型”去“模仿”老师模型的行为, 从而提升学生模型的性能。这一思想最早被用于模型压缩 [106–110]: 人们发现让大模型甚至一组集成 (ensemble) 的模型作为老师模型, 去训练体积较小的模型, 比直接用标注来训练小模型获得的性能要好。这一过程即可以被看做是将知识从大模型迁移到小模型, 将大模型压缩为小模型。这一方法也被应用在自适应 (adaptation) 领域 [111, 112]: 用源领域数据上训练的模型提供监督信号, 训练目标领域上的模型, 使目标领域的模型同时获得两个领域的知识。

根据老师模型所提供的监督信号的不同, 老师-学生学习可以分为基于标签和基于隐藏层特征两类¹[113]。基于标签的老师-学生学习, 老师模型提供概率分布作为软标签 (soft labels), 通过优化学生模型和老师模型概率分布之间的差异, 令学生模型模仿老师模型。相比于人类标注的基于独热码 (one-hot) 的硬标签 (hard labels), 软标签包含了更多知识: 如不同类别之间置信度相对大小等。这种知识被称为暗知识 (dark knowledge)[109, 114]。另一种基于隐藏层特征的方法则抽取老师模型的中间层表示作为监督信号, 来优化学生模型 [107, 110, 115]。相对于基于标签的方法, 这种方法不需要老师模型和学生模型有相同的输出, 而只需有匹配的隐藏层语义表示即可, 所以更为灵活。总的来说, 老师-学生学习就是通过某种损失函数来最小化老师模型和学生模型之间的差异。

对于基于标签的老师-学生学习, 最常见的度量模型差异的损失函数是 KL 散度 (Kullback–Leibler divergence, KLD):

$$D_{\text{KL}}(P_{\text{teacher}} || P_{\text{student}}) = \sum_x P_{\text{teacher}}(x) \log \frac{P_{\text{teacher}}(x)}{P_{\text{student}}(x)}, \quad (3.1)$$

其中 P_{teacher} 和 P_{student} 分别表示老师模型和学生模型输出的概率分布。对于基于隐藏层特征的老师-学生学习, 最常见的度量模型差异的损失函数是均方误差 (mean squared error, MSE):

$$\text{MSE} = \frac{1}{2N} \sum \|\mathbf{h}_{\text{teacher}} - \mathbf{h}_{\text{student}}\|^2, \quad (3.2)$$

¹还有一种基于关系的方法, 实际上是针对多个特征计算相似度矩阵再进行优化, 是基于隐藏层特征的一种拓展。

其中, $\mathbf{h}_{\text{teacher}}$ 和 $\mathbf{h}_{\text{student}}$ 分别为老师模型和学生模型的隐藏层表示, N 为样本总数。

3.3 本文方法: LST 训练

本节具体地介绍 LST 训练方法和上文语言知识的迁移。首先介绍语言模型的基本背景, 然后介绍 LST 方法。

3.3.1 语言模型

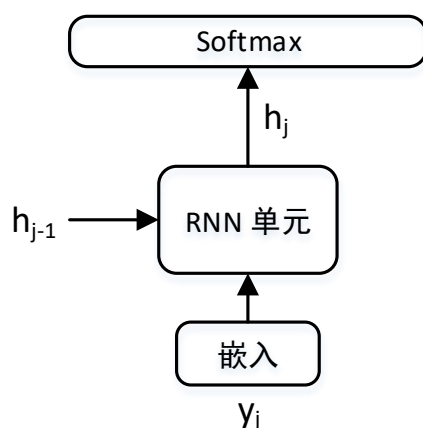


图 3.2 循环神经网络语言模型

Figure 3.2 recurrent neural network based language models

语言模型 [15, 116–118] 计算一句话发生的概率。基于概率的乘法法则, 语言模型也可表示为给定上文, 预测下一个词发生的概率:

$$P(y) = P(y_1) \prod_j^L P(y_j | y_{<j}), \quad (3.3)$$

与上一章相同, y 表示长度为 L 的一个词序列, y_j 表示第 j 个词, $y_{<j}$ 表示 y_j 的前缀。

以目前流行的循环神经网络语言模型 (recurrent neural network based language models, RNNLMs) 为例。如图 3.2 所示, 循环神经网络输入当前时刻词 y_j 对应的嵌入向量²和表示上文的隐藏表示向量 \mathbf{h}_{j-1} , 输出当前的隐藏表示 \mathbf{h}_j 。经过仿射变换和 softmax 函数, 就可以生成预测下一时刻词 y_{j+1} 的概率分布。同时, 表示上文的向量 \mathbf{h}_{j-1} 也更新为 \mathbf{h}_j 。整个计算过程依次迭代, 最终得到整个句子

²见第二章。

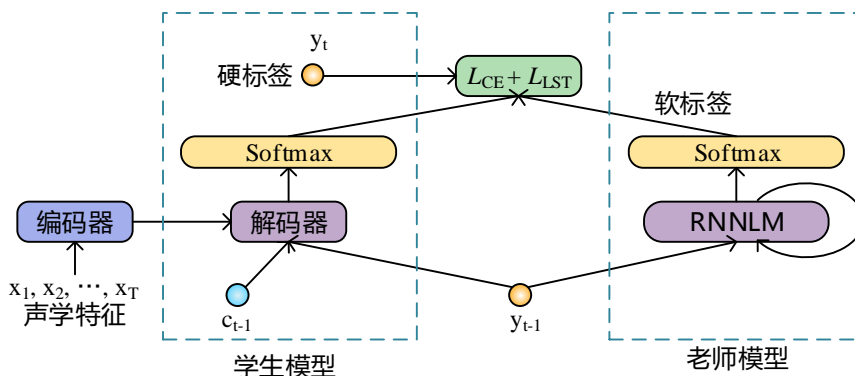


图 3.3 以循环神经网络为老师模型的 LST 方法

Figure 3.3 the LST method based on RNNLMs

所有的概率。图 3.2 中的 RNN 单元可以为长短时记忆 (long short-term memory, LSTM)[119], 门控循环单元 (gated recurrent unit, GRU)[120] 等有效防止梯度弥散和爆炸的形式。

神经网络语言模型的训练准则为极大似然估计。假设用 $f(y_{<j}, \theta)$ 表示以 θ 为参数的神经网络语言模型, 那么训练时利用纯文本语料最小化如下负对数似然函数:

$$\begin{aligned}
 -\log P_{LM}(y) &= -\sum_{j=2}^J \log P_{LM}(y_j|y_{<j}) \\
 &= -\sum_{j=2}^J f(y_{<j}, \theta).
 \end{aligned}
 \tag{3.4}$$

第2章介绍基于图结构的前馈神经网络也可以用来作为语言模型, 即抛去编码器部分, 只保留带掩蔽的解码器部分, 来计算整个句子的概率。

3.3.2 语言模型作为老师模型

我们以循环神经网络语言模型为例子, 介绍 LST 的具体实现, 如图 3.3所示。首先, 利用上一小节叙述的方法在纯文本语料上训练一个循环神经网络语言模型。然后, 在训练端到端语音识别模型的时候, 将声学特征对应的文本序列同时输入到循环神经网络语言模型中, 获得对应的软标签, 用此软标签来训练端到端语音识别模型。为了方便训练, 端到端语音识别和循环神经网络语言模型应共享相同的词表, 即同一个索引对应相同的词。

具体来说, 假设对于一个训练好的循环神经网络语言模型, 词表 \mathcal{V} 中第 k

| | | |
|-----|----------|---------------------------------|
| 0 | 0.0036 | 留 璃 檬 : 火 急 凉 |
| 0 | 2.50e-05 | |
| 0 | 2.68e-05 | |
| ⋮ | ⋮ | |
| 0 | 0.0062 | |
| 1 | 0.078 | |
| 0 | 0.0034 | |
| 硬标签 | 软标签 | 词表中对应的汉字 |

图 3.4 软标签与硬标签的比较。示意图中假设语言模型输入上文“不要着”。相比于硬标签，软标签可以提供不同词概率相对大小的信息。

Figure 3.4 A comparison of hard labels and soft labels. The example assumes that the prefix “不要着” is inputted into the LM. Compared with the hard labels, the soft labels provide information about the relativity of probabilities of different labels.

个词的概率由如下 softmax 函数计算：

$$\frac{\exp(o_k/T)}{\sum_{m=1}^{|\mathcal{V}|} \exp(o_m/T)}, \quad (3.5)$$

其中， o_k 是仿射变换后第 k 个元素， T 为控制概率分布平滑程度的温度系数。这个计算出来的概率分布即作为软标签训练语音识别模型。

图 3.4 给出一个软标签和硬标签的比较示意。假设将前缀“不要着”输入到语言模型中，那么下一个词最有可能的是“急”，而“火”或者“凉”也有较大的概率，但“璃”、“檬”等字的概率是非常小的，不大可能发生。对于这些情况，左边的硬标签是无法反映出的：除了“急”标记为 1，其它都为 0，没有区别。这个示意图说明软标签比硬标签包含了更丰富的知识，即暗知识。

LST 损失函数为如下 KL 散度：

$$L_{\text{LST}}(\theta) = \sum_{j=2}^J \sum_{y_j \in \mathcal{V}} P_{\text{LM}}(y_j | y_{<j}) \log \frac{P_{\text{LM}}(y_j | y_{<j})}{P_{\text{ASR}}(y_j | y_{<j}, \mathbf{X})}, \quad (3.6)$$

其中 J 为词序列长度， θ 为语音识别模型参数，与式 2.6 相同。

由于训练语音识别模型时，语言模型的参数是固定的，所以上述 KL 散度可

算法 1 LST 训练方法

- 1: 随机初始化端到端语音识别模型的参数.
 - 2: **while** 还未收敛 **do**
 - 3: 随机选择小批量语音-文本成对数据.
 - 4: 将声学特征 \mathbf{X} 输入到语音识别模型, 并前向传播.
 - 5: 将每一个文本序列 $y^{(m)}$ 输入到语言模型, 并前向传播. 依照式 3.5, 基于温度 T 计算软标签.
 - 6: 依照式 2.6 计算交叉熵损失 $L_{CE}(\theta)$.
 - 7: 依照式 3.7 计算 LST 损失 $L_{LST}(\theta)$.
 - 8: 依照式 3.8, 基于平衡系数 λ 组合两个损失.
 - 9: 反向传播, 更新参数 θ .
 - 10: **end while**
- return** θ
-

以简化为如下交叉熵损失:

$$L_{LST}(\theta) = - \sum_{j=2}^J \sum_{y_j \in \mathcal{V}} P_{LM}(y_j | y_{<j}) \log P_{ASR}(y_j | y_{<j}, \mathbf{X}). \quad (3.7)$$

从上面的式子可以看出, LST 损失 L_{LST} 的标签中不包含发音特征。所以, 我们还需要引入带发音特征的人工标注的知识, 即式 2.6 的交叉熵损失 L_{CE} 。综合二者, 我们得到最终的损失函数

$$L(\theta) = (1 - \lambda)L_{CE}(\theta) + \lambda L_{LST}(\theta), \quad (3.8)$$

其中 λ 为平衡系数。

算法 1 给出了 LST 方法的训练步骤。可以看出, 语言模型只在训练时利用, 在测试的时候是不需要使用的, 所以采用 LST 方法不增加测试阶段模型开销。

3.4 实验

3.4.1 实验数据

3.4.1.1 语音数据

本文采用开源的中文普通话数据 AISHELL-1³和 AISHELL-2 进行实验⁴[121, 122]。

³AISHELL-1 的下载地址为 <http://www.aishelltech.com/kysjcp>

⁴AISHELL-2 的申请地址为 http://www.aishelltech.com/aishell_2。

表 3.1 语音数据情况

Table 3.1 The description of speech data.

| | | 句子数 | 小时数 | 说话人数 |
|-----------|-----------------|---------|------|------|
| | 训练集 | 120098 | 150 | 340 |
| AISHELL-1 | 开发集 | 14326 | 18 | 40 |
| | 测试集 | 7176 | 10 | 20 |
| | 训练集 (iPhone) | 1009223 | 1000 | 1347 |
| | 开发集 (iPhone) | 2500 | 2 | 5 |
| | 开发集 (Android) | 2500 | 2 | 5 |
| AISHELL-2 | 开发集 (HiFi Mic.) | 2500 | 2 | 5 |
| | 测试集 (iPhone) | 5000 | 4 | 10 |
| | 测试集 (Android) | 5000 | 4 | 10 |
| | 测试集 (HiFi Mic.) | 5000 | 4 | 10 |

AISHELL-1 数据集包含 178 小时中文普通话语音。语音数据采用高保真 (high fidelity, HiFi) 麦克风录制为 44.1kHz 采样率的音频，之后降采样到 16kHz 并存储为位深度为 16 比特的 PCM 格式。语音由 400 个说话人录制，说话人的性别、年龄和出生地平衡分布。数据集的内容包金融，科技，体育，娱乐，和新闻 5 个领域。训练集包含 150 小时语音，验证集包含 18 小时语音，测试集包含 10 小时语音。

AISHELL-2 数据集包含 1000 小时中文普通话语音。所有的音频都存储为 16kHz 采样率，16 位深度的 PCM 格式。语音内容覆盖命令词、数字串、地名、娱乐、金融、科技、体育、英文拼写和自由朗读等。1000 小时训练数据由 1991 个说话人使用 iPhone 手机录制。开发集包括 5 个说话人录制的 2 小时语音，测试集包括 10 个说话人录制的 4 小时语音。开发集和测试集都由 iPhone 手机、Android 手机和高保真麦克风三种设备平行录制。

3.4.1.2 纯文本数据

本文抽取中文语言模型数据集 CLMAD[123–125] 的一个子集作为纯文本数据来训练语言模型。本文采用开源工具 XenC[126] 从全部 CLMAD 数据集中选

表 3.2 文本数据情况

Table 3.2 The description of text data.

| | | 句子数 | 字数 | 字节数 |
|-----------|---------|---------|-----------|---------|
| AISHELL-1 | 训练集标注文本 | 120098 | 1730113 | 5.1MB |
| | 开发集标注文本 | 14326 | 205341 | 0.8MB |
| | 测试集标注文本 | 7176 | 104765 | 0.4MB |
| | 纯文本 | 3703982 | 75893998 | 221.0MB |
| AISHELL-2 | 训练集标注文本 | 1009223 | 10995287 | 33MB |
| | 开发集标注文本 | 2500 | 24802 | 0.08MB |
| | 测试集标注文本 | 2500 | 104765 | 0.16MB |
| | 纯文本 | 7874474 | 130895577 | 381MB |

* AISHELL-2 开发集和测试集是 3 种设备平行录制，所以内容一样。

择了主题匹配的句子作为子集。预处理步骤如下：

1. 从 CLMAD 数据中选择 300 万条和语音识别训练集标注文本交叉熵差别较小的句子；
2. 移除太长的句子；
3. 将语音识别训练集标注文本和抽取出的句子混合；
4. 将文本逐字切分。

对于 AISHELL-1，本文混合了 10 份训练集标注文本，对于 AISHELL-2 本文训练了 5 份训练集合标注文本。最终获得的文本数据的情况见表 3.2⁵。

为了检查提取的纯文本数据是否能有效提升语言模型的性能，本文利用 N 元语法语言模型进行了测试。本文在训练集标注文本和纯文本上分别训练 3 元语法语言模型，然后在开发集上计算二者的困惑度 (perplexity, PPL)。结果如表 3.3 所示。可以看出，在纯文本数据上训练的 3 元语法语言模型，相比于在训练集标注文本上训练的，困惑度大大下降了。这说明外部纯文本与语音数据领域匹配，并可以显著地提升语言建模能力。

⁵我们开放了处理后的文本数据。AISHELL-1 的数据可以从 https://1drv.ms/u/s!An08U7hvUohBb234-V-Z0Qb_Zcc?e=fK02E0 下载，AISHELL-2 的数据可以从 <https://1drv.ms/u/s!An08U7hvUohBcznPgD5Io0AZlrU?e=UUhkDF> 下载

表 3.3 不同数据训练的 3 元语法语言模型在开发集标注文本上的困惑度

Table 3.3 Perplexities of trigram LMs trained on different training data on the development set.

| | 语言模型使用的数据 | 开发集困惑度 |
|-----------|-----------|--------|
| AISHELL-1 | 训练集标注文本 | 70 |
| | 纯文本 | 47 |
| AISHELL-2 | 训练集标注文本 | 78 |
| | 纯文本 | 62 |

3.4.2 实验设置

3.4.2.1 端到端语音识别模型

本文采用第2章介绍的基于图结构的前馈神经网络作为端到端语音识别模型。所采用的模型结构在 Vaswani 等 [70] 的 transformer 基础模型上略做改进以适应语音识别任务。模型的编码器和解码器各有 6 层 transformer 模块。模型的维度是 512 维，多头注意力机制的头数是 8，逐位置多层感知机的中间层维度是 2048，激活函数采用 GLU。本文采用 2 层卷积神经网络作为降采样，卷积层每层有 32 个滤波器，每个滤波器尺寸为 3×3 ，滤波器在时间轴移动的步幅为 2，所以帧率会被降采样到原始的 1/4。卷积神经网络的激活函数是 ReLU。

模型采用 80 维梅尔滤波器组特征 (Mel-filter bank, FBANK) 作为输入。特征的帧长为 25 毫秒，帧移 10 毫秒。词表为训练集中所有的汉字和 3 个特殊符号。“<unk>”表示未见的字，“<s>”表示句子开始，“<e>”表示句子结束。AISHELL-1 的词表大小为 4232，AISHELL-2 的词表大小为 5252。

本文采用一些训练技巧来避免过拟合，提高模型性能。采用概率为 0.1 的丢弃 (dropout) 策略。采用谱增强 (SpecAugment)[127] 策略作为数据增强策略，但是不使用时间弯折。谱增强技术将一些频率和时间范围内的元素用均值替换。频率替换宽度为 27，时间替换宽度为 40。频率替换和时间替换各进行 2 遍。

优化器为 Adam 优化器 [128]。学习率变化曲线采用如下方式更新：

$$\alpha = D^{-0.5} \cdot \min(\text{step}^{-0.5}, \text{step} \cdot \text{warmup}^{-1.5}), \quad (3.9)$$

其中 D 为模型维度 512， step 是优化步数， warmup 是热身步数。热身步数设为

20000。训练时，150 秒语音为一批进行训练。对于 AISHELL-1，模型训练 80 轮，对于 AISHELL-2，模型训练 40 轮。取最后 10 轮模型参数的平均作为最终模型。解码时，束搜索宽度为 5。

3.4.2.2 语言模型

本章采用两种语言模型：基于长短时记忆网络的循环神经网络语言模型，和基于图结构前馈神经网络的语言模型 [129]。两种语言模型和语音识别模型采用相同的词表。语言模型分别在训练集标注文本和纯文本上训练。

循环神经网络语言模型由两层长短时记忆网络构成，每层 1024 个单元。模型采用随机梯度下降 (stochastic gradient descent, SGD) 算法训练 20 轮。批大小设为 128。初始学习率 0.1。每一轮训练完毕后，在开发集上评估损失，如果损失相对下降不超过 10%，则将学习率折半。

基于图结构前馈神经网络的语言模型，维度为 512，多头注意力机制的头数为 8，逐位置多层感知机的中间层维度是 2048，一共 5 层。其依然使用 Adam 优化器训练，采用前述的热身训练的学习率曲线。热身步数为 16000。模型也训练 20 轮。

本章在每一轮训练以后，保存语言模型参数作为检查点，采取在开发集上困惑度最小的检查点作为最后使用的模型。

3.4.3 实验结果

本文采用针对中文的标准性能评测指标字错误率 (character error rate, CER) 来评价语音识别模型的性能。字错误率在计算时，首先通过编辑距离对齐算法，将识别结果和标准答案进行对齐，然后计算

$$\text{CER} = \frac{S + D + I}{N}, \quad (3.10)$$

其中， S 是指替换错误字数， D 是指删除错误字数， I 是指插入错误字数， N 是标准答案的字数。

本文首先在较小的数据集 AISHELL-1 上实验，比较语言模型的困惑度，然后在较小的数据集 AISHELL-1 上选择 LST 方法超参数 T 和 λ (见式 3.5 和式 3.8)，最后与其它方法进行比较，证明所提方法的有效性。之后，再将实验扩展到 1000 小时数据 AISHELL-2。

表 3.4 AISHELL-1: 困惑度测试

Table 3.4 AISHELL-1: Perplexity

| 语言模型 | 困惑度 | | 参数量 ⁶ |
|-----------|---------|---------|------------------|
| | 开发集标注文本 | 测试集标注文本 | |
| LSTM-IN | 59.82 | 57.09 | 25.5M |
| LSTM-EXT | 35.12 | 33.84 | 25.5M |
| TRANS-IN | 55.77 | 52.96 | 27.9M |
| TRANS-EXT | 28.50 | 27.66 | 27.9M |

3.4.3.1 AISHELL-1 实验

在 AISHELL-1 上, 本文首先测试语言模型的困惑度来证明语言模型的效果, 然后选择 LST 方法的超参数, 最后与其它方法进行比较, 证明所提方法的有效性。

困惑度测试。表 3.4 展示了语言模型在开发集和测试集上的困惑度, 其中 LSTM 表示基于长短时记忆的循环神经网络语言模型, TRANS 表示基于图结构的前馈神经网络的语言模型, IN 表示在训练集标注文本训练, EXT 表示在纯文本数据训练。首先, 我们可以看出相比表 3.3 中 3 元语法语言模型的结果, 神经网络语言模型的困惑度大大降低, 这表明了神经网络语言模型的有效性。第二, 我们可以看出, 使用大规模的纯文本数据可以显著地降低困惑度, 比如对于循环神经网络语言模型, 开发集上的困惑度从 59.82 降低到 35.12。其它几个结果也有一致的表现。这是因为大规模纯文本里面有更丰富的句子, 可以提升语言模型的泛化能力。第三, 基于图结构的前馈神经网络语言模型相对循环神经网络语言模型效果更好一点, 这表现出其更好的语言建模能力。总的来说, 这组困惑度实验证明了语言模型的有效性, 并证明了使用额外的外部纯文本的有效性。

超参数选择。我们在 AISHELL-1 的开发集上选择超参数。为了节省时间, 每一个模型训练 50 轮, 选择最后一个检查点来测试超参数。本文测试了典型的超参数值, 对 λ 取 0.1、0.2、0.5, 对 T 取 1.0、2.0、5.0, 所以我们对每一种模型比较 9 种情况。超参数 λ 控制 L_{LST} 的比例, T 控制作为老师模型的语言模型输出

⁶M 表示百万。

表 3.5 AISHELL-1: 超参数选择

Table 3.5 AISHELL-1: Hyper-parameter Selection

| λ | T | 字错误率% | | | |
|-----------|-----|------------|------------|------------|------------|
| | | LSTM-IN | LSTM-EXT | TRANS-IN | TRANS-EXT |
| 0.1 | 1.0 | 9.2 | 9.2 | 10.4 | 8.9 |
| 0.1 | 2.0 | 7.8 | 8.1 | 7.9 | 7.6 |
| 0.1 | 5.0 | 7.8 | 7.9 | 7.4 | 7.7 |
| 0.2 | 1.0 | 12.5 | 11.3 | 14 | 10.3 |
| 0.2 | 2.0 | 7.8 | 7.3 | 7.7 | 7.6 |
| 0.2 | 5.0 | 7.5 | 7.6 | 8.1 | 7.5 |
| 0.5 | 1.0 | 36.8 | 26.9 | 40 | 28.1 |
| 0.5 | 2.0 | 10.9 | 9.9 | 12.2 | 9.5 |
| 0.5 | 5.0 | 8.1 | 7.8 | 8.4 | 7.8 |

概率分布的平滑程度。表 3.5展示了实验结果。

可以看出,对于不同的老师模型具有不同的较优的超参数。训练语音识别模型时,较大的 λ 会引入更多的来自语言模型的知识,较大的 T 则会让语言模型输出的分布更为平滑。如果语言模型性能不佳,则会给训练过程引入更多噪声,所以需要控制 λ 和 T 来控制扰动的大小。同时,如前文所述,语言模型不包含发音的信息,所以标注文本带来的信息依然是很重要的。接下来,作者选择最优超参数,即表 3.5中粗体对应的部分来进行接下来的实验。在后边的实验中,模型训练到 80 轮直至收敛,然后平均最后 10 个检查点的模型参数作为最终模型。

与其它方法的比较。表 3.6展示了比较的结果。其中标签平滑 (label smoothing) [130–132] 可以被看做为一种零元语法统计语言模型,也被列在表中。基于极大似然估计准则训练的基线系统,其字错误率为 7.6%。LST 方法可以显著地提升性能。其中,相比于采用标签平滑,其它几种语言模型采用数据驱动的方法动态估计,效果更佳。本文发现,并非困惑度越低的语言模型带来的效果越好。本文分析这有可能是因为不同的困惑度的模型存在不同的最优超参数。另外,老师模型和学生模型的模型结构差异也会对结果造成影响。

表 3.6 AISHELL-1: 测试集上的字错误率

Table 3.6 AISHELL-1: CERs on test sets

| | λ | T | 字错误率% | 参数量 |
|------------------------------|-----------|-----|------------|-------|
| KALDI (nnet3) * † ‡ | - | - | 8.6 | - |
| KALDI (chain) * † ‡ | - | - | 7.4 | - |
| LAS [133] | - | - | 10.5 | - |
| ESPnet (Transformer) † [134] | - | - | 6.7 | - |
| Fan et al. [135] | - | - | 6.7 | - |
| An et al. † [136] | - | - | 6.3 | - |
| 基线系统 | - | - | 7.6 | 67.5M |
| + 标签平滑 | 0.1 | - | 6.7 | 67.5M |
| +LSTM-IN 作为老师 | 0.2 | 5 | 6.5 | 67.5M |
| +LSTM-EXT 作为老师 | 0.2 | 2 | 6.3 | 67.5M |
| +TRANS-IN 作为老师 | 0.1 | 5 | 6.5 | 67.5M |
| +TRANS-EXT 作为老师 | 0.2 | 5 | 6.4 | 67.5M |

* KALDI 官方记录中的结果。

† 使用了速度扰动作为数据增强。

‡ 使用了基于 I-Vector 的说话人自适应。

我们从两方面分析 LST 方法的影响。首先， L_{LST} 引入了标签的不确定性。具体来说，当语言模型是在训练集标注文本上训练的时候，它的知识和标注文本是同源的，此时 LST 没有引入外部知识。但是，软标签是一种期望形式下的正则化。其次，当语言模型是在纯文本上训练的时候，其知识源与标注文本不同。此时，它不仅引入了不确定性，还将外部知识引入到了模型中。

LST 与浅融合的组合。如前所述，本文所提 LST 方法是训练阶段的方法，在测试阶段不增加计算代价。实际上，在测试阶段还可以进一步地与浅融合结合。为了更好地理解 LST 方法，本文进一步地使用浅融合技术进行解码。具体上，在束搜索的时候，同时综合解码器和语言模型所计算的分数(见式 2.20)。本文使用困惑度最低的 TRANS-EXT 作为语言模型进行浅融合，语言模型插值系数设为 0.1。表 3.7 展示了浅融合的结果。可以看出，采用浅融合方法可以提升语音识别

表 3.7 AISHELL-1: LST 与浅融合的组合

Table 3.7 AISHELL-1: Combining LST and shallow fusion.

| 模型 | CER% | 参数量 |
|----------------------|------------|---------------|
| 基线系统 | 7.6 | 67.5M |
| 基线系统 + 浅融合 | 6.4 | 67.5M + 27.9M |
| 标签平滑 | 6.7 | 67.5M |
| 标签平滑 + 浅融合 | 5.9 | 67.5M + 27.9M |
| LSTM-EXT 作为老师 | 6.3 | 67.5M |
| LSTM-EXT 作为老师 + 浅融合 | 7.2 | 67.5M + 27.9M |
| TRANS-EXT 作为老师 | 6.4 | 67.5M |
| TRANS-EXT 作为老师 + 浅融合 | 5.8 | 67.5M + 27.9M |

的效果，然而浅融合方法增加了测试阶段的计算代价和模型尺寸。同时，本文还发现了一些现象。首先，“基线系统 + 浅融合”的效果和“TRANS-EXT 作为老师”的效果相同。这可以解释为浅融合是一种测试阶段集成方法，而老师-学生学习是一种训练阶段集成方法 [109, 137]，在这里，它们表现出了相同的效果。一个奇怪的现象是浅融合并没有提升“LSTM-EXT 作为老师”的结果。这可能是由于浅融合所用语言模型和 LST 训练语言模型之间的差异所导致的。

3.4.3.2 AISHELL-2 实验

接下来，本文将实验扩展到 1000 小时 AISHELL-2 上。此组实验采用的模型结构和前面 AISHELL-1 实验中的相同。模型参数量的差异是由于不同的词表。

困惑度测试。表 3.8 展示了 AISHELL-2 上各语言模型的困惑度。与 AISHELL-1 实验中的现象一致，大规模的纯文本数据显著地降低了困惑度。此组实验证明了语言模型的有效性，并证明了引入额外纯文本的有效性。

与其它方法的比较。本组实验直接使用了前面 AISHELL-1 实验中选择的超参数。本文在 AISHELL-2 1000 小时数据上训练 40 轮，平均了最后 10 个检查点。实验结果展示在表 3.9 中。与 AISHELL-1 中的结果相同，利用 LST 训练，语音识别模型的识别性能提升了。与基线系统相比，LST 带来了约 8% 的相对字错误

⁷M 表示百万。

表 3.8 AISHELL-2: 困惑度测试

Table 3.8 AISHELL-2: Perplexity

| 语言模型 | 困惑度 | | 参数量 ⁷ |
|-----------|---------|---------|------------------|
| | 开发集标注文本 | 测试集标注文本 | |
| LSTM-IN | 62.58 | 64.96 | 25.6M |
| LSTM-EXT | 47.99 | 49.83 | 25.6M |
| TRANS-IN | 56.18 | 59.45 | 29.0M |
| TRANS-EXT | 39.97 | 41.35 | 29.0M |

表 3.9 AISHELL-2: 测试集上的字错误率

Table 3.9 AISHELL-2: CERs on test sets

| 模型 | λ | T | 字错误率 | | |
|------------------------|-----------|-----|--------|---------|------|
| | | | iPhone | Android | HiFi |
| KALDI (chain) † [122] | - | - | 8.8 | 9.6 | 10.9 |
| LAS [133] | - | - | 9.2 | 9.7 | 10.3 |
| ESPnet (Transformer) * | - | - | 7.5 | 8.9 | 8.6 |
| 基线系统 | - | - | 7.1 | 8.0 | 8.2 |
| + 标签平滑 | 0.1 | - | 6.8 | 7.3 | 7.6 |
| + LSTM-IN 作为老师 | 0.2 | 5 | 6.8 | 7.2 | 7.7 |
| + LSTM-EXT 作为老师 | 0.2 | 2 | 6.5 | 7.0 | 7.4 |
| + TRANS-IN 作为老师 | 0.1 | 5 | 6.6 | 7.2 | 7.8 |
| + TRANS-EXT 作为老师 | 0.2 | 5 | 6.4 | 7.1 | 7.5 |

* ESPnet 官方报道的结果。

† 使用速度扰动作为数据增强。

率下降。

LST与浅融合的组合。同样地,本文比较了LST方法与浅融合的组合,表3.10展示了比较结果。可以看出,与AISHELL-1相同,LST方法表现出了和浅融合相同的集成效果。然而,进一步地同时使用LST和浅融合并没有带来性能提升,反

表 3.10 AISHELL-2: LST 与浅融合的组合

Table 3.10 AISHELL-2: Combining LST and shallow fusion.

| 模型 | 字错误率 | | | 参数量 |
|----------------------|--------|---------|------|---------------|
| | iPhone | Android | HiFi | |
| 基线系统 | 7.1 | 8.0 | 8.2 | 68.5M |
| 基线系统 + 浅融合 | 6.6 | 7.3 | 7.6 | 68.5M + 29.0M |
| 标签平滑 | 6.8 | 7.3 | 7.6 | 68.5M |
| 标签平滑 + 浅融合 | 5.9 | 6.7 | 7.1 | 68.5M + 29.0M |
| LSTM-EXT 作为老师 | 6.5 | 7.0 | 7.4 | 68.5M |
| LSTM-EXT 作为老师 + 浅融合 | 7.4 | 7.8 | 8.2 | 68.5M + 29.0M |
| TRANS-EXT 作为老师 | 6.4 | 7.1 | 7.5 | 68.5M |
| TRANS-EXT 作为老师 + 浅融合 | 6.5 | 7.0 | 7.3 | 68.5M + 29.0M |

而有了性能下降 (LSTM-EXT 作为老师 + 浅融合)。这可能是因为 LST 使用的语言模型和浅融合使用的语言模型之间的差异所致。

3.4.4 分析与讨论

本章前面的部分叙述了 LST 方法，并通过实验证明了其效果。本节进一步地分析 LST 方法的作用原理。

作者在实验的过程中观察到使用不同的损失函数训练得到的模型在语音识别的时候行为表现不同。具体来说，不同的模型在解码时得到的结果 (即搜索得到的最高分的假设文本) 的分数在数值上差异比较大。为了更好地理解使用不同损失训练出的模型的行为，本文绘制 AISHELL-1 测试集识别结果的概率的直方图 (histogram)。模型都是使用3.4节中的效果最好的超参数得到的模型。本节展示了“基线系统”、“基线系统 + 标签平滑”、“基线系统 +LSTM-EXT 作为老师”、“基线系统 +TRANS-EXT 作为老师”、“基线系统 + 浅融合”、“基线系统 + TRANS-EXT 作为老师 + 浅融合” 六个系统的识别结果的直方图。

图 3.5展示了结果。可以看出，使用极大似然估计准则训练的基线系统，直方图与其它几个结果完全不同。具体上，对于基线系统，大部分概率集中在 [0.9, 1.0] 这个区间。但是对于其它的结果，大部分分数都是集中在 [0, 0.1] 区间。这说明，

使用极大似然估计准则训练的模型的解码结果具有非常高的置信度 (confidence)。这种过置信 (overconfident) 一定程度上的是一种过拟合, 使可能存在于分数较低的正确结果没有办法被选中。使用 LST 训练则可以大大缓解这一问题。这些直方图部分程度地展示了模型估计的概率空间。由于计算所有可能的概率序列是不可能的, 这些直方图用识别结果的概率来部分地展示概率空间。可以想象出, 对于使用极大似然准则训练出的模型, 其估计的概率集中在某些样本上, 而 LST 训练则可以平滑概率空间。

图 3.5 的最后两个图展示了浅融合的分数的直方图。其计算如下:

$$\begin{aligned} \text{Score} &= \exp(\log P_{\text{ASR}}(y|\mathbf{X}) + \gamma \log P_{\text{LM}}(y)) \\ &= P_{\text{ASR}}(y|\mathbf{X})P_{\text{LM}}(y)^\gamma. \end{aligned} \tag{3.11}$$

所以这些分数并不能直接看成概率。从图中可以看出, 浅融合也平滑了分数的空间。

浅融合可以看做是测试阶段的一种集成。LST 则是训练阶段的集成。图 3.5 显示出二者具有类似的性质: 都平滑了分数空间。并且二者都可以提升识别的准确率。平滑模型估计的分数空间是 LST 和浅融合起作用的重要特性。

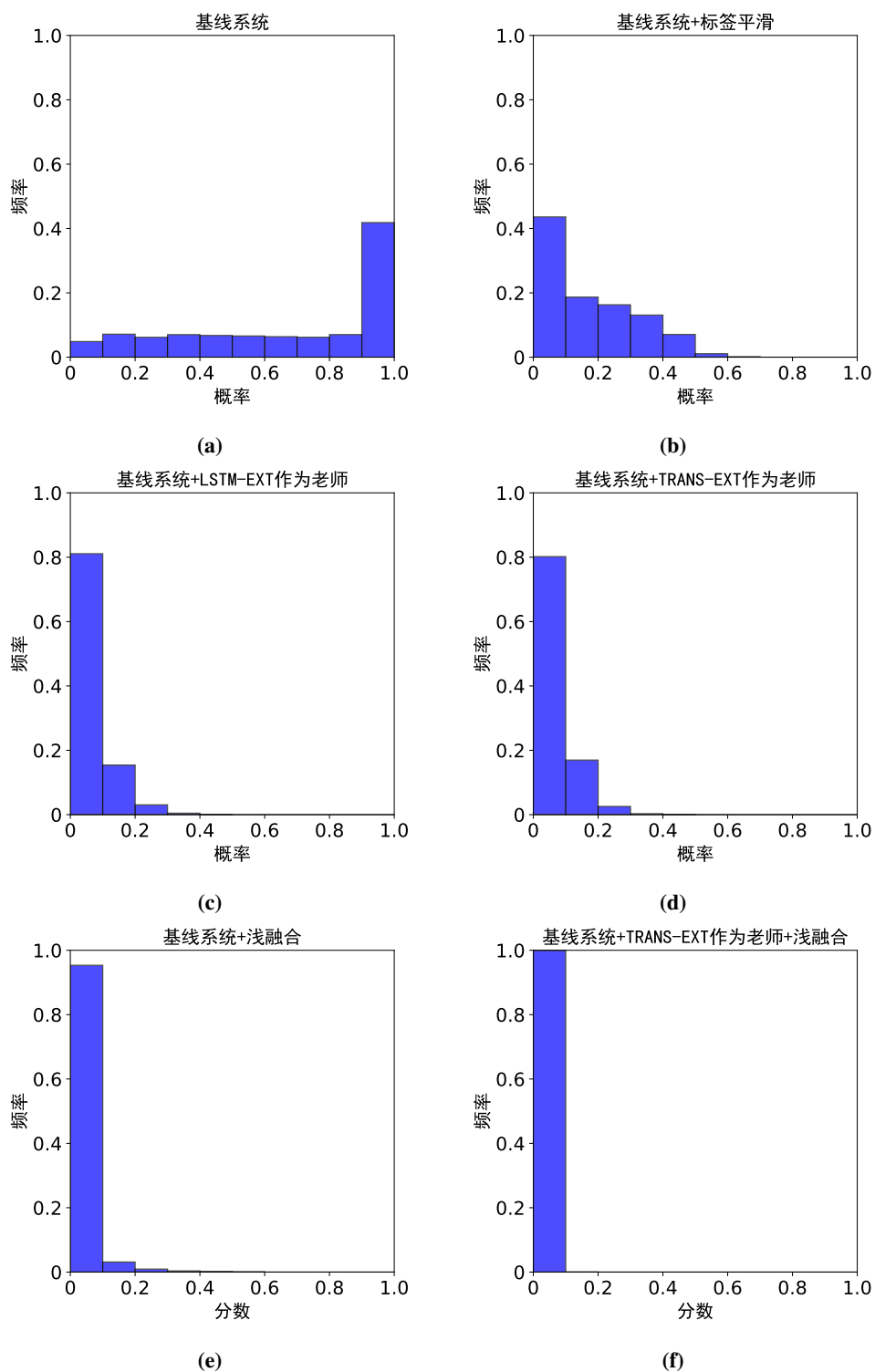


图 3.5 识别结果分数的直方图。前四个图模型输出的是概率。对于后两个图中的浅融合，模型输出的分数不可以看做概率。

Figure 3.5 Histograms of the scores. The models in the first four figures output probabilities. For shallow fusion in the last two figures, the outputs of the models cannot be seen as probabilities.

3.5 小结

针对端到端语音识别系统难以直接利用纯文本数据训练,所以难以利用纯文本知识的问题,本章提出了一种基于老师-学生学习的方法将纯文本数据中的语言知识迁移到端到端语音识别系统中。该方法首先在大规模纯文本数据上训练语言模型,将纯文本数据中的语言知识表示在语言模型中,然后利用老师-学生学习将此知识迁移到端到端语音识别系统。本章在小规模和大规模的中文公开数据集上验证了方法的效果,实验证明本章提出的方法可以提升语音识别的准确率,同时相比于原先的融合法,不增加测试阶段的计算代价。本章还分析了模型估计的分数空间的结构,发现本章所提 LST 方法和融合法都可以起到平滑模型分数空间的效果。这为更为细致地理论分析序列级集成方法的原理提供了材料。

本章所提的方法中,端到端语音识别模型和语言模型都是利用上文知识去预测下一个词。然而,整个的句子文本包含了全局上下文知识而不仅仅是上文。如何利用全局上下文知识就成了一个问题。在下一章中,本文提出一种利用全局上下文语言知识来提升语音识别效果的方法。

第4章 全局上下文语言知识迁移

4.1 引言

根据前面的介绍可以看出，基于注意力机制的编码器-解码器模型在生成文本的时候采用了一种自回归 (autoregressive) 的模式，即根据上文来预测下一个词。然而，有的时候，下文的信息对上文的预测有重要的影响。举例来说，对于句子“上地怎么走”，在预测“地”字的时候，只根据左边的“上”字，是很难判断正确的，因为“帝”也很有可能是一个正确的答案。只有当句子的最后一个字“走”出现了以后，我们才能判断出词语“上地”应该是一个地名，第二个字应该是“地”字。如果只根据上文来预测下一个字，文本生成过程中的一些小错误可能累积到后边，在束搜索时将正确答案排除在候选之外，影响最后识别结果的正确性。这种在左至右生成过程中由于前面生成的错误影响了最终结果的现象被称为曝光偏差 (exposure bias)[138, 139]。

如果能够将下文信息利用起来，甚至全局上下文信息同时利用起来，则有可能缓解上述的问题。所以，如何在基于注意力机制的编码器-解码器模型上利用下文信息，就成了一个受人关注的问题。Mimura 等 [140] 提出了一种前向-后向算法，在识别时同时自左向右、自右向左解码。然而，这种方法需要三遍解码，在测试阶段的计算代价较高。Zhou 等 [141] 等提出了一种双向同步模型用于机器翻译，在解码的过程中同步交互地自左向右、自右向左解码。然而，双向的注意力机制使模型结构变得更为复杂。“双向一致” (bidirectional agreement) 方法 [142] 事先训练一个自右向左的生成序列的编码器-解码器模型，然后在训练左至右模型的时候，最小化左至右模型和右至左模型的差异。这种方法不增加测试阶段的复杂度，提升了机器翻译 [142, 143] 和语音合成 [139] 端到端模型的性能。然而，这种方法优化的是左至右模型和右至左模型的差异，单独看这两种模型中的一个都并没有使用全局上下文。而且，右至左模型依然是一个端到端编码器-解码器模型，必须使用语音-文本平行数据训练，这影响了方法的灵活性。

针对这一问题，本章提出一种基于自注意力机制的全局上下文语言模型“因果完形填空器” (Causal cIOze completeR, COR)，将文本的全局上下文利用表示起来。再基于前文所提的 LST 方法，将因果完形填空器中的知识迁移到端到端语

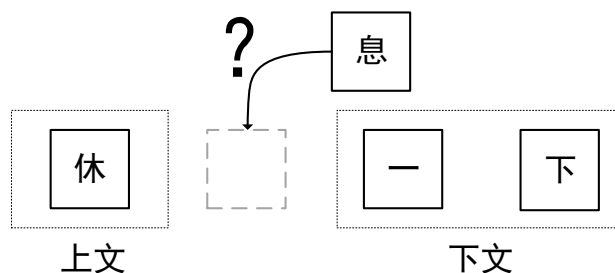


图 4.1 “完形填空”的示意图

Figure 4.1 an example of cloze

音识别模型。该方法是训练阶段的知识迁移方法，所以不增加测试阶段的计算代价；同时因果完形填空器是一个语言模型，所以在纯文本数据上进行训练即可，不需要使用语音-文本平行数据。实验证明，因果完形填空器和 LST 方法可以提升端到端编码器-解码器的语音识别性能。

在本章后续部分，第4.2节介绍因果完形填空器模型，第4.3节比较和其它全局语言模型的区别，第4.4节介绍实验，最后第4.5节小结本章内容。

4.2 基于完形填空的上下文全局语言建模

本节首先介绍完形填空问题，然后介绍因果完形填空器，最后介绍利用 LST 方法将上下文全局知识引入到端到端语音识别。

4.2.1 完形填空

受到大规模预训练语言模型 BERT[7] 的启发，本文引入完形填空游戏 (cloze) [144]。完形填空游戏是指将一个句子中的某一个或几个词替换为空白，让人根据空白左右的上下文将句子补全。完形填空最早被用来测试一个人的阅读能力，现在也被广泛地应用在英语考试中。由于在完形填空的过程中需要同时根据上下文来预测一个词，所以自然地想到使它作为一个全局语言建模方法。

图 4.1 给出了一个完形填空的例子。在这个例子中，我们需要同时根据上文 (“休”) 和下文 (“一下”) 去预测出空白处应该填入的 “息”。这一过程利用了文本的全局上下文。

形式化地，完形填空可以表述为如下过程。对于一个长度为 J 的文本序列 $y = (y_1, \dots, y_J)$ ，其中， y_j 表示在第 j 位置的词。如果给定了其中某一个位置的词 y_j 的上文 (y_1, \dots, y_{j-1}) 和下文 (y_{j+1}, \dots, y_J) ，我们需要估计概率

$P(y_j|y_1, \dots, y_{j-1}, y_{j+1}, \dots, y_J)$ 。

4.2.2 因果完形填空器

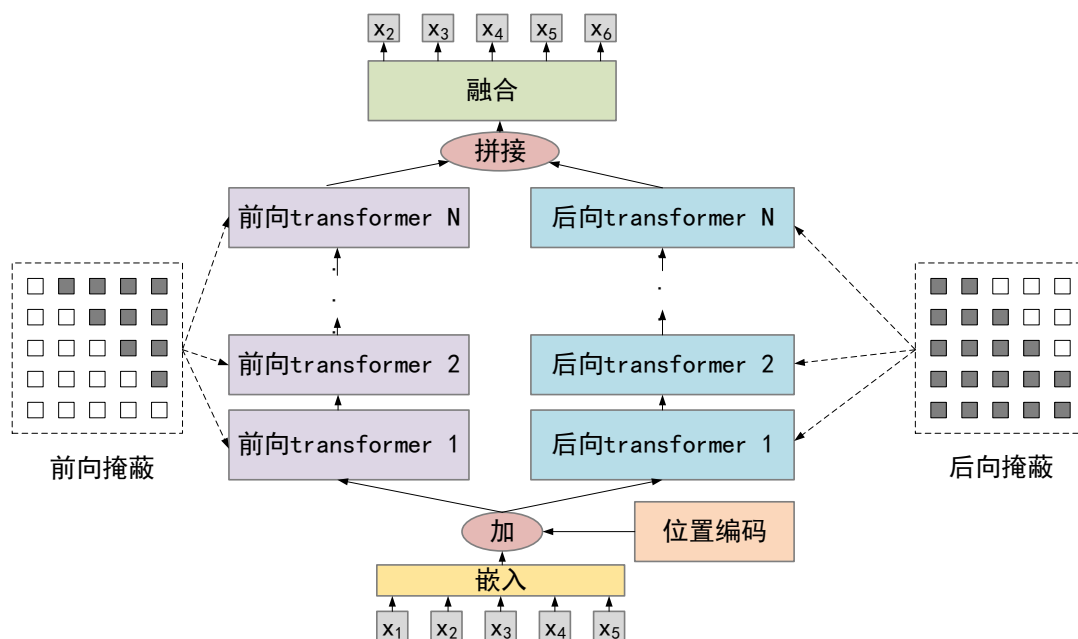


图 4.2 因果完形填空器的结构示意图

Figure 4.2 an illustration of COR

本章提出采用基于图结构的前馈神经网络来并行地估计每一个位置的概率 $P(y_j|y_1, \dots, y_{j-1}, y_{j+1}, \dots, y_J)$ 。自注意力机制可以直接捕捉序列中的长期依赖关系。

使用全局上下文建模的一个主要问题是，在预测一个词的时候，如何避免模型在输入序列里“看见”这个词。这会造成整个预测变为平凡的 (trivial)。因为根据自身预测自身，只需要一个恒等变换就可以了。BERT[7] 解决这一问题的方法是，用 [MASK] 符号随机替换一些词，在训练时预测这些 [MASK] 符号位置的词。但是这样做和我们的下游 LST 任务不匹配。举例来说，对于句子“休息一下”，如果直接把整个句子输入模型，那么概率就变成了 $P(\text{休}|\text{休}, \text{息}, \text{一}, \text{下})$ 。这是没有意义的。一种解决方法是对于句子中每一个词，都用 [MASK] 替换一次，一个一个的计算概率，但是这样需要模型分次多遍前向计算，无法并行地同时把所有概率算出来。

与 BERT 不同，本文不引入 [MASK] 符号，而是直接估计概率 $P(y_j|y_1, \dots, y_{j-1}, y_{j+1}, \dots, y_J)$ 。本文通过掩蔽掉一些注意力的分数 (见第2章)，直

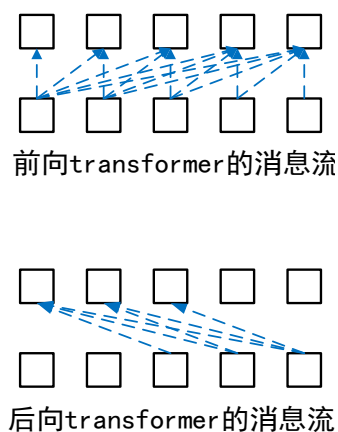


图 4.3 因果完形填空器的消息流

Figure 4.3 the context flow of COR

接控制上下文消息传播的路径。

图 4.2 展示了因果完形填空器的结构。因果完形填空器由前向部分和反向部分构成。每一个词得到了嵌入表示构成的序列以后，加上位置编码，分别传入前向网络和反向网络。对于一个词，前向网络根据上文给出这个词的表示，后向网络根据下文给出这个词的表示。最后，在最后一层，将两个表示拼接起来，送入最后的融合多层感知器，通过 softmax 函数给出对应位置词表上概率分布。

另一个问题是编码器-解码器模型是自回归的，所以输入和输出之间会错位。但是原先的双向语言模型 [6, 7, 145, 146] 的输入和输出的位置是相同的，与编码器-解码器的自回归性不匹配，不好直接用在 LST 方法中。为了解决这一问题，本文提出模仿编码器-解码器模型的因果性 (causality)，在输入和输出之间添加一个偏移量 (offset)。

每一个 transformer 模块都是由多头自注意力机制和逐位置多层感知机构成，与第2章介绍的一致，这里不再赘述。因果完形填空器的主要区别在于掩码的构成。图 4.2展示了前向掩蔽和后向掩蔽。考虑句子 (y_1, \dots, y_J) ，对于前向网络，掩码控制一个词只能“看见”它左边的词，所以右边的词和它本身 (从 j 到 $J - 1$ 位置) 的注意力分数掩蔽为 $-\infty$ ，这样经过 softmax 函数以后掩蔽部分就会变为 0。对于后向网络，掩蔽则控制一个词只能“看见”它右边的词，所以左边的词和它本身 (1 到 j 位置) 注意力分数掩蔽为 $-\infty$ 。注意到输出序列和输入序列中间有一个偏移量，所以注意力掩蔽矩阵也会有一个偏移量。这会造成后向网络的掩

蔽矩阵存在一些全部为 0 的行，使 softmax 函数无法进行。在这种情况下，本文在 softmax 之后将这些行掩蔽为 0。图 4.3 展示了受控制的消息流传播路线。

整个模型通过极大似然估计优化，经过优化以后，因果完形填空器就可以给出给定上下文，每一个位置的词的概率分布：

$$P_{\text{COR}}(y_j | y_1, \dots, y_{j-1}, y_{j+1}, \dots, y_J) = \text{COR}(y).j = 2, \dots, J \quad (4.1)$$

4.2.3 将全局上下文知识迁移到端到端语音识别

基于第3章介绍的 LST 方法，本文利用老师-学生学习来将因果完形填空器中的全局上下文知识迁移到端到端语音识别系统。具体上，首先在文本上训练因果完形填空器，然后再用其作为老师模型训练端到端语音识别。具体的步骤为算法 1。修改式 3.7 并综合式 3.8，我们可以得到如下的将全局上下文知识迁移到端到端语音识别的公式：

$$L_{\text{LST}}(\theta) = - \sum_{j=2}^J \sum_{y_j \in \mathbb{V}} P_{\text{COR}}(y_j | y_1, \dots, y_{j-1}, y_{j+1}, \dots, y_J)(y_j | y_{<j}) \log P_{\text{ASR}}(y_j | y_{<j}, \mathbf{X}),$$

$$L(\theta) = (1 - \lambda)L_{\text{CE}}(\theta) + \lambda L_{\text{LST}}(\theta). \quad (4.2)$$

可以看出，利用因果完形填空器和 LST 训练，我们在训练阶段将全局上下文语言知识迁移到了端到端语音识别系统。该方法在测试阶段不增加系统的计算代价。同时该方法可以灵活地使用纯文本，而不必限制语音-文本平行数据。

4.3 相关工作

本节比较所提因果完形填空器和其它全局(整句)建模语言模型。

最早的全局建模的语言模型是最大熵 (maximum entropy) 语言模型 [147–149]。这种方法利用指数函数构建势函数来整体优化句子发生的概率，而不需要利用链式法则拆分概率。近期，变维随机场 (trans-dimensional random fields)[150] 针对不同长度的句子用不同的归一化因子来优化势函数，得到整句的概率。最大熵语言模型和变维随机场语言模型由于不能给出每一个词的概率，一般用在语音识别结果的重打分上。双向循环神经网络也被用于双向语言模型 [151–153]，然而，这些工作由于在预测一个词的时候模型会“看见”这个词本身，所以只能计算伪似然 (pseudo-likelihoods) 而不是真正的似然 [151]。同样地，这些基于双

向循环神经网络的语言模型一般用于重打分。近期，在自然语言处理领域，一些基于双向循环神经网络 [6, 145] 或者图结构前馈神经网络 [7, 63, 154] 双向语言模型提升了下游自然语言处理任务的效果。因果完形填空器也是基于图结构前馈神经网络的，然而，其不同于使用 [MASK] 词的类 BERT 的网络，而是直接建模两边的上下文，减少了 LST 任务和语言模型任务的不匹配。[146] 也使用了双塔结构，但是其不是用于文本生成任务，所以网络不具有因果性。本文所提因果完形填空器则模仿因果性，在输入和输出添加了偏移量。

4.4 实验

4.4.1 实验数据

本章沿用和第3章相同的数据集：语音数据 AISHELL-1 和 AISHELL-2，以及从 CLMAD 中提取的纯文本数据。具体的数据描述和文本数据处理与第3.4.1小节一致。

4.4.2 实验设置

4.4.2.1 端到端语音识别模型

本章采用的基线端到端语音识别模型，编码器和解码器都有 6 层 transformer 模块。模型维度是 512，多头注意力机制头数是 8，逐位置多层感知机中间层维度 2048，激活函数为 GLU。采用 2 层卷积层作为降采样，每层 32 个滤波器，尺寸均为 3×3 ，时间轴移动步幅为 2，帧率为原始 1/4。卷积层激活函数是 ReLU。

端到端识别系统依然采用 80 维梅尔滤波器组特征作为输入，帧长 25 毫秒，帧移 10 毫秒。词表为训练集所有汉字和特殊符号 “<unk>”，“<s>” 和 “<e>”，AISHELL-1 词表为 4232，AISHELL-2 词表为 5252。训练中采用概率为 0.1 的丢弃 (dropout) 策略。采用谱增强 (SpecAugment)[127] 策略作为数据增强策略，但是不使用时间弯折，频率替换宽度为 27，时间替换宽度为 40。频率替换和时间替换各进行 2 遍。学习率变化曲线依照式 3.9，热身步数为 20000。训练时 150 秒语音为一批，AISHELL-1 训练 80 轮，AISHELL-2 训练 40 轮，取最后 10 轮模型参数的平均作为最终模型，解码时束搜索宽度为 5。

4.4.2.2 因果完形填空器

对于因果完形填空器，模型的维度为 512，前向 transformer 模块和后向 transformer 模块的个数都是 5。每一个模块有 8 个注意力机制的头，逐位置多层感知机的维度为 2048，使用 GLU 作为激活函数。训练时，采用 Adam 优化器 [128] 和式 3.9 的学习率曲线，热身步数为 16000 步。

本章延续上一章的配置，对比基于长短时记忆网络的循环神经网络语言模型和基于图结构前馈神经网络的语言模型两种语言模型。这两种模型的配置和第3章相同。所有模型都训练 20 轮，取在开发集上损失最低的检查点作为最后的损失。

4.4.2.3 语言模型的评价

双向语言模型无法计算困惑度 [151]。本章引入完形填空正确率来评价完形填空器的效果。具体上，计算下式：

$$ACC = \frac{\#Corr}{\#Total}. \quad (4.3)$$

其中，ACC 表示完形填空正确率， $\#Corr$ 为模型预测正确的词的数量， $\#Total$ 为测试集上总的词的数量。

4.4.3 实验结果

沿用第3章的命名规则，本章分别用 LSTM，TRANS，和 COR 分别表示基于长短时记忆记忆的循环神经网络语言模型，基于图前馈神经网络的语言模型，和本文所提的因果完形填空器。IN 表示使用训练集标注文本训练，EXT 表示使用纯文本训练。本章还是在 AISHELL-1 上选择 LST 的超参数，然后再将实验扩展到大规模数据集 AISHELL-2 上。

表 4.1 AISHELL-1: 完形填空正确率

Table 4.1 AISHELL-1: The accuracy of cloze.

| 语言模型 | 完形填空正确率 | | 参数量 |
|-----------|---------|---------|-------|
| | 开发集标注文本 | 测试集标注文本 | |
| LSTM-IN | 0.31 | 0.32 | 25.5M |
| LSTM-EXT | 0.36 | 0.37 | 25.5M |
| TRANS-IN | 0.32 | 0.33 | 27.9M |
| TRANS-EXT | 0.39 | 0.40 | 27.9M |
| COR-IN | 0.51 | 0.52 | 32.6M |
| COR-EXT | 0.63 | 0.65 | 32.6M |

表 4.2 AISHELL-1: 超参数选择

Table 4.2 AISHELL-1: Hyper-parameter Selection

| λ | T | 字错误率% | |
|-----------|-----|------------|------------|
| | | COR-IN | COR-EXT |
| 0.1 | 1.0 | 8.7 | 8.7 |
| 0.1 | 2.0 | 7.8 | 7.7 |
| 0.1 | 5.0 | 7.8 | 7.7 |
| 0.2 | 1.0 | 8.9 | 8.6 |
| 0.2 | 2.0 | 7.5 | 7.4 |
| 0.2 | 5.0 | 7.6 | 7.6 |
| 0.5 | 1.0 | 14.4 | 10.6 |
| 0.5 | 2.0 | 8.0 | 7.3 |
| 0.5 | 5.0 | 7.6 | 7.5 |

表 4.3 AISHELL-1: 测试集上的字错误率

Table 4.3 AISHELL-1: CERs on test sets

| | λ | T | 字错误率% | 参数量 |
|------------------------------|-----------|-----|------------|-------|
| KALDI (nnet3) * † ‡ | - | - | 8.6 | - |
| KALDI (chain) * † ‡ | - | - | 7.4 | - |
| LAS [133] | - | - | 10.5 | - |
| ESPnet (Transformer) † [134] | - | - | 6.7 | - |
| Fan et al. [135] | - | - | 6.7 | - |
| An et al. † [136] | - | - | 6.3 | - |
| 基线系统 | - | - | 7.6 | 67.5M |
| + 标签平滑 | 0.1 | - | 6.7 | 67.5M |
| +LSTM-IN 作为老师 | 0.2 | 5 | 6.5 | 67.5M |
| +LSTM-EXT 作为老师 | 0.2 | 2 | 6.3 | 67.5M |
| +TRANS-IN 作为老师 | 0.1 | 5 | 6.5 | 67.5M |
| +TRANS-EXT 作为老师 | 0.2 | 5 | 6.4 | 67.5M |
| +COR-IN 作为老师 | 0.2 | 2 | 6.2 | 67.5M |
| +COR-EXT 作为老师 | 0.5 | 5 | 5.8 | 67.5M |

* KALDI 官方记录中的结果。

† 使用了速度扰动作为数据增强。

‡ 使用了基于 I-Vector 的说话人自适应。

4.4.3.1 AISHELL-1 上的实验

完形填空正确率。表 4.1展示了语言模型的完形填空正确率。可以看出，循环神经网络语言模型和基于图前馈神经网络的语言模型都是单向语言模型，即根据上文去预测下一个词，这样限制了完形填空的性能。而因果完形填空器则同时利用上文和下文去预测一个词，完形填空的正确率相对提高较大。利用外部纯文本数据，则进一步地提高了完形填空的正确率。

LST 超参数选择。本章在 AISHELL-1 开发集上选择超参数。与第3相同，为了节省时间，每一个模型训练 50 轮，选择最后一个检查点测试超参数。本文测试了典型的超参数值，对 λ 取 0.1、0.2、0.5，对 T 取 1.0、2.0、5.0，所以我们对

每一种模型比较 9 种情况。超参数 λ 控制 L_{LST} 的比例， T 控制作为老师模型的语言模型输出概率分布的平滑程度 (见式 3.5 和章 4)。式 3.8 展示了实验结果。相对于单向语言模型，因果完形填空器的分类准确率更高，输出的概率置信度更高，噪声更少，所以其 LST 损失的比例更大一点，温度 T 更低一点。

语音识别词错误率。表 4.3 展示了与其它方法的比较。可以看出，利用因果完形填空器作为老师模型训练的语音识别系统，效果最优。特别是，以外部纯文本数据作为老师训练的 COR-EXT 作为老师，得到了最好的效果，其字错误率为 5.8%。

4.4.3.2 AISHELL-2 上的实验

本节将实验扩展到 1000 小时 AISHELL-2 数据集。模型采用与前一节 AISHELL-1 相同的结构。

完形填空正确率。表 4.4 展示了各语言模型在 AISHELL-2 数据集上完形填空正确率。可以看出，与 AISHELL-1 上的情形类似，因果完形填空器相对于其它两种语言模型获得了更高的正确率。这说明在更大规模的数据上因果完形填空器依然有效。

语音识别词错误率。由于训练 1000 小时模型较为耗时，本节直接选择前面 AISHELL-1 数据上选择的超参数进行实验。本节训练 40 轮模型，平均最后 10 个检查点参数。和前面 AISHELL-1 上的实验结果相同，使用因果完形填空器可以提升语音识别的效果。特别地，使用额外的纯文本数据训练的因果完形填空器作为老师模型，来训练端到端语音识别模型，获得了最低的字错误率。

表 4.4 AISHELL-2: 完形填空正确率

Table 4.4 AISHELL-2: The accuracy of cloze.

| 语言模型 | 完形填空正确率 | | 参数量 |
|-----------|---------|---------|-------|
| | 开发集标注文本 | 测试集标注文本 | |
| LSTM-IN | 0.30 | 0.30 | 27.6M |
| LSTM-EXT | 0.32 | 0.32 | 27.6M |
| TRANS-IN | 0.31 | 0.31 | 29.0M |
| TRANS-EXT | 0.35 | 0.34 | 29.0M |
| COR-IN | 0.51 | 0.50 | 33.7M |
| COR-EXT | 0.59 | 0.58 | 33.7M |

表 4.5 AISHELL-2: 测试集上的字错误率

Table 4.5 AISHELL-2: CERs on test sets

| | λ | T | 字错误率% | | |
|------------------------|-----------|-----|------------|------------|------------|
| | | | iPhone | Android | HiFi |
| KALDI (chain) † [122] | - | - | 8.8 | 9.6 | 10.9 |
| LAS [133] | - | - | 9.2 | 9.7 | 10.3 |
| ESPnet (Transformer) * | - | - | 7.5 | 8.9 | 8.6 |
| 基线系统 | - | - | 7.1 | 8.0 | 8.2 |
| + 标签平滑 | 0.1 | - | 6.8 | 7.3 | 7.6 |
| +LSTM-IN 作为老师 | 0.2 | 5 | 6.8 | 7.2 | 7.7 |
| +LSTM-EXT 作为老师 | 0.2 | 2 | 6.5 | 7.0 | 7.4 |
| +TRANS-IN 作为老师 | 0.1 | 5 | 6.6 | 7.2 | 7.8 |
| +TRANS-EXT 作为老师 | 0.2 | 5 | 6.4 | 7.1 | 7.5 |
| +COR-IN 作为老师 | 0.2 | 2 | 6.0 | 6.8 | 7.3 |
| +COR-EXT 作为老师 | 0.5 | 2 | 5.7 | 6.8 | 7.1 |

* ESPnet 官方记录中的结果。

† 使用了速度扰动作为数据增强。

本组实验的模型参数量约为 68.5M。

4.5 小结

针对端到端编码器-解码器语音识别系统在预测时只利用了上文而没有利用下文的问题，本章提出一种同时利用上文和下文来预测一个词的语言模型因果完形填空器，再基于本文所提出的 LST 训练方法将上下文全局语言知识迁移到编码器-解码器模型。本章所提的因果完形填空器基于图结构的前馈神经网络，可以并行地将给定上下文，每一个词的概率计算出来。本章在小规模数据集 AISHELL-1 和大规模数据集 AISHELL-2 上进行实验，结果表明利用所提因果完形填空器迁移全局上下文语言知识可以提升端到端编码器-解码器语音识别的准确率。

本章提出的方法将全局上下文语言知识迁移到编码器-解码器模型，然而编码器-解码器模型本身是条件化的语言模型，其对文本输入的依赖造成模型需要多遍前馈导致预测速度相对较慢。在下一章中，本文提出一种纯语音模态的语音识别模型，并进一步地，将产品规模的大规模预训练语言模型中的语言知识迁移到非自回归语音识别模型中，实现跨模态的知识迁移。

第 5 章 跨模态全局上下文语言知识迁移

5.1 引言

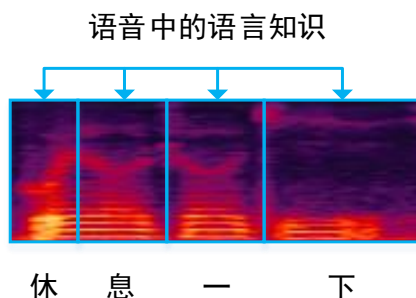


图 5.1 语音中蕴含的语言知识示意图

Figure 5.1 an illustration of language semantics

前面的两章中，本文提出利用 LST 训练方法将纯文本数据中的知识迁移到端到端语音识别系统，并进一步地提出了一种全局上下文语言模型来表示语言知识。这两种方法都着眼于自回归的基于注意力机制的编码器-解码器模型，即解码器根据预测出的上文和注意力机制提取的声学特征，去预测下一个词。整个解码过程是搜索概率最高的句子。模型中的解码器显式地估计句子发生的概率，建模词与词之间的关系，扮演了自回归语言模型的角色。

然而，我们可以观察到，语音信号上应该已经包含了这种词与词之间的关系。图 5.1 给出了一个例子，语谱图中的一个片段代表了其对应的汉字，片段与片段之间的关系对应于字与字之间的关系。也就是说，语音和其对应的文本某种意义上是同构的 (isomorphic)¹。这里我们将这种词与词 (字与字) 之间的关系称为语言语义 (language semantics)。

语音和文本中的语言语义同构给我们带来了两个问题。

1. 既然语音中也包含语言语义，那么能不能直接利用语音中的语言语义有效地进行语音识别，避免显式地对文本自回归语言建模？

2. 既然文本和语音的语言语义是同构的，那么能不能跨模态地将文本模型里的语言知识迁移到语音模型上，提升语音识别性能？

本章依次序回答上面提出的两个问题。

¹同构性 (isomorphism) 是指“具有相同的结构”，比如顺序等。

首先, 本章提出一种基于图结构前馈神经网络的非自回归语音识别模型 LASO (Listen Attentively, and Spell Once)[155]。该模型基于图结构构建语言语义的表示, 利用注意力机制自适应地将代表词的语音高层表示片段聚合 (aggregate) 起来, 并利用自注意力机制构建片段与片段之间的关系。识别时, 直接对每一个位置的词计算概率分布, 取概率最大的词, 不需要束搜索。实验证明, 不进行显式的语言建模, 直接利用语音中的语言语义进行语音识别并达到较好的准确率是可行的。实验还证明, 由于避免了束搜索中多次网络前馈计算, 基于 LASO 的语音识别系统的识别速度相对自回归的编码器-解码器模型大大提高。

其次, 本章提出一种跨模态全局上下文语言知识迁移方法, 进一步地提升 LASO 的语音识别准确率。该方法基于本文提出的 LST 训练, 将大规模双向语言模型 BERT[7] 中的全局上下文文本知识迁移到 LASO 中。实验证明, 跨模态地将文本模型里的语言知识迁移到端到端语音识别模型上可以提升语音识别的效果。这证实了我们的猜测: 利用文本和语音语言语义的同构性质提升语音识别效果是可行的。

在本章的后续部分, 第5.2 节介绍语音识别中的语言语义建模的相关方法作为背景, 并引出本文所提非自回归端到端语音识别模型 LASO 的动机; 第5.3 节介绍所提非自回归语音识别模型 LASO; 第5.4 节介绍跨模态语言知识迁移; 第5.5 节介绍和比较非自回归编码器-解码器模型和跨模态知识迁移的相关工作; 第5.6 节介绍实验; 最后, 第5.7 节小结本章。

5.2 语音识别中的语言语义

本章所称的语言语义, 即词之间的关系, 是语音识别系统所考虑的重要部分。语言语义保证语音识别出的句子除了要合乎发音以外, 还要合乎语义, 是一个合法的句子²。

传统上, 语言语义是显式地通过自回归语言模型建模的。比如, 基于噪声信道模型的语音识别系统 (混合模型 [15] 或基于 CTC 的模型 [52, 54] 等), 声学模

²这里的合法不仅是语法 (syntactic) 上的, 同时还要在语义上符合常识和说话人本身意图。

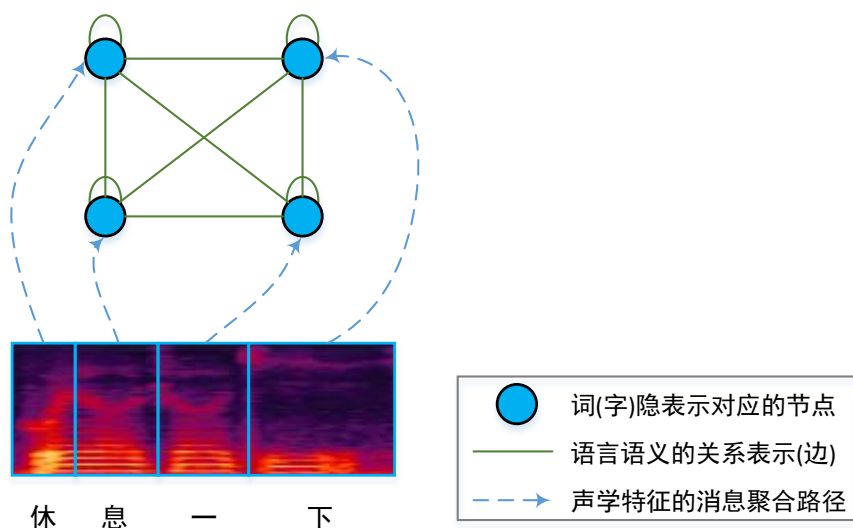


图 5.2 语言语义的无向图表示

Figure 5.2 an undirected graph to represent language semantics

型和语言模型分开建模：

$$\begin{aligned}
 P(y|\mathbf{X}) &= \frac{P(\mathbf{X}|y)P(y)}{P(\mathbf{X})}, \\
 &= \frac{P(\mathbf{X}|y)}{P(\mathbf{X})} P(y_1) \prod_{j=2}^J P(y_j|y_{<j}),
 \end{aligned} \tag{5.1}$$

其中 \mathbf{X} 表示声学特征序列， y 表示词序列， $y_{<j}$ 表示句子中第 j 个词 y_j 的前缀。 $P(y)$ 是一个自回归的语言模型（N 元语法或循环神经网络语言模型等）。声学模型和语言模型在识别阶段通过解码器结合。基于自注意力机制的编码器-解码器模型则为语音语言一体化建模 [56, 57]，直接估计概率

$$P(y|\mathbf{X}) = P(y_1|\mathbf{X}) \prod_{j=2}^J P(y_j|y_{<j}, \mathbf{X}). \tag{5.2}$$

模型中蕴含了显式地自回归语言模型。而对于转换器模型 [60–62]，则是通过神经网络来将声学模型和自回归语言模型融合：

$$P(y|\mathbf{X}) = \text{Fuse}(\text{Pred}(\mathbf{X}), \text{Trans}(y)), \tag{5.3}$$

其中，Pred 为建模声学特征序列的预测网络 (prediction network)，Trans 为建模语言的转录网络 (transcription network)，Fuse 为融合二者的神经网络³。

可以看出，上述 3 种经典的语音识别系统都是显式地建模语言语义。

³转换器模型中，声学序列和词序列的对齐关系是通过前向-后向算法枚举出所有可能的路径学习得到的。

本文中,我们希望采用无向图(undirected graph)来表示语言语义。举例来说,如图 5.2所示,语音中字对应的语音段聚合到图中的隐变量节点中,隐变量节点就对应于句子中的每一个字,字和字两两之间都可能存在关系,所以构成全连接无向图。整个图利用深度神经网络表示语言语义,即图 5.2 中的绿色边,通过神经网络训练得到。通过这样的方式,模型即可以捕捉到语言语义,预测每一个词了。

采用无向图表示语言语义的一个困难点在于,词对应的语音段十分多样,长度、发音、音色等都富于变化,那么如何自适应地聚合词对应的语音段?下一节中介绍所提 LASO 基于位置编码的注意力机制,自适应地聚合语音段。

5.3 基于图结构前馈神经网络的非自回归语音识别模型

本节具体介绍所提出的 LASO 模型。首先介绍问题的表示,然后介绍具体的模型结构、训练方法、识别方法。

5.3.1 语音识别作为逐位置的分类问题

传统上,语音识别是一个整句概率估计的问题:声学模型估计发音特征,语言模型估计词与词之间的转移概率,综合起来估计整句概率。本章给出一个语音识别问题的新视角。基本的想法是语言语义已经隐式包含在语音信号中,所以可以直接利用整句语音去预测一个词。于是,语音识别问题就可以考虑为一个逐位置的分类问题。当每一个位置的词都预测出来,拼接起来以后就是整个一句话。形式化地,模型估计如下概率

$$P(y_j|\mathbf{X}) = f(\mathbf{X}), j = 1, \dots, L, \quad (5.4)$$

其中, \mathbf{X} 是声学特征序列, y_j 是第 j 个位置的词, L 是词序列长度, f 是一个函数。本章采用基于图结构的前馈神经网络作为 f 。通常来说词序列长度 L 提前是不知道的。本章用一个简单的方法来处理这个问题,即设定 L 是一个足够大的固定的长度,结尾全部用填充词填充,在测试阶段简单地去掉结果中的填充词即可。

5.3.2 模型

图 5.3展示了 LASO 的模型结构。LASO 由 3 部分组成:编码器(encoder)、位置相关总结器(position dependent summarizer, PDS)和解码器(decoder)。编码

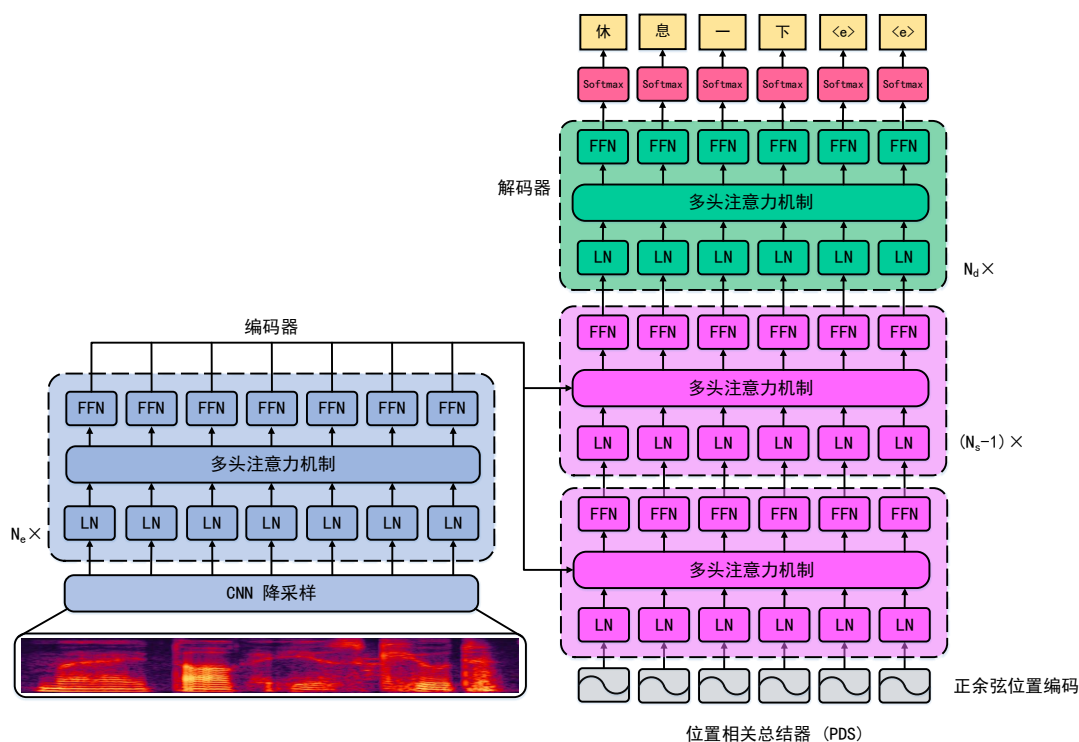


图 5.3 LASO 的模型结构

Figure 5.3 the architecture of LASO

器提取高层级的声学序列表示；位置相关总结器自适应地聚合词对应的语音段，将声学特征序列转换到对应于词的隐藏表示；解码器扮演了语言模型的角色，捕捉词级别隐藏表示之间的关系，即语言语义。在最后，通过一个 softmax 函数逐位置地计算词表的概率分布。在识别时，将声学特征输入模型以后，在每一个位置取概率最大的词即可。在处理长度不够 L 的句子时，在尾部用 $\langle e \rangle$ 符号补齐；在识别时，结果中的 $\langle e \rangle$ 符号简单地删除即可。整个模型由 transformer 模块构成 (见 2.2.5.2 小节)。

编码器。编码器从声学特征序列中提取高层表示。与第 2 中所述编码器-解码器模型类似，编码器首先利用两层卷积神经网络降采样，然后利用 N_e 层自注意力机制模块提取高层表示，即注意力机制的查询向量序列、键向量序列和值向量序列都是前一层的输出。

位置相关总结器。位置相关总结器 PDS 是 LASO 模型的核心模块。它根据位置编码，自适应地聚合词对应语音段的声学特征。具体地，位置相关总结器是 N_s 层注意力机制模块，其第一层的查询向量是位置编码，键向量和值向量是编码器的输出的高层声学表示，后边几层的查询向量是前一层输出，键向量和

值向量依然是编码器输出。可以看出，无论编码器输出的长度是多少，位置相关总结器输出的序列长度总是和位置编码序列长度一样。位置编码的长度 L 是通过统计训练集长度得到的。一般可以设置为最大长度加一些余量，如训练集最大长度为 90 个词，那么 L 可以设置为 100。本章中位置编码采用正余弦函数，如式 2.14。正余弦函数的一个优势在于，它有利于模型学习到相对位置。因为两个位置的距离可以表示为两个位置编码线性组合 [70]。

解码器。解码器进一步地提炼语言语义。与编码器相同，它由数层自注意力模块构成，查询向量序列、键向量序列和值向量序列都是前一层的输出。它通过自注意力机制计算位置相关总结器输出的隐表示之间的关系，并传播到下一层。可以看出，与以前的编码器-解码器模型不同，此解码器使用了全局的上下文。解码器后边是逐位置的线性变换和 softmax 函数，在每一个位置计算词表上词的概率分布。

形式化描述。形式化地，LASO 可以表述如下

$$\begin{aligned} \mathbf{Z} &= \text{Enc}(\mathbf{X}), \\ \mathbf{q}_i &= \text{Summarize}(\mathbf{Z}, \mathbf{P}_i), \quad i = 1, 2, \dots, L, \\ \mathbf{Q} &= [\mathbf{q}_1, \dots, \mathbf{q}_L] \\ P(y_j|\mathbf{X}) &= \text{Dec}(\mathbf{Q}), \quad i = 1, 2, \dots, L, \end{aligned} \tag{5.5}$$

其中， \mathbf{X} 是声学特征序列， \mathbf{Z} 为编码器输出的高层声学表示， \mathbf{P} 为位置编码， \mathbf{Q} 为位置相关总结器输出的隐表示，Enc, Summarize, Dec 分别表示编码器、位置相关总结器和解码器， $P(y_j|\mathbf{X})$ 为解码器输出的每一个位置上词表上词的概率分布。可以看出，位置编码是一个确定性的参数而不是随机变量，是模型的一部分。

5.3.3 训练

模型采用极大似然准则训练。具体上，最小化如下负对数似然函数：

$$NLL(\theta) = -\frac{1}{L} \sum_{j=1}^L \log P_{\theta}(y_j|\mathbf{X}). \tag{5.6}$$

L 是提前设定好的最大长度，如果一个词序列长度小于 L ，那么使用 <e> 符号在末尾补齐。这些 <e> 实际上被用来自适应地预测句子长度。 θ 是整个模型的可训练参数。

可以看出，整个训练过程是端到端的，不依赖其它识别模型生成的帧级别标签。

5.3.4 识别

LASO 模型的识别过程非常简单。直接在每一个位置选择模型输出的概率最大的词即可：

$$\hat{y}_j = \arg \max_{y_j} P_{\theta}(y_j | \mathbf{X}), j = 1, \dots, L. \quad (5.7)$$

模型会在句子末尾预测出 <e> 符号，直接删除即可。

可以看出，这个识别过程不依赖束搜索，不需要多遍网络前馈。而整个模型又是由前馈神经网络构成的，所以计算消耗的时间大大减少，识别速度快。

5.4 跨模态语言知识迁移

由前面的介绍可知，LASO 模型的解码器扮演了语言模型的角色，其采用自注意力机制捕捉语言语义。可以看出，这和预训练语言模型 BERT[7] 模型的出发点相似：都是表示全局上下文的知识。这提示我们可以使用 BERT 模型来提升 LASO 的性能。BERT 是在大规模文本上训练的强大的语言模型，在诸多自然语言处理任务上获得了很好的性能，展现出了强大的语言表示能力。受到前面所介绍的 LST 方法的启发，本节将 BERT 中的知识迁移到 LASO 模型。

本节的基本想法是利用 BERT 生成每个位置词的向量表示，然后令 LASO 生成的表示去逼近 BERT 的表示。具体上，最小化 LASO 的隐层表示和 BERT 的隐层表示之间的均方误差 [110]。如图 5.4 所示。为了和 BERT 训练匹配，在句子开头加一个起始符号 <s>。

在训练时，将训练标注文本输入到 BERT 获得隐层表示，再最小化 LASO 的隐层表示和 BERT 的隐层表示之间的差异。注意到 BERT 的训练文本的标准形式中，句子开头为 [CLS]，结束为 [SEP]，这两个特殊符号代表了 BERT 模型的句子开始和结束⁴，和 LASO 的 <s> 和 <e> 对应，所以要把 LASO 训练文本中的 <s> 和 <e> 替换为 [CLS] 和 [SEP]。另外需要注意的是，BERT 是不需要补齐长度的，有效的句子实际上是 <s> 到第一个 <e> 的词，输入 LASO 模型尾部的

⁴BERT 做预训练时，有一个“预测下一个句子”的任务，所以实际模型一次输入两个句子。但是在提取特征的时候，可以输入一个句子。

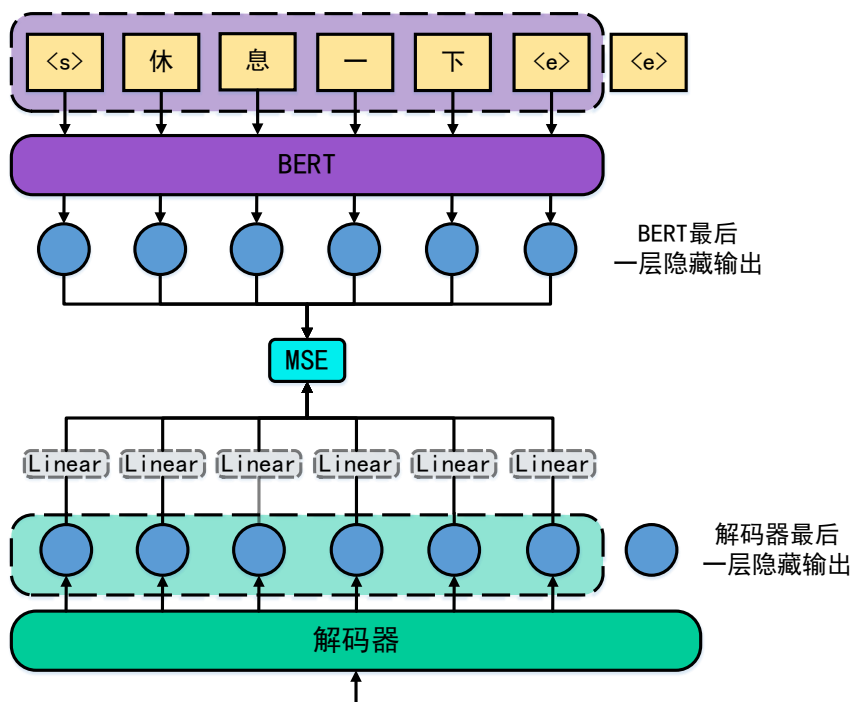


图 5.4 跨模态语言知识迁移

Figure 5.4 cross-modal knowledge transferring

<e> 不需要输入到 BERT 中，训练的时候只优化有效部分对应的损失。损失函数为如下形式

$$\text{MSE}(\theta) = \frac{1}{L_v} \sum_j^{L_v} \sum_d^D (\mathbf{H}_{j,d} - \mathbf{S}_{j,d})^2, \quad (5.8)$$

其中， L_v 表示句子有效长度， $\mathbf{H}_{j,d}$ 表示 LASO 隐层第 j 个表示第 d 个元素， $\mathbf{S}_{j,d}$ 表示 BERT 隐层第 j 个表示第 d 个元素， D 为维度， θ 为 LASO 模型的参数。如果 LASO 的隐层维度和 BERT 不一致，则可以加一个线性变换使维度匹配，如图 5.4 所示。

使用隐层表示而不是输出概率的第一个优势为它避免了采用 BERT 的 [MASK] 符号来计算每一个位置概率造成不能并行计算的问题，效率较高。第二个优势在于，这可以令 LASO 的词表构建更加灵活：只需要让隐藏表示的语义对应，而不需词表完全对应。我们可以只使用 BERT 词表的一部分。

最后，将负对数似然损失和均方误差损失组合起来作为最后的损失：

$$L(\theta) = \text{NLL}(\theta) + \lambda \text{MSE}(\theta), \quad (5.9)$$

其中， λ 是一个系数，用来平衡两者的量级，典型值是 0.005。

可以看出，此跨模态语言知识迁移是前边的 LST 方法的扩展。与 LST 方法一样，它只在训练时使用，不增加识别时的计算代价。

5.5 相关工作

本节介绍并比较相关的工作。

非自回归编码器-解码器模型。非自回归的编码器-解码器模型首先被用在神经机器翻译。Gu 等 [156] 首先提出了一种非自回归编码器解码器模型，其使用一种“繁殖”机制来解决概率分布的多峰问题。Lee 等 [157] 提出一种迭代提炼方法来提升非自回归神经机器翻译的效果。后来辅助损失函数 [158] 和增强的解码器输入 [159] 被用来提升翻译效果。Ma 等 [160] 提出一种流模型 (flow-based models) 来进行机器翻译。这些模型展示了良好的性能和很快的计算速度。然而，这些工作都是机器翻译的模型而不是语音识别的模型。语音识别的特性在于，语音特征序列和文本语言语义同构性，其内在结构都是单调的。这启发本文提出非自回归的编码器-解码器模型。Chen 等 [161] 提出一种迭代补充的方式进行非自回归语音识别。然而，这种方式和本文模型的内在机理不同：本文模型采用位置编码来进行聚合，他们的模型采用文本来进行聚合。所以他们的模型需要多次计算解码器。本文工作重新将语音识别定义为一个逐位置分类问题，提出了位置相关编码器来获得词级别隐藏表示。位置相关编码器是不同长度的声学特征序列和文本序列的桥梁。

跨模态语义对齐。本章提出的跨模态语言知识迁移是一种跨模态的语义空间对齐，即对齐 LASO 和 BERT 的语义。跨模态语义对齐的概念最早用在跨模态检索任务 [162–167]。特别地，2020 年以来基于深度学习的方法学习出图片和文本共享的语义空间 [164–167]，其基本思想是将不同模态的特征映射到同一语义空间，继而可以计算相似度。近期的无语音识别系统的语音检索方法也采用了这个思想 [168]，学习语音和文本查询词的共享语义空间，在检索时让文本查询词和语音进行匹配。本文提出的跨模态语言知识迁移的出发点也是将 LASO 模型和 BERT 模型的语义空间对齐。但是，和检索的工作不同，本文不同时训练两个模型，而是只训练 LASO 模型，BERT 则是起到一个辅助引导的作用，不参与参数更新。另外，我们的方法关注的是将 BERT 模型的知识迁移到 LASO，而非共享语义空间计算相似度。由于本章所提的 LASO 模型是只输入语音的，是一

表 5.1 数据集长度信息统计

Table 5.1 statistics of the lengths in the datasets

| | | 时长 (秒) | | | 句子字数 | | |
|-----------|-----|--------|------|-----|------|------|------|
| | | 最小值 | 最大值 | 平均值 | 最小值 | 最大值 | 平均值 |
| AISHELL-1 | 训练集 | 1.2 | 14.5 | 4.5 | 1.0 | 44.0 | 14.4 |
| | 开发集 | 1.6 | 12.5 | 4.5 | 3.0 | 35.0 | 14.3 |
| | 测试集 | 1.9 | 14.7 | 5.0 | 3.0 | 21.0 | 14.0 |
| AISHELL-2 | 训练集 | 0.5 | 19.3 | 3.6 | 1.0 | 53.0 | 10.9 |
| | 开发集 | 1.1 | 9.4 | 2.9 | 1.0 | 25.0 | 9.9 |
| | 测试集 | 1.1 | 9.4 | 2.9 | 1.0 | 25.0 | 9.9 |

AISHELL-2 中的开发集和测试集为三种设备分别录制的，但内容完全一样，所以表中只列出一种。

个单模态模型，BERT 则是只输入文本的单模态模型，所以本章提出的方法是跨模态知识迁移，即从文本模态的 BERT 迁移到语音模态的 LASO。

5.6 实验

5.6.1 实验数据

语音数据。在语音数据方面，本章采用和前两章相同的开源语音数据集 AISHELL-1 和 AISHELL-2。AISHELL-1 是一个总共 178 小时的中文普通话数据集，AISHELL-2 则为一个规模更大的 1000 小时的中文普通话数据集。关于句子数量、说话人数量、文本规模等信息请见 3.4.1.1 节中的介绍。

由于 LASO 中的位置相关总结器将边长的语音转换为一个固定长度的隐表示序列，并且包含了一个预先设定的参数 L ，所以语音数据集中标注文本的长度可能会成为一个影响模型的重要因素。表 5.1 统计了数据集的长度信息。

纯文本数据。本章的预训练语言模型采用 Google 公司提供的工业级大规模预训练语言模型 BERT[7]。其采用了全部的维基百科中文数据⁵，约两千五百万个句子⁶。

⁵https://en.wikipedia.org/wiki/Wikipedia:Database_download

⁶Google 的中文预训练语言模型没有官方地公布数据细节，这里列出的是从开源数据调研到的信息 <https://github.com/google-research/bert/issues/155>。

表 5.2 模型结构配置符号

Table 5.2 symbols of the architecture

| 符号 | 描述 |
|------------|-----------------|
| D_m | 多头注意力机制的维度 |
| D_{in} | 逐位置的多层感知机的中间层维度 |
| Activation | 激活函数 |
| #Enc. | 编码器层数 |
| #PDS | 位置相关总结器层数 |
| #Dec. | 解码器层数 |

5.6.2 实验设置

基本设置。本章首先在小规模数据 AISHELL-1 (150 小时) 上调整 and 比较模型, 然后将实验扩展到大规模数据 AISHELL-2 上 (1000 小时)。与前两章实验相同, 我们采用 80 维梅尔滤波器组特征作为输入, 特征的帧长为 25 毫秒, 帧移是 10 毫秒。对于 AISHELL-1, 词表由 4231 个训练集中的汉字和 3 个特殊符号 $\langle s \rangle$, $\langle e \rangle$, $\langle \text{unk} \rangle$ 组成。 $\langle s \rangle$ 表示句子开始, $\langle e \rangle$ 表示句子结束, $\langle \text{unk} \rangle$ 表示未见的字。AISHELL-2 的词表大小为 5252。

基线设置。本章采用两个基线系统。我们采用自回归的基于图结构的编码器-解码器模型 (transformer) 作为第一个基线系统 [71], 非自回归的基于自注意力机制的 CTC (SAN-CTC) 作为第二个基线系统 [169]。这两种系统与 LASO 系统都是由基本的 transformer 模块构成的。由于我们想比较非自回归设置下的 CTC 模型, 所以解码过程采用了贪心解码, 没有采用结合 N 元语法的束搜索, 即在解码时我们直接选取概率最大的字, 然后移除空白符号。

对于基于图结构的编码器-解码器模型, 编码器和解码器都采用了 6 层 transformer 模块, 模型维度 512, 注意力机制头数为 8, 逐位置感知机的维度为 2048, 激活函数为 GLU。模型也是用两层 CNN 做降采样。这个模型记为 TRANS。对于基于自注意力机制的 CTC, 因为其只包含编码器部分, 所以我们将编码器层数设置为 12, 使两个基线系统的参数量基本可比。我们将基于自注意力机制的 CTC 记为 SAN-CTC。

LASO 设置。我们比较不同的 LASO 模型的不同结构设置。表示模型结构配置的符号见表 5.2。位置相关总结器模块的位置编码长度设置为 60 (式 5.5 中的 L)，比训练集的最大长度稍大。

训练设置。我们使用 Adam 优化器训练模型。使用热身学习率曲线 [70]:

$$\alpha = D^{-0.5} \cdot \min(\text{step}^{-0.5}, \text{step} \cdot \text{warmup}^{-1.5}). \quad (5.10)$$

热身步数设为 12000。“丢弃” (dropout) 的概率设为 0.1。每一个批包含 100 秒语音。我们累积 12 次前馈的梯度来模拟大的批来稳定训练。模型训练多轮直到收敛。LASO 的典型训练轮数为 130，TRANS 和 SAN-CTC 为 80。

我们使用谱增强作为数据增强方式，频率掩蔽的宽度为 27，时间掩蔽的宽度为 40。频率掩蔽和时间掩蔽都进行 2 次。但是我们没有使用时间弯折。我们采用了标签平滑来对抗过度置信问题，系数为 0.1。我们平均最后 10 轮的检查点的模型参数作为最后的模型。

我们使用 Google 的中文预训练模型 BERT⁷ 来作为跨模态语言知识迁移的语言模型。其由 12 层 transformer 模块构成，总参数量为 1.1 亿，词表大小为 21128。训练时式 5.9 中的 λ 参数为 0.005。

5.6.3 评价准则

对于模型识别准确性，我们采用标准的基于编辑距离的错误率来衡量。具体地，我们针对更适用于中文的字错误率 (character error rates, CERs) 来衡量。

对于速度评价，我们同时计算实时因子 (real-time factor, RTF) 和平均处理时间 (averaged processing time, APT)。实时因子是衡量语音识别系统处理速度的经典指标，它的计算方式为

$$\text{RTF} = \frac{\#\text{Proc.}}{\#\text{Dur.}}, \quad (5.11)$$

其中， $\#\text{Proc.}$ 表示总的处理时间， $\#\text{Dur.}$ 表示总时长。实时因子表示的是处理单位时间语音花费的时间，是一个无量纲的比率。它排除了句子长度对处理时间的影响。为了将句子长度的影响考虑进来，我们还计算平均处理时间：

$$\text{APT} = \frac{\#\text{Proc.}}{\#\text{Utt.}}, \quad (5.12)$$

其中， $\#\text{Utt.}$ 表示测试集句子总数。

⁷https://storage.googleapis.com/bert_models/2018_11_03/chinese_L-12_H-768_A-12.zip

使用平均处理时间的原因是它能衡量处理一个句子的效率。它考虑了用户的等待时间，并且同时适用于在线应用（比如语音交互系统）和离线应用（比如语音文档转写系统）。具体上，对于在线应用，用户的等待时间是“用户停止说话到屏幕上显示识别结果”中间的时间，而对于离线应用，用户等待时间是“用户输入句子和屏幕显示识别结果”中间的时间。实时因子和平均处理时间忽略了语音识别系统不可控的因素，比如网络传输速度等。

对于整句识别的语音识别系统（编码器-解码器模型，基于双向声学模型的混合系统等），实时因子会低估用户等待时间。这是因为整句识别系统需要等待整个语音接收以后才能开始处理。这和流式识别模型不一样。对于流式识别模型，用户等待时间可以由下式估计：

$$\text{RTF} \times |\text{Pkg}|, \quad (5.13)$$

其中， $|\text{Pkg}|$ 表示了最后一个数据包的时长。然而，对于整句识别的系统，句子长度是需要考虑的。所以，本章还计算平均处理时间，直接评估一句话的平均处理时间。我们在常见的深度学习设备上计算这两个指标。我们采用 RTX 2080Ti GPU 来进行实验。需要指出的是，本章的结果中包含了特征提取的时间。

5.6.4 实验结果

5.6.4.1 AISHELL-1 上的结果

模型结构比较。首先，我们比较不同的模型结构配置对模型性能的影响。表 5.3 展示了不同结构配置下的字错误率。我们将模型不同结构配置的影响总结如下：

1. D_m : 更大的多头注意力机制的维度能给模型提供更强的表示能力，使其获得更高性能。

2. **Activation:** 我们发现 GLU 几乎一致地好于 ReLU 激活函数。GLU 包含更多的参数。但是，比较更深的 ReLU 激活函数的模型和更浅的 GLU 模型，二者的参数量差不多，但是采用 GLU 依然能够取得更好的效果。

3. **#Enc. 和 #Dec.:** 增加编码器和解码器的层数能够提升性能。

表 5.3 AISHELL-1: 不同模型结构配置下的字错误率

Table 5.3 The Character Error Rates on AISHELL-1 with Different Hyper-parameters

| Model | Dm | Din | Activation | #Enc. | #PDS | #Dec. | 模型尺寸 (百万) | 不使用 BERT | | 使用 BERT | |
|-------|-----|------|------------|-------|------|-------|-----------|----------|-----|---------|-----|
| | | | | | | | | 开发集 | 测试集 | 开发集 | 测试集 |
| 1 | 256 | 2048 | ReLU | 4 | 1 | 4 | 15.9M | 7.9 | 8.8 | 7.7 | 8.6 |
| 2 | 256 | 2048 | GLU | 4 | 1 | 4 | 20.6M | 7.1 | 8.1 | 7.0 | 7.8 |
| 3 | 256 | 2048 | ReLU | 6 | 1 | 6 | 21.2M | 7.2 | 8.2 | 7.1 | 8.0 |
| 4 | 256 | 2048 | GLU | 6 | 1 | 6 | 28.0M | 6.6 | 7.5 | 6.6 | 7.4 |
| 5 | 256 | 2048 | ReLU | 6 | 2 | 6 | 22.8M | 7.4 | 8.3 | 7.3 | 8.1 |
| 6 | 256 | 2048 | GLU | 6 | 2 | 6 | 30.1M | 6.6 | 7.5 | 6.3 | 7.1 |
| 7 | 256 | 2048 | ReLU | 8 | 2 | 6 | 25.4M | 7.3 | 8.4 | 7.3 | 8.1 |
| 8 | 256 | 2048 | GLU | 8 | 2 | 6 | 33.8M | 6.7 | 7.5 | 6.5 | 7.4 |
| 9 | 512 | 2048 | ReLU | 4 | 1 | 4 | 37.0M | 7.0 | 7.8 | 6.6 | 7.4 |
| 10 | 512 | 2048 | GLU | 4 | 1 | 4 | 46.5M | 6.4 | 7.4 | 5.9 | 6.7 |
| 11 | 512 | 2048 | ReLU | 6 | 1 | 6 | 49.6M | 6.6 | 7.5 | 6.0 | 6.7 |
| 12 | 512 | 2048 | GLU | 6 | 1 | 6 | 63.3M | 6.2 | 7.0 | 5.4 | 6.2 |
| 13 | 512 | 2048 | ReLU | 6 | 2 | 6 | 53.9M | 6.6 | 7.5 | 6.6 | 7.5 |
| 14 | 512 | 2048 | GLU | 6 | 2 | 6 | 68.6M | 6.1 | 6.9 | 5.4 | 6.1 |
| 15 | 512 | 2048 | ReLU | 8 | 2 | 6 | 60.2M | 6.6 | 7.5 | 6.2 | 6.9 |
| 16 | 512 | 2048 | GLU | 8 | 2 | 6 | 80.0M | 5.9 | 6.6 | 5.2 | 5.8 |
| 17 | 768 | 2048 | ReLU | 6 | 1 | 6 | 85.5M | 6.5 | 7.3 | 5.9 | 6.6 |
| 18 | 768 | 2048 | GLU | 6 | 1 | 6 | 105.8M | 5.9 | 6.9 | 5.3 | 6.1 |

4. #PDS: 我们比较了位置相关总结器取一层和两层的情形, 发现二者没有很大的区别。

5. 基于 BERT 的跨模态语言知识迁移: 我们可以看出当 $D_m = 512$ 的时候, 采用基于 BERT 的跨模态位置迁移, 性能提升十分明显。但是当 $D_m = 256$ 的时候性能提升就比较小了。我们分析这是由于当 LASO 和 BERT 的模型维度差异太大的时候, LASO 比较难以学习到 BERT 的表示。

在本章后边部分, 我们将表 5.3 中的模型 2, 模型 12 和模型 16 记为 LASO-small, LASO-middle 和 LASO-big, 用于后边的实验。

与其它方法比较。接下来, 我们比较模型和基线系统的性能。我们还同时和其它工作的结果做一个比较。比较结果展示在表 5.4。可以看出, 本章所提方法 LASO 显示出了有竞争力的性能。特别地, 使用跨模态迁移学习方法将 BERT 中的知识迁移到 LASO 进一步地提升了性能。同时, 实验展示了 LASO 在测试时的处理速度大大快于自回归模型。

我们实现了一个具有竞争力的自回归编码器-解码器模型 TRANS, 其 AISHELL-1 上的字错误率为 6.6%。规模差不多的 LASO-middle 的性能和其接近。更大的模型 LASO-big 的性能超过了 TRANS。LASO-middle 和 LASO-big 超过了基于 CTC 的非自回归模型 SAN-CTC。

表 5.4 还列出了实时因子和平均处理时间。这两个指标都是在一块 RTX 2080Ti GPU 上, 以一次处理一个句子的方式计算的。可以看出, 非自回归模型的识别速度大大快于自回归模型。平均处理时间方面, LASO 是自回归模型的 1/50。同时, 可以看出, 虽然 LASO-big 参数量比自回归模型 TRANS 大了很多, 但是处理速度依然很快。这是因为 LASO 是非自回归模型和前馈神经网络, 并行实现非常高效。

非自回归模型 SAN-CTC 识别速度也很快。但是, 和同规模的 LASO 相比, 识别准确率略差。我们分析这是由于 LASO 的解码器可以有效地捕捉语言语义。另外, CTC 模型插入的空白符号有可能会影响模型捕捉语言语义, 造成性能下降。这证明了本章所提出的扮演自回归语言模型角色的解码器的有效性。

对于 LASO-middle 和 LASO-big, 基于 BERT 的跨模态迁移学习使字错误率相对下降了约 11% 到 12%。这证明了本章所提出的将 BERT 中的语言知识迁移到 LASO 的方法可以帮助 LASO 提升语言语义的捕捉能力。然而, 对于小尺寸

表 5.4 AISHELL-1: 和基线模型比较

Table 5.4 Comparisons with Baselines on AISHELL-1

| 模型 | 参数量 | 字错误率 | | RTF / APT |
|--------------------------------|-------|------------|------------|---------------|
| | | 开发集 | 测试集 | |
| KALDI(nnet3) * † ‡ | - | - | 8.6 | - |
| KALDI(chain) * † ‡ | - | - | 7.4 | - |
| LAS [133] | - | 9.4 | 10.6 | - |
| ESPNet (Transformer) † ‡ [134] | - | 6.0 | 6.7 | - |
| A-FMLM [161] | - | 6.2 | 6.7 | - |
| Fan 等 (Transformer) [135] | - | - | 6.7 | - |
| AGS CTC ‡ [170] | - | 7.0 | 7.9 | - |
| TRANS (基线 1) | 67.5M | 6.1 | 6.6 | 0.19 / 961ms |
| SAN-CTC (基线 2) | 56.4M | 7.2 | 7.8 | 0.0033 / 16ms |
| LASO-small | 20.6M | 7.1 | 8.1 | 0.0027 / 13ms |
| LASO-small + BERT | 20.6M | 7.0 | 7.8 | 0.0027 / 13ms |
| LASO-middle | 63.3M | 6.2 | 7.0 | 0.0035 / 17ms |
| LASO-middle + BERT | 63.3M | 5.4 | 6.2 | 0.0035 / 17ms |
| LASO-big | 80.0M | 5.9 | 6.6 | 0.0040 / 20ms |
| LASO-big + BERT | 80.0M | 5.2 | 5.8 | 0.0040 / 20ms |

* KALDI 官方代码库包含结果⁸。

† 使用速度扰动做数据增强。

‡ 测试时采用额外的语言模型。

的 LASO-small, 该方法的提升并不明显。我们分析这是由于小模型 LASO-small 和 BERT 的维度差别太大 (256 维和 768 维), 影响了知识迁移。

⁸<https://github.com/kaldi-asr/kaldi/blob/master/egs/aishell/s5/RESULTS>

表 5.5 AISHELL-2: 与基线系统的比较
Table 5.5 Comparisons with Baselines on AISHELL-2

| 模型 | 参数量 | 开发集 | | | | 测试集 | | | |
|----------------------------|-------|------------|------------|------------|------------|------------|------------|------------|------------|
| | | iPhone | Android | 高保真麦克风 | 平均 | iPhone | Android | 高保真麦克风 | 平均 |
| KALDI (chain) [122] † ‡ | - | 9.1 | 10.4 | 11.8 | 10.4 | 8.8 | 9.6 | 10.9 | 9.8 |
| LAS [133] | - | - | - | - | - | 9.2 | 9.7 | 10.3 | 9.7 |
| ESPNet (transformer) * † ‡ | - | - | - | - | - | 7.5 | 8.9 | 8.6 | 8.3 |
| TRANS (基线 1) | 67.5M | 6.4 | 7.2 | 7.7 | 7.1 | 7.1 | 8.0 | 8.2 | 7.8 |
| SAN-CTC (基线 2) | 56.4M | 8.3 | 8.9 | 8.8 | 8.6 | 8.0 | 9.0 | 8.9 | 8.7 |
| LASO-small | 20.6M | 8.2 | 9.5 | 9.7 | 9.1 | 8.5 | 9.5 | 9.5 | 9.2 |
| LASO-small + BERT | 20.6M | 8.9 | 10.0 | 10.3 | 9.7 | 8.8 | 9.8 | 10.5 | 9.7 |
| LASO-middle | 63.3M | 6.6 | 7.5 | 7.6 | 7.2 | 6.8 | 7.4 | 7.3 | 7.2 |
| LASO-middle + BERT | 63.3M | 6.5 | 7.2 | 7.4 | 7.0 | 6.6 | 7.2 | 7.1 | 7.0 |
| LASO-large | 80.0M | 6.4 | 7.3 | 7.3 | 7.0 | 6.7 | 7.4 | 7.4 | 7.1 |
| LASO-large + BERT | 80.0M | 6.2 | 7.2 | 7.3 | 6.9 | 6.5 | 7.2 | 7.1 | 6.9 |

* ESPnet 官方代码库的结果 <https://github.com/espnet/espnet/blob/master/egs/aishell2/asr1/RESULTS.md>。

† 使用了速度扰动数据增强。

‡ 使用了额外语言模型。

5.6.4.2 AISHELL-2 上的结果

接下来，本章将实验扩展到更大规模的数据集 AISHELL-2。AISHELL-2 包含 1000 小时语音数据，内容覆盖也更为丰富。并且，其训练集是用 iPhone 手机录制，测试集则分为 iPhone 手机，Android 手机和高保真麦克风，方便测试模型在不同信道条件下的鲁棒性。由于 AISHELL-2 的规模比较大，训练模型比较耗时，所以我们直接采用了前边 AISHELL-1 中选择的模型结构。

表 5.5 展示了 AISHELL-2 上的结果。可以看出，LASO 模型依然显示出了较好的性能。特别是我们发现，在使用了更大规模的数据的时候，和同规模的自回归模型基线系统相比，LASO 取得了更好的性能。使用基于 BERT 的跨模态迁移学习的时候，识别正确率有了进一步提升，但是不像前边小规模数据那样显著了。特别地，对于 LASO-small，性能还有所下降。这个现象的原因可能有两个：1) 模型规模上 LASO-small 和 BERT 差异太大，影响了迁移学习；2) AISHELL-2 比 AISHELL-1 数据更多，复杂性更大，影响了老师-学生学习的效果 [171, 172]。

5.6.5 分析与讨论

前边的实验中，我们展示了 LASO 模型的性能。本小节进一步分析与讨论 LASO 模型的性质。首先我们展示 LASO 模型的注意力模式，然后分析训练集句子长度带来的影响。

5.6.5.1 注意力机制的可视化与分析

为了更好地理解 LASO 模型的行为，我们对 LASO-big 模型生成的注意力分数可视化。我们从测试集中选择了一个句子，供模型计算。为了节省空间，这里我们只展示各注意力分数 8 个头中的前 4 个。其它的可视化结果见附录。图 5.5、图 5.6、图 5.7 分别展示了编码器、解码器和位置相关总结器的注意力分数。我们可以将观察总结如下。

1. 不同的头有不同的注意力模式。这意味着同一个表示向量在不同的头注意到了不同的表示向量。
2. 对于编码器，有一些注意力模式是呈现出左上角到右下角的对角线(图 5.5(b)和(c))。这是符合直觉的。对一个语音帧，相邻的语音帧与其最为匹配。然而，还有一些头并没有显示明显的注意力模式。
3. 对于解码器，我们可以看到有的字注意到了前一个字(图 5.6(c))，有的注

意到了后一个字 (图 5.6(b))。大部分用于补全的结束符号 $\langle e \rangle$ 注意到句子起始 $\langle s \rangle$ 。

4. 对位置相关总结器, 字的位置注意到了一个小范围的编码器输出。总体而言注意力模式是左上到右下的 (图 5.7)。对于前几个位置的 $\langle e \rangle$ 符号, 其注意力模式表现为一条竖线。但是, 不同的头竖线的位置不一样 (图 5.7(b) 和 (d))。大部分其它 $\langle e \rangle$ 符号均匀地注意到前面的一些编码器输出位置。我们分析这是因为它是一个填充符号, 扮演全局的角色。

从以上观察, 我们可以总结出如下结论: 1) 注意力机制不同的头可以从不同的方面对特征融合; 2) 根据位置编码, 注意力机制可以学习出有意义的对齐模式; 3) 填充符号 $\langle s \rangle$ 和 $\langle e \rangle$ 吸收了编码器中无意义的输出 (比如静音等); 4) 解码器的注意力机制里存在一些典型模式。这些表明位置相关总结器可以注意到对应不同位置的声学特征, 解码器可以利用自注意力机制捕捉到词的隐层表示之间的关系。

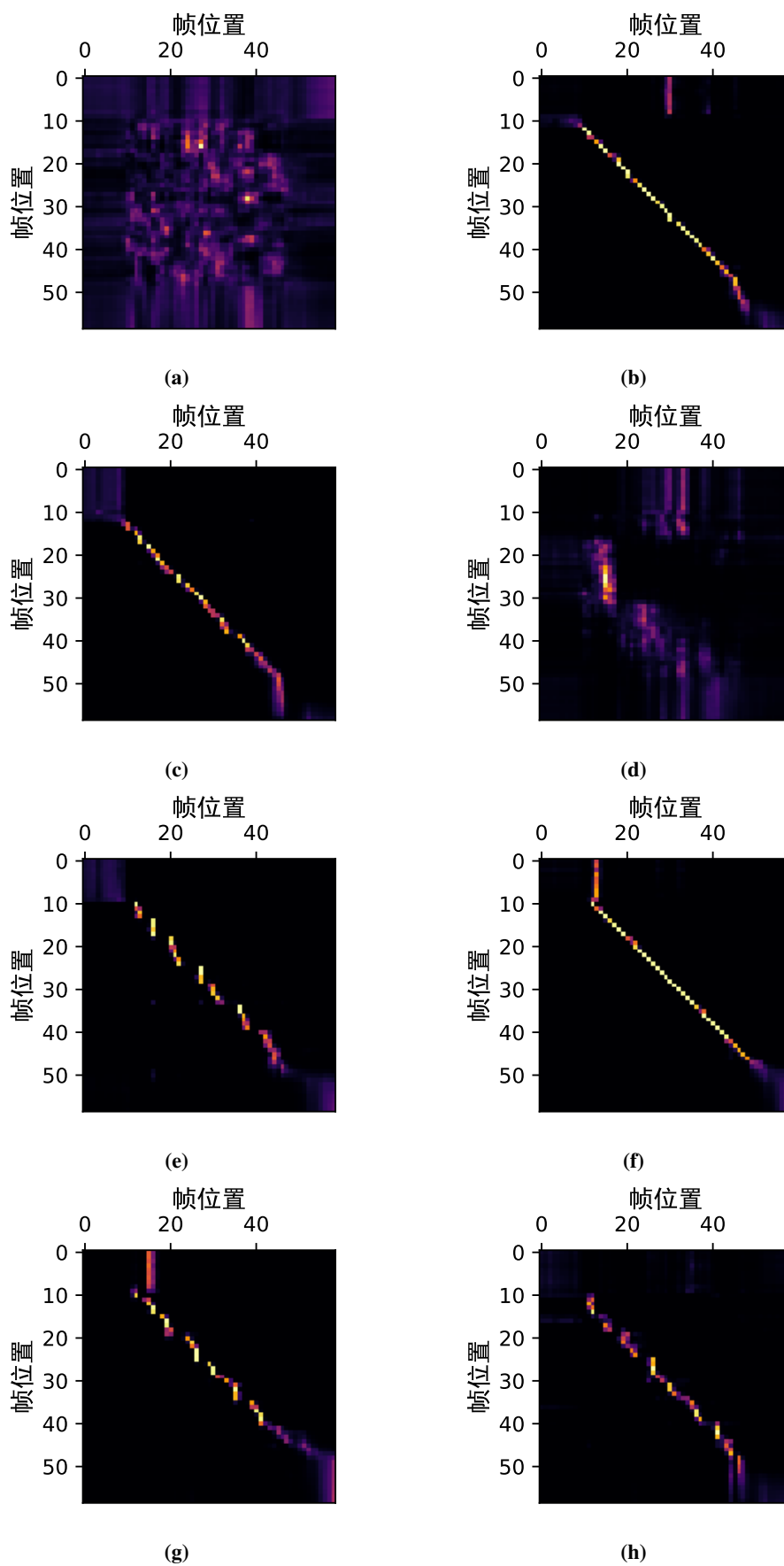


图 5.5 LASO 编码器最后一层自注意力分数的可视化结果

Figure 5.5 the visualization of the last self-attention of LASO encoder.

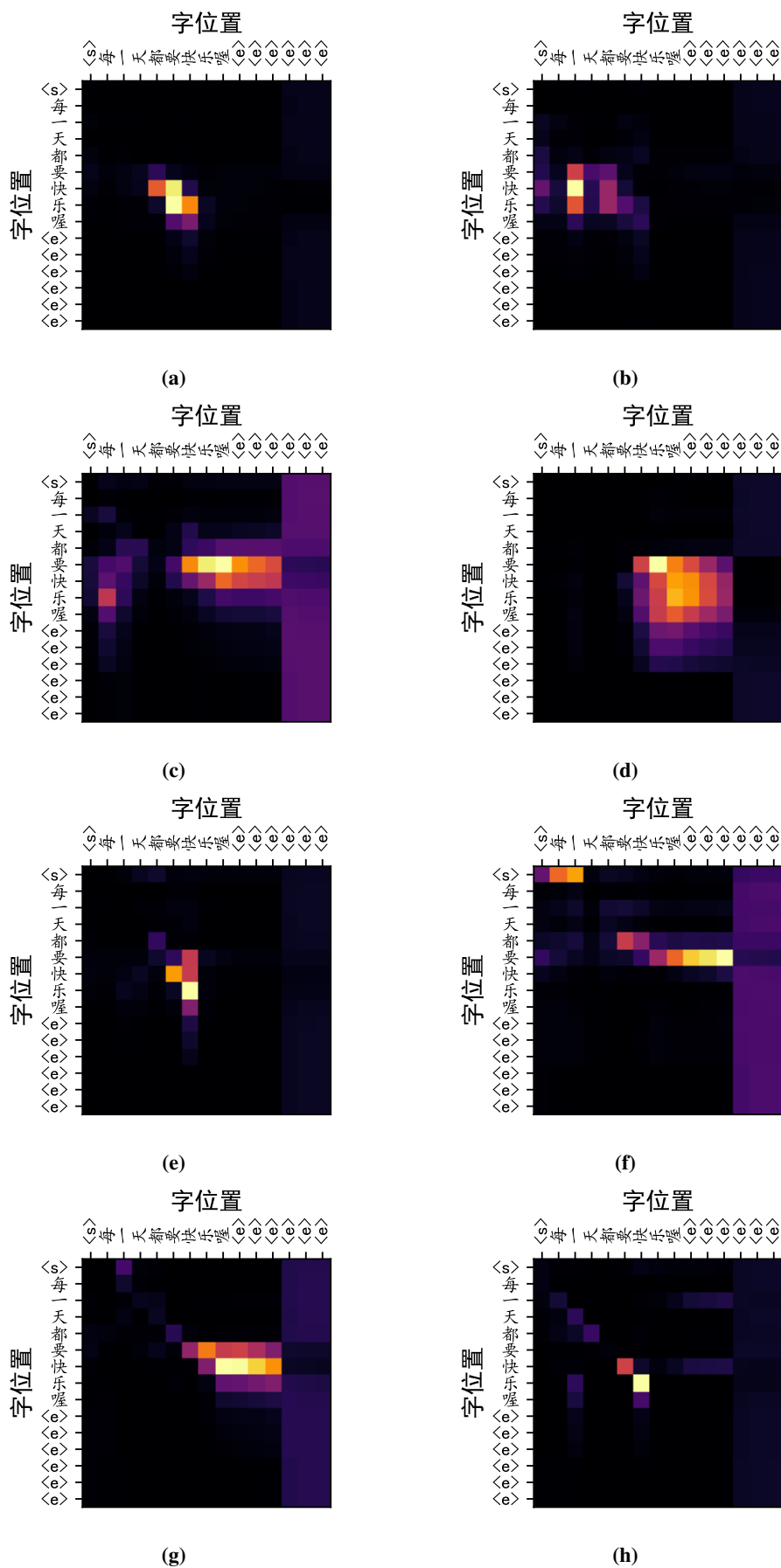


图 5.6 LASO 解码器最后一层自注意力分数的可视化结果

Figure 5.6 the visualization of the last self-attention of LASO decoder.

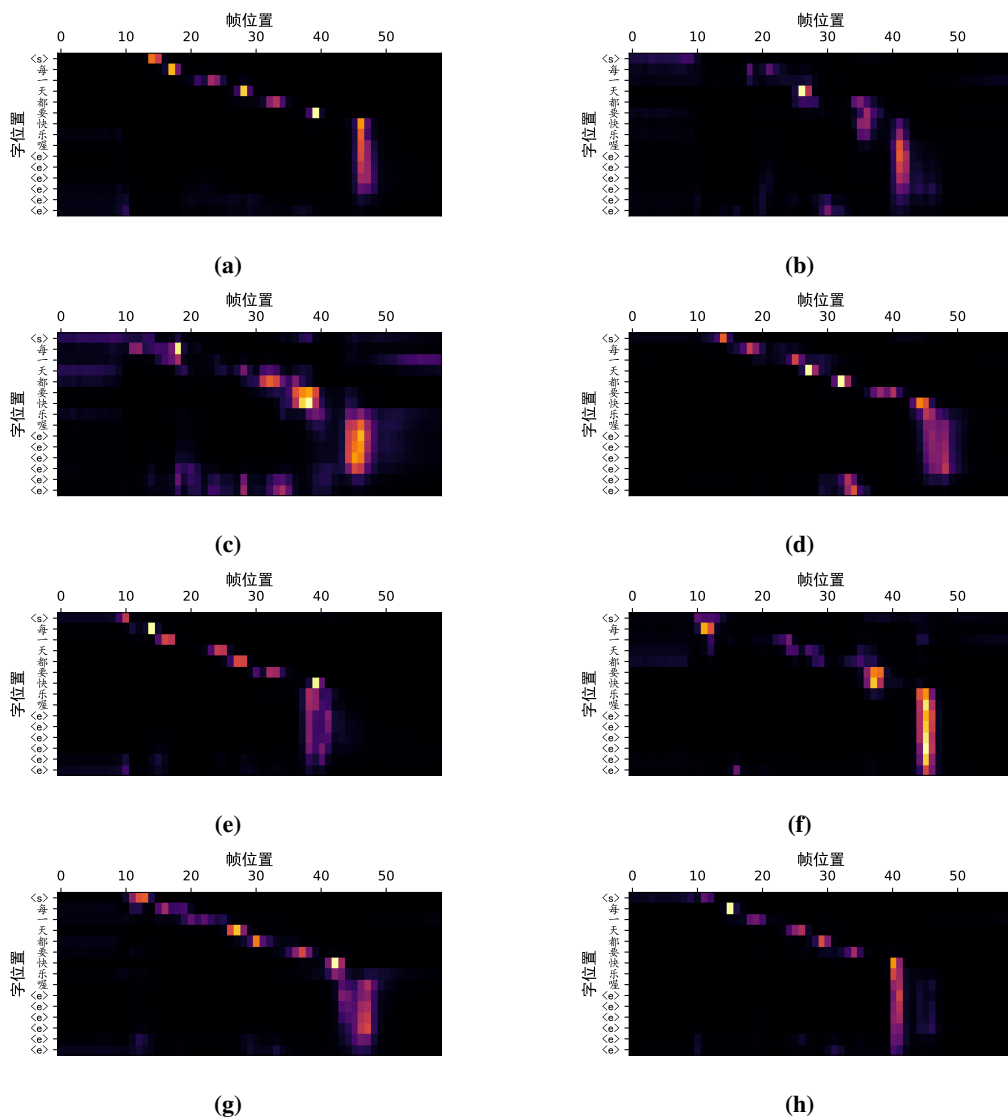


图 5.7 LASO 位置相关总结器最后一层注意力分数的可视化结果

Figure 5.7 the visualization of the last attention of LASO PDS.

5.6.5.2 句子长度的分析

位置相关总结器模块包含位置参数，这意味着句子的长度有可能影响性能。为了检查这一点，我们绘制训练集句子长度的分布直方图，和测试集句子长度与字错误率关系的散点图。我们使用了 TRANS, SAN-CTC 和 LASO-middle 模型，在 AISHELL-1 和 AISHELL-2 (iPhone) 上的结果上进行统计。图 5.8 展示了结果。可以看出，三个模型没有表现出显著区别。

图 5.8 的灰色部分为训练集句子长度的分布。可以看出，句子长度分布近似于一个高斯分布的形状，这是符合中心极限定律的预期的。字错误率的趋势和句子长度分布有关：更多的句子在中间部分字错误率为 0，也就是说训练样本多的长度范围内，模型表现更好。三个模型都可以识别未见长度 (或者少见的长度) 的样本，即图 5.8(a) 的 few-shot 部分。但是，其错误率相对来说比“见过的”长度高。这个现象是符合预期的。这是因为三个模型 (TRANS, SAN-CTC, LASO) 都是整句识别的语音识别系统，句子长度是一个训练模型的隐含因素。这和基于局部窗口的模型 (如时延神经网络、卷积神经网络、延迟控制的双向长短时记忆网络的声学模型) 不一样。LASO 相对来说比 TRANS 和 SAN-CTC 对句子长度更敏感。这是因为位置相关总结器需要被不同长度的句子训练来提升泛化能力。

在工程实践中，有两种方法可以来帮助整句语音识别系统解决上述未见长度的问题。1) 选择长度更丰富的训练语料来训练模型；2) 使用音频端点检测系统将长句子截为短句子，再进行识别。

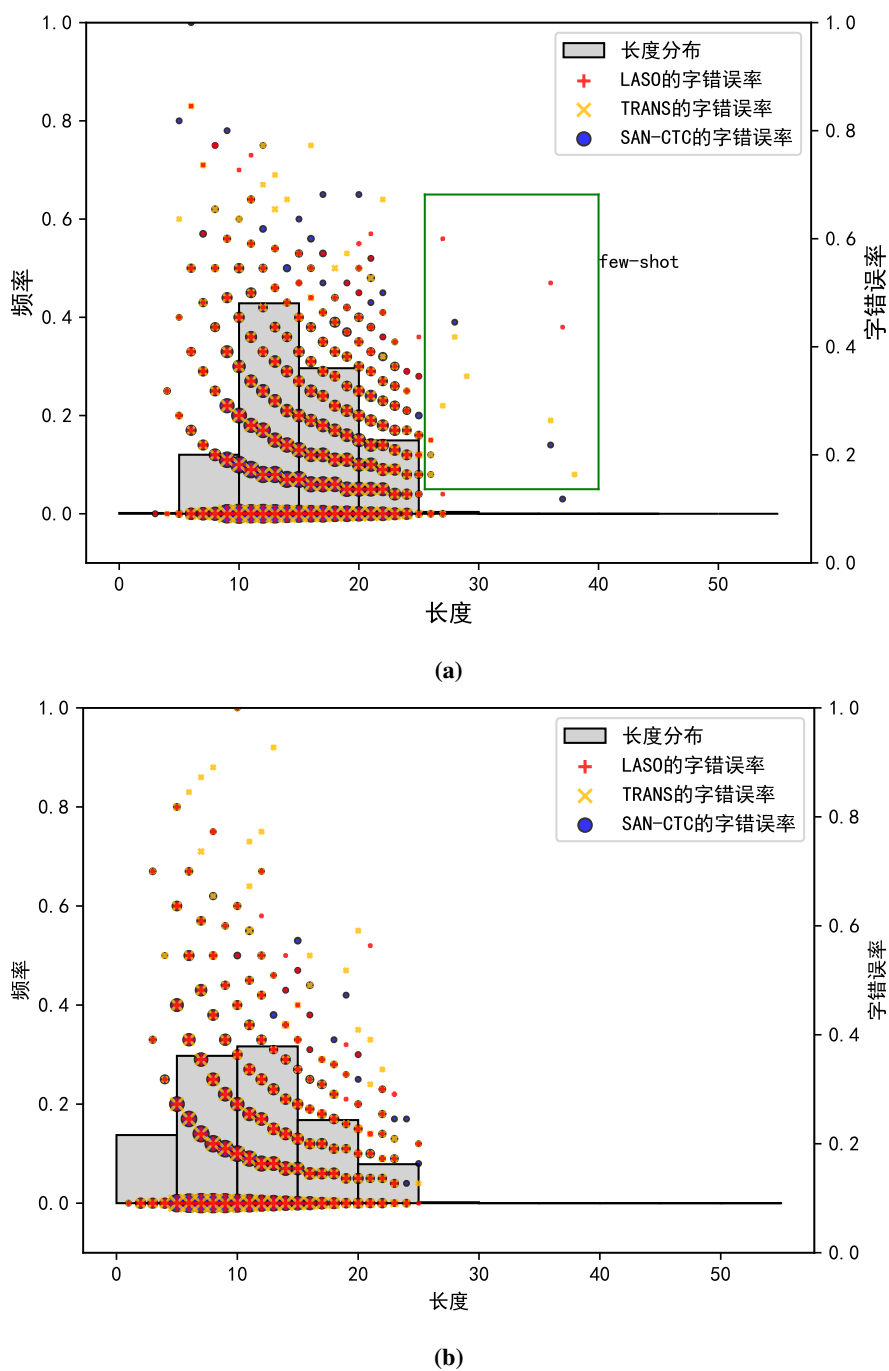


图 5.8 训练集长度分布直方图和测试集句子长度与字错误率关系的散点图。散点的面积大小和字错误率数值大小成正关系。(a) AISHELL-1 上的结果, (b) AISHELL-2 上的结果。
 Figure 5.8 Histograms of lengths on training set and scatter plots of CER (lengths vs. CERs). (a) AISHELL-1, (b) AISHELL-2.

5.7 小结

基于对“语音和其对应文本中词与词之间为同构的”这一现象的观察，本章提出两个问题。

1. 能否利用语音中的语言语义直接进行语音识别，而不进行显式地语言建模。
2. 能否跨模态地将文本模型中的语言知识迁移到语音模型，提升语音识别性能。

本章提出两个方法，依次序回答上述问题。首先提出了基于图结构前馈神经网络的非自回归语音识别 LASO，其利用位置相关总结器自适应地将声学特征序列聚合为对应词的隐表示，然后通过解码器进一步地捕捉隐表示之间的关系，最后逐位置地分类。由于 LASO 模型是由前馈神经网络构成，且预测一个词的时候不依赖其它词，非常利于并行实现，所以识别速度极快，达到自回归基线系统的近 50 倍。这表明，直接利用语音而不进行显式语言建模来实现语音识别是可行的。

其次，本章提出了跨模态全局语言知识迁移，将商用预训练语言模型 BERT 中的语言知识迁移到 LASO 中。实验表明，跨模态语言知识迁移可以有效提升 LASO 模型的识别性能。这表明，跨模态地将文本模型中的语言知识迁移到语音模型，确实可以提升语音识别的性能。

第6章 总结与展望

6.1 本文工作总结

大规模文本语料预训练出的文本表示模型已被证明能够提取到语言知识,可以提升分类和匹配等语言信息处理任务的性能。然而,如何在文本生成任务中将大规模文本语料中的知识利用起来还缺乏深入的研究。本文针对这一问题,从“上文语言知识迁移”、“全局上下文语言知识迁移”、“跨模态全局上下文语言知识迁移”三个递进的层面进行研究,取得了以下成果。

1. 提出一种文本知识利用方法。本文提出了一种基于老师-学生学习的文本知识利用方法 LST,来提升端到端语音识别的性能:首先利用语言模型将大规模纯文本中的语言知识表示起来,然后利用老师-学生学习将此语言知识迁移到端到端语音识别系统中。与其它方法相比,该方法不增加预测阶段的计算代价,比较高效;同时,该方法可以利用其它开放获取的已经训练好的语言模型而不需要自行训练,方便灵活。本文还分析比较了该方法与另一种典型的文本知识利用方法浅融合,发现平滑模型估计的分数空间是这两种方法提升识别性能的重要性质。

2. 提出一种全局上下文语言模型。针对端到端编码器-解码器没有利用文本中上下文知识的问题,本文提出一种全局上下文语言模型“因果完形填空器”,然后利用 LST 方法将此全局上下文语言知识迁移到端到端编码器-解码器模型中,使得编码器-解码器模型也可以利用全局上下文语言知识。相比其它的利用双向语言知识的方法,该方法不增加识别阶段复杂性,还可以灵活地利用无标注纯文本。

3. 证实了利用语音中包含的语言知识可以有效进行语音识别。基于观察到的语音与文本的语言知识同构现象,本文提出一种端到端非自回归语音识别模型 LASO。该方法没有显式地自回归语言建模,所以可以并行地实现同时预测所有的词。实验表明,所提模型在两种规模的公开中文语音数据集上都可以表现出与自回归模型可比的性能,但处理速度是自回归模型的近 50 倍。这些结果表明,不进行显式地自回归语言建模,而是利用语音中的语言知识,也可以进行高效的语音识别。

4. 提出一种跨模态全局语言知识迁移方法,有效提升了单模态语音识别模型性能。根据文本与语音的语言知识同构性,本文提出将大规模预训练语言模型中的语言知识跨模态地迁移到非自回归语音识别模型 LASO 中。实验证明,所提方法可以提升纯语音模态建模的端到端语音识别模型的效果。同时结果表明,利用不同模态的语言知识同构性进行知识迁移,可以有效地提升不同模态模型的性能。

6.2 未来工作展望

本文认为,在将来的工作中,有两方面问题值得进一步研究。

一、针对端到端语音识别系统更为灵活的语言知识集成方法

端到端语音识别系统以其性能好、系统总体体积小等特点正慢慢地成为语音识别的主流实用方法。然而,相对于传统的声学、语言分开建模的混合系统,端到端语音识别系统难以通过更换语言模型的方法快速适配到某些特定领域。具体来说,传统混合模型系统可以通过搜集一些特定领域(如法律、医学等)文本训练 N 元语法语言模型,然后通过更换语言模型快速适配到这些领域,而端到端语音识别系统由于是一体化建模的,难以做到这一点。而这对于某些场景的快速部署是很重要的。所以,如何灵活地将语言知识集成到端到端识别系统,使其能够快速适配到特定领域,是一个值得研究的问题。

二、老师-学习的理论分析与新方法研究

老师-学生学习是一种简单而灵活的训练方法,被广泛地应用在迁移学习、模型压缩等问题中。然而,老师-学生学习的理论目前还几乎处于空白阶段。比如,什么样的老师模型是好的老师模型?老师-学生学习的泛化边界是什么样的?学习隐藏层表示和学习后验概率哪种方法更好?这些问题还都没有理论分析。目前针对老师-学生学习的理论有了一些较为初步的结果,Phuong 等 [173]针对线性分类器作为学生模型的情形证明了泛化界。其它还有一些实验方面的解释,如 Cheng 等 [174]说明老师-学生学习可以让神经网络更多地学习到一些视觉概念,并帮助稳定学生模型的训练。然而,由于老师-学生学习方法的灵活性(比如可以学习不同网络结构的隐藏层,老师模型和学生模型可以是不同任务的模型等),该方法的理论分析还远不完善,值得进一步探索。另外,什么样的方法是更为有效的老师-学生学习方法也值得进一步研究。

附录 A 梅尔滤波器组特征的提取

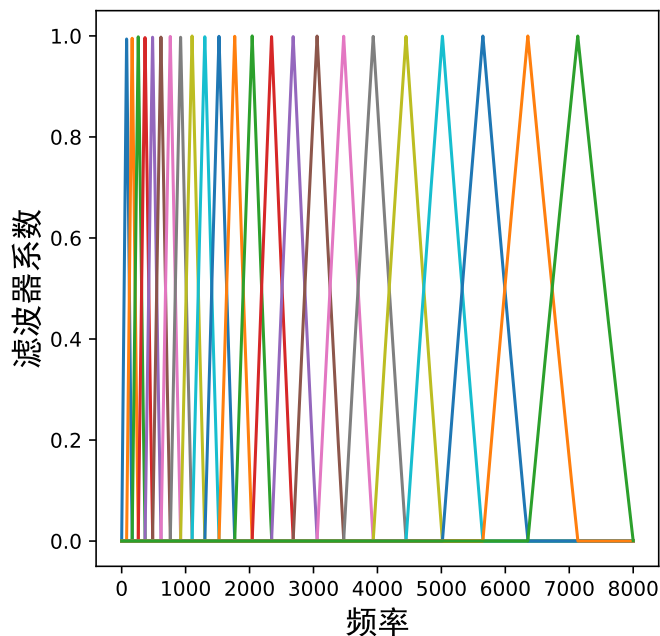


图 A.1 梅尔尺度三角滤波器组示意图

Figure A.1 an illustration of mel-scale triangular filter banks.

梅尔滤波器组特征 (mel-scale filter bank features, FBANK) 被广泛地应用在基于深度学习声学建模的语音识别系统中，是目前最为流行的用于语音识别的声学特征。本文的所有实验都采用了梅尔滤波器组特征。

梅尔滤波器组特征的提取可以分为 3 步¹。

1. 划窗、分帧、预加重等；
2. 对每一帧计算快速傅里叶变换，并计算功率谱；
3. 计算每一个梅尔尺度下滤波器组中，每一个三角滤波器中的能量，并取对数²。

如果进一步地提取梅尔频率倒谱 (mel-frequency cepstral coefficient, MFCC) 系数特征，只需要再进行离散余弦变换即可。

在划窗、分帧时，一般对语音信号每 10 毫秒截取一个帧长为 25 毫秒的帧，

¹具体的实现可以参考 KALDI 工具箱。

<https://github.com/kaldi-asr/kaldi/blob/master/src/feat/feature-fbank.cc#L72>。

²当前流行的做法是取对数能量，这样的话就只和 MFCC 特征差一个离散余弦变换了。

比如一秒的语音就可以分出约 100 帧。为了防止信号截断造成的频谱泄露，需要再利用汉明窗等对边界进行平滑。然后再进行快速傅里叶变换，并计算出功率谱(每一个频率点模长的平方)。

梅尔尺度是基于人耳听感构建的一种非线性尺度。它表示人耳对等量音高变化的感觉，也就是说在一定的范围内，人对高频的音高变化相对不敏感。梅尔尺度和线性尺度的一种经验变换公式为

$$\text{Mel}(f) = 1127 \ln\left(1 + \frac{f}{700}\right), \quad (\text{A.1})$$

其中， f 是在线性尺度下的频率，单位为赫兹。

具体在计算梅尔尺度三角滤波器能量的时候，是通过设定三角滤波器带宽直接在线性尺度频域实现的，不需要先变换再计算。也就是说，是在梅尔尺度下构建等带宽的滤波器组，再将其反变换到线性尺度。得到的滤波器组的形状大约为如图 A.1。

一个滤波器能量就是功率谱以三角滤波器的系数为权重的加权和。

附录 B 端到端语音识别模型的一些训练技巧

本附录介绍一些常用的提升端到端语音识别模型的训练技巧。

B.1 谱增强

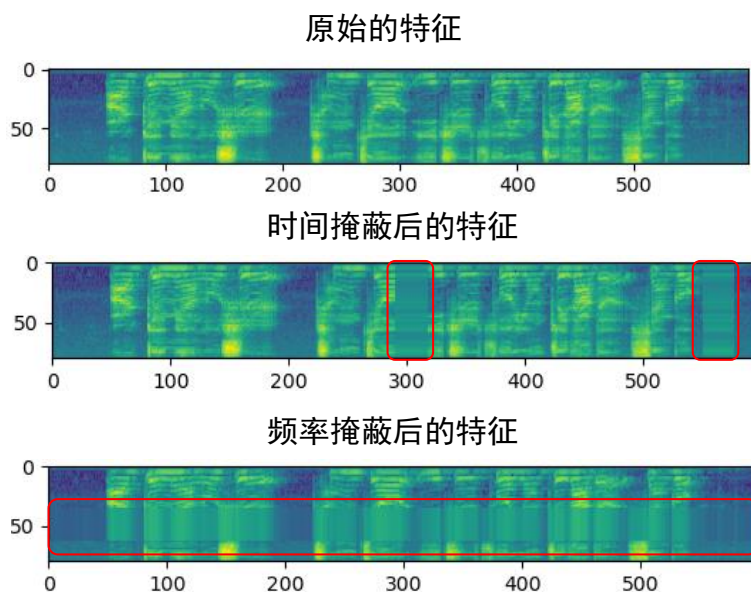


图 B.1 谱增强技术示意图

Figure B.1 an illustration of SpecAug

谱增强 [127] 技术是一种简单而有效地数据增强技术。在训练模型时，谱增强随机地将语音特征中的某些部分进行替换。这种方法已经被证明是提升语音识别模型泛化能力的有效方法。其中最简单而有效的两个操作是时间掩蔽和频率掩蔽。本文所有的实验都应用了这一技术。

时间掩蔽的操作步骤为：

1. 全局特征均值；
2. 随机选取掩蔽宽度，宽度服从均匀分布，范围大小为超参数；
3. 随机选取时间起始点，使用均值替换从起始点开始的掩蔽范围内的特征。

频率掩蔽的操作步骤为：

1. 全局特征均值；

2. 随机选取掩蔽宽度，宽度服从均匀分布，范围大小为超参数；
3. 随机选取掩蔽频率起始点，使用均值替换从起始点开始的掩蔽范围内的特征。

谱增强方法是近年来被证明最为有效的数据增强方法，在大规模工业数据集和小规模数据集上都有良好表现。

B.2 模型平均

模型平均是一种简单而有效的提升语音识别模型性能的方法。它将不同检查点的参数进行平均。具体上它的做法是

$$\bar{\theta} = \frac{1}{N} \sum_{i=1}^N \theta_i. \quad (\text{B.1})$$

其中， θ_i 是第 i 个检查点的参数， θ 是模型最终参数。

模型平均可以看做是一种集成方法，在大规模工业数据集和小规模数据集上都有良好表现。

B.3 最小词错误率训练

最小词错误率训练是一种以词错误率 (word error rate, WER) 作为风险的最小化贝叶斯风险训练 [175]。它的损失函数如下

$$\begin{aligned} \text{MWER}(\theta) &= E[w(y|y^*)] \\ &\approx \frac{1}{N} w(\tilde{y}|y^*) P(\tilde{y}|\mathbf{X}), \end{aligned} \quad (\text{B.2})$$

其中， $w(y|y^*)$ 表示一个词序列 y 和标准答案 y^* 的编辑距离或词错误率 (根据标准答案长度归一化的编辑距离)， $E[\cdot]$ 表示期望， \tilde{y} 表示采样的词序列， \mathbf{X} 表示声学特征序列。在优化时， w 是作为常数出现的，不参与优化，优化的参数包含在概率里。

由于不可能枚举出所有词序列，所以最小词错误率的计算一般采用采样方法近似计算，和混合模型的序列级区分性训练相同。最小词错误率一般可以分为两个步骤。

1. 采样：利用束搜索等方法采样出 N 个候选；
2. 计算：计算候选的词错误率和模型估计的概率，以进行优化。

一种最小词错误率训练的变种是

$$\text{MWER}(\theta) \approx \frac{1}{N}(w(\tilde{y}|y^*) - \bar{w})P(\tilde{y}|\mathbf{X}), \quad (\text{B.3})$$

其中， \bar{w} 是平均词错误率。可以看出，当 $w(\tilde{y}|y^*) > \bar{w}$ 时，证明采样到了一个比较差的样本 \tilde{y} ，所以最小化 $P(\tilde{y}|\mathbf{X})$ ；当 $w(\tilde{y}|y^*) < \bar{w}$ 时，证明采样到了一个比较好的样本 \tilde{y} ，所以最大化 $P(\tilde{y}|\mathbf{X})$ 。

在我们的实验中，对基于双向循环神经网络的模型，最小词错误率训练效果比较显著，但对基于图的前馈神经网络模型上效果相对较小。其中的原因还需要进一步探索。

参考文献

- [1] Bengio Y, Courville A, Vincent P. Representation learning: A review and new perspectives [J/OL]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 35(8): 1798-1828. DOI: [10.1109/TPAMI.2013.50](https://doi.org/10.1109/TPAMI.2013.50).
- [2] Jing L, Tian Y. Self-supervised visual feature learning with deep neural networks: A survey [J/OL]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020: 1-1. DOI: [10.1109/TPAMI.2020.2992393](https://doi.org/10.1109/TPAMI.2020.2992393).
- [3] He K, Fan H, Wu Y, et al. Momentum contrast for unsupervised visual representation learning [C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 9729-9738.
- [4] Chen T, Kornblith S, Norouzi M, et al. A simple framework for contrastive learning of visual representations [C]//III H D, Singh A. Proceedings of Machine Learning Research: volume 119 Proceedings of the 37th International Conference on Machine Learning. Virtual: PMLR, 2020: 1597-1607.
- [5] Grill J B, Strub F, Alché F, et al. Bootstrap your own latent - a new approach to self-supervised learning [C]//Advances in Neural Information Processing Systems. 2020.
- [6] Peters M, Neumann M, Iyyer M, et al. Deep contextualized word representations [J]. 2018.
- [7] Devlin J, Chang M W, Lee K, et al. BERT: Pre-training of deep bidirectional transformers for language understanding [J]. 2019.
- [8] Brown T, Mann B, Ryder N, et al. Language models are few-shot learners [C]//Advances in Neural Information Processing Systems. 2020.
- [9] Baevski A, Zhou Y, Mohamed A, et al. wav2vec 2.0: A framework for self-supervised learning of speech representations [C]//Advances in Neural Information Processing Systems. 2020.
- [10] Chung Y A, Glass J. Improved speech representations with multi-target autoregressive predictive coding [C/OL]//Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Online: Association for Computational Linguistics, 2020: 2353-2358. <https://www.aclweb.org/anthology/2020.acl-main.213>. DOI: [10.18653/v1/2020.acl-main.213](https://doi.org/10.18653/v1/2020.acl-main.213).
- [11] Zhu J, Xia Y, Wu L, et al. Incorporating BERT into neural machine translation [C/OL]//8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net, 2020. <https://openreview.net/forum?id=Hyl7ygStwB>.
- [12] Davis K H, Biddulph R, Balashek S. Automatic recognition of spoken digits [J]. The Journal of the Acoustical Society of America, 1952, 24(6): 637-642.

- [13] Huang X, Baker J, Reddy R. A historical perspective of speech recognition [J]. *Communications of the ACM*, 2014, 57(1): 94-103.
- [14] Baker J. The dragon system—an overview [J]. *IEEE Transactions on Acoustics, speech, and signal Processing*, 1975, 23(1): 24-29.
- [15] Jelinek F, Bahl L, Mercer R. Design of a linguistic statistical decoder for the recognition of continuous speech [J]. *IEEE Transactions on Information Theory*, 1975, 21(3): 250-256.
- [16] Lee K F. Automatic speech recognition: the development of the sphinx system: volume 62 [M]. Springer Science & Business Media, 1988.
- [17] Juang B H, Levinson S, Sondhi M. Maximum likelihood estimation for multivariate mixture observations of markov chains (corresp.) [J]. *IEEE Transactions on Information Theory*, 1986, 32(2): 307-309.
- [18] Hinton G, Deng L, Yu D, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups [J/OL]. *IEEE Signal Processing Magazine*, 2012, 29(6): 82-97. DOI: [10.1109/MSP.2012.2205597](https://doi.org/10.1109/MSP.2012.2205597).
- [19] Xiong W, Droppo J, Huang X, et al. Toward human parity in conversational speech recognition [J/OL]. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2017, 25(12): 2410-2423. DOI: [10.1109/TASLP.2017.2756440](https://doi.org/10.1109/TASLP.2017.2756440).
- [20] He Y, Sainath T N, Prabhavalkar R, et al. Streaming end-to-end speech recognition for mobile devices [C]//ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019: 6381-6385.
- [21] Olson H F, Belar H. Phonetic typewriter [J]. *The Journal of the Acoustical Society of America*, 1956, 28(6): 1072-1081.
- [22] Denes P. The design and operation of the mechanical speech recognizer at university college london [J/OL]. *Journal of the British Institution of Radio Engineers*, 1959, 19(4): 219-229. DOI: [10.1049/jbire.1959.0027](https://doi.org/10.1049/jbire.1959.0027).
- [23] Forgie J W, Forgie C D. Results obtained from a vowel recognition computer program [J]. *The Journal of the Acoustical Society of America*, 1959, 31(11): 1480-1489.
- [24] Nagata K, Kato Y, Chiba S. Spoken digit recognizer for japanese language [C]//Audio Engineering Society Convention 16. Audio Engineering Society, 1964.
- [25] Martin T B, Nelson A, Zadell H. Speech recognition by feature-abstraction techniques. [R]. RAYTHEON CO WALTHAM MASS, 1964.
- [26] Vintsyuk T K. Speech discrimination by dynamic programming [J]. *Cybernetics*, 1968, 4(1): 52-57.
- [27] Sakoe H, Chiba S. A dynamic programming approach to continuous speech recognition [C]//Proc. 7th Int. Congr. Acoustics. 1971.

- [28] Itakura F. Minimum prediction residual principle applied to speech recognition [J/OL]. IEEE Transactions on Acoustics, Speech, and Signal Processing, 1975, 23(1): 67-72. DOI: [10.1109/TASSP.1975.1162641](https://doi.org/10.1109/TASSP.1975.1162641).
- [29] Sakoe H, Chiba S. Dynamic programming algorithm optimization for spoken word recognition [J]. IEEE transactions on acoustics, speech, and signal processing, 1978, 26(1): 43-49.
- [30] Bahl L, Jelinek F. Decoding for channels with insertions, deletions, and substitutions with applications to speech recognition [J/OL]. IEEE Transactions on Information Theory, 1975, 21(4): 404-411. DOI: [10.1109/TIT.1975.1055419](https://doi.org/10.1109/TIT.1975.1055419).
- [31] Jelinek F. Continuous speech recognition by statistical methods [J/OL]. Proceedings of the IEEE, 1976, 64(4): 532-556. DOI: [10.1109/PROC.1976.10159](https://doi.org/10.1109/PROC.1976.10159).
- [32] Rabiner L R. A tutorial on hidden markov models and selected applications in speech recognition [J/OL]. Proceedings of the IEEE, 1989, 77(2): 257-286. DOI: [10.1109/5.18626](https://doi.org/10.1109/5.18626).
- [33] Mei-Yuh Hwang, Xuedong Huang. Shared-distribution hidden markov models for speech recognition [J/OL]. IEEE Transactions on Speech and Audio Processing, 1993, 1(4): 414-420. DOI: [10.1109/89.242487](https://doi.org/10.1109/89.242487).
- [34] Young S J, Odell J J, Woodland P C. Tree-based state tying for high accuracy modelling [C]// HUMAN LANGUAGE TECHNOLOGY: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994. 1994.
- [35] Seide F, Li G, Yu D. Conversational speech transcription using context-dependent deep neural networks [C]//Twelfth annual conference of the international speech communication association. 2011.
- [36] Dahl G E, Yu D, Deng L, et al. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition [J/OL]. IEEE Transactions on Audio, Speech, and Language Processing, 2012, 20(1): 30-42. DOI: [10.1109/TASL.2011.2134090](https://doi.org/10.1109/TASL.2011.2134090).
- [37] Bourlard H, Wellekens C J. Links between markov models and multilayer perceptrons [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1990, 12(12): 1167-1178.
- [38] Graves A, Jaitly N, Mohamed A r. Hybrid speech recognition with deep bidirectional lstm [C]//2013 IEEE workshop on automatic speech recognition and understanding. IEEE, 2013: 273-278.
- [39] Abdel-Hamid O, Mohamed A, Jiang H, et al. Applying convolutional neural networks concepts to hybrid nn-hmm model for speech recognition [C/OL]//2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2012: 4277-4280. DOI: [10.1109/ICASSP.2012.6288864](https://doi.org/10.1109/ICASSP.2012.6288864).
- [40] Peddinti V, Povey D, Khudanpur S. A time delay neural network architecture for efficient

- modeling of long temporal contexts [C]//Sixteenth Annual Conference of the International Speech Communication Association. 2015.
- [41] Qian Y, Bi M, Tan T, et al. Very deep convolutional neural networks for noise robust speech recognition [J]. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2016, 24(12): 2263-2276.
- [42] Zhang S, Lei M, Yan Z, et al. Deep-fsmn for large vocabulary continuous speech recognition [C]//2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018: 5869-5873.
- [43] Bahl L, Brown P, De Souza P, et al. Maximum mutual information estimation of hidden markov model parameters for speech recognition [C]//ICASSP'86. IEEE International Conference on Acoustics, Speech, and Signal Processing: volume 11. IEEE, 1986: 49-52.
- [44] Chou W, Lee C H, Juang B H. Minimum error rate training based on n-best string models [C/OL]//1993 IEEE International Conference on Acoustics, Speech, and Signal Processing: volume 2. 1993: 652-655 vol.2. DOI: [10.1109/ICASSP.1993.319394](https://doi.org/10.1109/ICASSP.1993.319394).
- [45] Biing-Hwang Juang, Wu Hou, Chin-Hui Lee. Minimum classification error rate methods for speech recognition [J/OL]. *IEEE Transactions on Speech and Audio Processing*, 1997, 5(3): 257-265. DOI: [10.1109/89.568732](https://doi.org/10.1109/89.568732).
- [46] Valtchev V, Odell J J, Woodland P C, et al. Lattice-based discriminative training for large vocabulary speech recognition [C/OL]//1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings: volume 2. 1996: 605-608 vol. 2. DOI: [10.1109/ICASSP.1996.543193](https://doi.org/10.1109/ICASSP.1996.543193).
- [47] Kaiser J, Horvat B, Kacic Z. A novel loss function for the overall risk criterion based discriminative training of hmm models [C]//Sixth International Conference on Spoken Language Processing. 2000.
- [48] Povey D, Woodland P C. Minimum phone error and i-smoothing for improved discriminative training [C]//2002 IEEE International Conference on Acoustics, Speech, and Signal Processing: volume 1. IEEE, 2002: I-105.
- [49] Kingsbury B. Lattice-based optimization of sequence classification criteria for neural-network acoustic modeling [C/OL]//2009 IEEE International Conference on Acoustics, Speech and Signal Processing. 2009: 3761-3764. DOI: [10.1109/ICASSP.2009.4960445](https://doi.org/10.1109/ICASSP.2009.4960445).
- [50] Vesely K, Ghoshal A, Burget L, et al. Sequence-discriminative training of deep neural networks. [C]//Interspeech: volume 2013. 2013: 2345-2349.
- [51] Povey D, Peddinti V, Galvez D, et al. Purely sequence-trained neural networks for asr based on lattice-free mmi. [C]//Interspeech. 2016: 2751-2755.
- [52] Graves A, Fernández S, Gomez F, et al. Connectionist temporal classification: labelling

- unsegmented sequence data with recurrent neural networks [C]//Proceedings of the 23rd international conference on Machine learning. 2006: 369-376.
- [53] Graves A, Mohamed A r, Hinton G. Speech recognition with deep recurrent neural networks [C]//2013 IEEE international conference on acoustics, speech and signal processing. IEEE, 2013: 6645-6649.
- [54] Miao Y, Gowayyed M, Metze F. Eesen: End-to-end speech recognition using deep rnn models and wfst-based decoding [C]//2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU). IEEE, 2015: 167-174.
- [55] Hadian H, Sameti H, Povey D, et al. Flat-start single-stage discriminatively trained hmm-based models for asr [J/OL]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2018, 26(11): 1949-1961. DOI: [10.1109/TASLP.2018.2848701](https://doi.org/10.1109/TASLP.2018.2848701).
- [56] Chorowski J K, Bahdanau D, Serdyuk D, et al. Attention-based models for speech recognition [J]. Advances in neural information processing systems, 2015, 28: 577-585.
- [57] Chan W, Jaitly N, Le Q, et al. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition [C]//2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2016: 4960-4964.
- [58] Li M, Liu M, Masanori H. End-to-end speech recognition with adaptive computation steps [C]//ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019: 6246-6250.
- [59] Dong L, Xu B. Cif: Continuous integrate-and-fire for end-to-end speech recognition [C/OL]// ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2020: 6079-6083. DOI: [10.1109/ICASSP40776.2020.9054250](https://doi.org/10.1109/ICASSP40776.2020.9054250).
- [60] Graves A. Sequence transduction with recurrent neural networks [J]. arXiv preprint arXiv:1211.3711, 2012.
- [61] Tian Z, Yi J, Tao J, et al. Self-attention transducers for end-to-end speech recognition [J]. Proc. Interspeech 2019, 2019: 4395-4399.
- [62] Tian Z, Yi J, Bai Y, et al. Synchronous transformers for end-to-end speech recognition [C]// ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020: 7884-7888.
- [63] Yang Z, Dai Z, Yang Y, et al. Xlnet: Generalized autoregressive pretraining for language understanding [J]. 2019: 5753-5763.
- [64] Calvert G A. Crossmodal processing in the human brain: insights from functional neuroimaging studies [J]. Cerebral cortex, 2001, 11(12): 1110-1123.
- [65] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate [C/OL]//Bengio Y, LeCun Y. 3rd International Conference on Learning Representen-

- tations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings. 2015. <http://arxiv.org/abs/1409.0473>.
- [66] Graves A. Generating sequences with recurrent neural networks [J]. arXiv preprint arXiv:1308.0850, 2013.
- [67] Xu K, Ba J, Kiros R, et al. Show, attend and tell: Neural image caption generation with visual attention [C]//International conference on machine learning. PMLR, 2015: 2048-2057.
- [68] Raffel C, Luong M T, Liu P J, et al. Online and linear-time attention by enforcing monotonic alignments [C]//International Conference on Machine Learning. PMLR, 2017: 2837-2846.
- [69] Reddy D R, et al. Speech understanding systems: A summary of results of the five-year research effort [J]. Department of Computer Science. Camegie-Mell University, Pittsburgh, PA, 1977, 17: 138.
- [70] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need [C/OL]//Guyon I, Luxburg U V, Bengio S, et al. Advances in Neural Information Processing Systems: volume 30. Curran Associates, Inc., 2017: 5998-6008. <https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf>.
- [71] Dong L, Xu S, Xu B. Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition [C]//2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018: 5884-5888.
- [72] Guo Q, Qiu X, Liu P, et al. Star-transformer [C]//Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). 2019: 1315-1325.
- [73] Jarrett K, Kavukcuoglu K, Ranzato M, et al. What is the best multi-stage architecture for object recognition? [C]//2009 IEEE 12th international conference on computer vision. IEEE, 2009: 2146-2153.
- [74] Nair V, Hinton G E. Rectified linear units improve restricted boltzmann machines [C/OL]//Fürnkranz J, Joachims T. Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel. Omnipress, 2010: 807-814. <https://icml.cc/Conferences/2010/papers/432.pdf>.
- [75] Glorot X, Bordes A, Bengio Y. Deep sparse rectifier neural networks [C]//Proceedings of the fourteenth international conference on artificial intelligence and statistics. JMLR Workshop and Conference Proceedings, 2011: 315-323.
- [76] Ramachandran P, Zoph B, Le Q V. Searching for activation functions [J]. arXiv preprint arXiv:1710.05941, 2017.
- [77] Dauphin Y N, Fan A, Auli M, et al. Language modeling with gated convolutional networks [C]//International conference on machine learning. PMLR, 2017: 933-941.

- [78] Shaw P, Uszkoreit J, Vaswani A. Self-attention with relative position representations [C]// Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers). 2018: 464-468.
- [79] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition [C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.
- [80] Ba L J, Kiros J R, Hinton G E. Layer normalization [J/OL]. CoRR, 2016, abs/1607.06450. <http://arxiv.org/abs/1607.06450>.
- [81] Nguyen T Q, Salazar J. Transformers without tears: Improving the normalization of self-attention [J]. arXiv preprint arXiv:1910.05895, 2019.
- [82] Xiong R, Yang Y, He D, et al. On layer normalization in the transformer architecture [C]// International Conference on Machine Learning. PMLR, 2020: 10524-10533.
- [83] Dai Z, Yang Z, Yang Y, et al. Transformer-XL: Attentive language models beyond a fixed-length context [C/OL]//Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Florence, Italy: Association for Computational Linguistics, 2019: 2978-2988. <https://www.aclweb.org/anthology/P19-1285>. DOI: 10.18653/v1/P19-1285.
- [84] Kitaev N, Kaiser Ł, Levskaya A. Reformer: The efficient transformer [J]. arXiv preprint arXiv:2001.04451, 2020.
- [85] Gulcehre C, Firat O, Xu K, et al. On using monolingual corpora in neural machine translation [J]. arXiv preprint arXiv:1503.03535, 2015.
- [86] Chorowski J, Jaitly N. Towards better decoding and language model integration in sequence to sequence models [C/OL]//Proc. Interspeech 2017. 2017: 523-527. <http://dx.doi.org/10.21437/Interspeech.2017-343>.
- [87] Sriram A, Jun H, Satheesh S, et al. Cold fusion: Training seq2seq models together with language models [C/OL]//Proc. Interspeech 2018. 2018: 387-391. <http://dx.doi.org/10.21437/Interspeech.2018-1392>.
- [88] Toshniwal S, Kannan A, Chiu C C, et al. A comparison of techniques for language model integration in encoder-decoder speech recognition [C]//2018 IEEE spoken language technology workshop (SLT). IEEE, 2018: 369-375.
- [89] Kannan A, Wu Y, Nguyen P, et al. An analysis of incorporating an external language model into a sequence-to-sequence model [C]//2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018: 1-5828.
- [90] Stahlberg F, Cross J, Stoyanov V. Simple fusion: Return of the language model [C]// Proceedings of the Third Conference on Machine Translation: Research Papers. 2018: 204-211.

- [91] Shan C, Weng C, Wang G, et al. Component fusion: Learning replaceable language model component for end-to-end speech recognition system [C]//ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019: 5361-5635.
- [92] Meng Z, Parthasarathy S, Sun E, et al. Internal language model estimation for domain-adaptive end-to-end speech recognition [J]. arXiv preprint arXiv:2011.01991, 2020.
- [93] McDermott E, Sak H, Variani E. A density ratio approach to language model fusion in end-to-end automatic speech recognition [C]//2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). IEEE, 2019: 434-441.
- [94] Tjandra A, Sakti S, Nakamura S. Listening while speaking: Speech chain by deep learning [C]//2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). IEEE, 2017: 301-308.
- [95] Hayashi T, Watanabe S, Zhang Y, et al. Back-translation-style data augmentation for end-to-end asr [C]//2018 IEEE Spoken Language Technology Workshop (SLT). IEEE, 2018: 426-433.
- [96] Renduchintala A, Ding S, Wiesner M, et al. Multi-modal data augmentation for end-to-end asr [C/OL]//Proc. Interspeech 2018. 2018: 2394-2398. <http://dx.doi.org/10.21437/Interspeech.2018-2456>.
- [97] Karita S, Watanabe S, Iwata T, et al. Semi-supervised end-to-end speech recognition using text-to-speech and autoencoders [C/OL]//ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2019: 6166-6170. DOI: [10.1109/ICASSP.2019.8682890](https://doi.org/10.1109/ICASSP.2019.8682890).
- [98] Li B, Sainath T N, Pang R, et al. Semi-supervised training for end-to-end models via weak distillation [C/OL]//ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2019: 2837-2841. DOI: [10.1109/ICASSP.2019.8682172](https://doi.org/10.1109/ICASSP.2019.8682172).
- [99] Sainath T N, Pang R, Weiss R J, et al. An attention-based joint acoustic and text on-device end-to-end model [C]//ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020: 7039-7043.
- [100] Wang P, Sainath T N, Weiss R J. Multitask training with text data for end-to-end speech recognition [J]. arXiv preprint arXiv:2010.14318, 2020.
- [101] Wang G, Rosenberg A, Chen Z, et al. Improving speech recognition using consistent predictions on synthesized speech [C/OL]//ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2020: 7029-7033. DOI: [10.1109/ICASSP40776.2020.9053831](https://doi.org/10.1109/ICASSP40776.2020.9053831).
- [102] Pham V T, Xu H, Khassanov Y, et al. Independent language modeling architecture for end-to-

- end asr [C/OL]//ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2020: 7059-7063. DOI: [10.1109/ICASSP40776.2020.9054116](https://doi.org/10.1109/ICASSP40776.2020.9054116).
- [103] Pan S J, Yang Q. A survey on transfer learning [J/OL]. IEEE Transactions on Knowledge and Data Engineering, 2010, 22(10): 1345-1359. DOI: [10.1109/TKDE.2009.191](https://doi.org/10.1109/TKDE.2009.191).
- [104] Bai Y, Yi J, Tao J, et al. Learn Spelling from Teachers: Transferring Knowledge from Language Models to Sequence-to-Sequence Speech Recognition [C/OL]//Proc. Interspeech 2019. 2019: 3795-3799. <http://dx.doi.org/10.21437/Interspeech.2019-1554>.
- [105] Zhang Z, Shi Y, Yuan C, et al. Object relational graph with teacher-recommended learning for video captioning [C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 13278-13288.
- [106] Bucila C, Caruana R, Niculescu-Mizil A. Model compression [C/OL]//Eliassi-Rad T, Ungar L H, Craven M, et al. Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, August 20-23, 2006. ACM, 2006: 535-541. <https://doi.org/10.1145/1150402.1150464>.
- [107] Ba L J, Caruana R. Do deep nets really need to be deep? [J]. arXiv preprint arXiv:1312.6184, 2013.
- [108] Li J, Zhao R, Huang J, et al. Learning small-size DNN with output-distribution-based criteria [C/OL]//Li H, Meng H M, Ma B, et al. INTERSPEECH 2014, 15th Annual Conference of the International Speech Communication Association, Singapore, September 14-18, 2014. ISCA, 2014: 1910-1914. http://www.isca-speech.org/archive/interspeech_2014/i14_1910.html.
- [109] Hinton G E, Vinyals O, Dean J. Distilling the knowledge in a neural network [J/OL]. CoRR, 2015, abs/1503.02531. <http://arxiv.org/abs/1503.02531>.
- [110] Romero A, Ballas N, Kahou S E, et al. Fitnets: Hints for thin deep nets [C/OL]//Bengio Y, LeCun Y. 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings. 2015. <http://arxiv.org/abs/1412.6550>.
- [111] Yu D, Yao K, Su H, et al. K1-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition [C/OL]//IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, BC, Canada, May 26-31, 2013. IEEE, 2013: 7893-7897. <https://doi.org/10.1109/ICASSP.2013.6639201>.
- [112] Asami T, Masumura R, Yamaguchi Y, et al. Domain adaptation of DNN acoustic models using knowledge distillation [C/OL]//2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017, New Orleans, LA, USA, March 5-9, 2017. IEEE, 2017: 5185-5189. <https://doi.org/10.1109/ICASSP.2017.7953145>.

- [113] Gou J, Yu B, Maybank S J, et al. Knowledge distillation: A survey [J/OL]. CoRR, 2020, abs/2006.05525. <https://arxiv.org/abs/2006.05525>.
- [114] Tang Z, Wang D, Zhang Z. Recurrent neural network training with dark knowledge transfer [C/OL]//2016 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016, Shanghai, China, March 20-25, 2016. IEEE, 2016: 5900-5904. <https://doi.org/10.1109/ICASSP.2016.7472809>.
- [115] Huang Z, Wang N. Like what you like: Knowledge distill via neuron selectivity transfer [J/OL]. CoRR, 2017, abs/1707.01219. <http://arxiv.org/abs/1707.01219>.
- [116] Xu W, Rudnicky A. Can artificial neural networks learn language models? [C/OL]//Sixth International Conference on Spoken Language Processing, ICSLP 2000 / INTERSPEECH 2000, Beijing, China, October 16-20, 2000. ISCA, 2000: 202-205. http://www.isca-speech.org/archive/icslp_2000/i00_1202.html.
- [117] Bengio Y, Ducharme R, Vincent P, et al. A neural probabilistic language model [J/OL]. J. Mach. Learn. Res., 2003, 3: 1137-1155. <http://jmlr.org/papers/v3/bengio03a.html>.
- [118] Mikolov T, Karafiát M, Burget L, et al. Recurrent neural network based language model [C/OL]//Kobayashi T, Hirose K, Nakamura S. INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010. ISCA, 2010: 1045-1048. http://www.isca-speech.org/archive/interspeech_2010/i10_1045.html.
- [119] Hochreiter S, Schmidhuber J. Long short-term memory [J/OL]. Neural Comput., 1997, 9(8): 1735-1780. <https://doi.org/10.1162/neco.1997.9.8.1735>.
- [120] Cho K, van Merriënboer B, Gulcehre C, et al. Learning phrase representations using RNN encoder–decoder for statistical machine translation [C/OL]//Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Doha, Qatar: Association for Computational Linguistics, 2014: 1724-1734. <https://www.aclweb.org/anthology/D14-1179>. DOI: 10.3115/v1/D14-1179.
- [121] Bu H, Du J, Na X, et al. Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline [C]//2017 20th Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment (O-COCOSDA). IEEE, 2017: 1-5.
- [122] Du J, Na X, Liu X, et al. Aishell-2: transforming mandarin asr research into industrial scale [J]. arXiv preprint arXiv:1808.10583, 2018.
- [123] Bai Y, Tao J, Yi J, et al. CLMAD: A chinese language model adaptation dataset [C]//The Eleventh International Symposium on Chinese Spoken Language Processing (ISCSLP 2018). 2018.

- [124] Bai Y, Yi J, Tao J, et al. A public chinese dataset for language model adaptation [J/OL]. *J. Signal Process. Syst.*, 2020, 92(8): 839-851. <https://doi.org/10.1007/s11265-019-01482-5>.
- [125] Li J, Sun M. Scalable term selection for text categorization [C]//*Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. 2007.
- [126] Rousseau A. Xenc: An open-source tool for data selection in natural language processing [J]. *The Prague Bulletin of Mathematical Linguistics*, 2013, 100: 73-82.
- [127] Park D S, Chan W, Zhang Y, et al. SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition [J]. *Proc. Interspeech 2019*, 2019: 2613-2617.
- [128] Kingma D P, Ba J. Adam: A method for stochastic optimization [J]. *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.
- [129] Irie K, Zeyer A, Schlüter R, et al. Language modeling with deep transformers [J]. *Proc. Interspeech 2019*, 2019: 3905-3909.
- [130] Szegedy C, Vanhoucke V, Ioffe S, et al. Rethinking the inception architecture for computer vision [C]//*Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016: 2818-2826.
- [131] Pereyra G, Tucker G, Chorowski J, et al. Regularizing neural networks by penalizing confident output distributions [J]. *International Conference on Learning Representations Workshop*, 2017.
- [132] Chorowski J, Jaitly N. Towards better decoding and language model integration in sequence to sequence models [J]. *Proc. Interspeech 2017*, 2017: 523-527.
- [133] Sun S, Guo P, Xie L, et al. Adversarial regularization for attention based end-to-end robust speech recognition [J]. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2019, 27(11): 1826-1838.
- [134] Karita S, Wang X, Watanabe S, et al. A comparative study on transformer vs RNN in speech applications [J]. *IEEE Automatic Speech Recognition and Understanding Workshop, ASRU 2019, Singapore, December 14-18, 2019*, 2019: 449-456.
- [135] Fan Z, Zhou S, Xu B. Unsupervised pre-training for sequence to sequence speech recognition [J]. *arXiv preprint arXiv:1910.12418*, 2019.
- [136] An K, Xiang H, Ou Z. Cat: Crf-based asr toolkit [J]. *arXiv preprint arXiv:1911.08747*, 2019.
- [137] Mun J, Lee K, Shin J, et al. Learning to specialize with knowledge distillation for visual question answering [C/OL]//Bengio S, Wallach H, Larochelle H, et al. *Advances in Neural Information Processing Systems: volume 31*. Curran Associates, Inc., 2018: 8081-8091. <https://proceedings.neurips.cc/paper/2018/file/0f2818101a7ac4b96ceeba38de4b934c-Paper.pdf>.

- [138] Bengio S, Vinyals O, Jaitly N, et al. Scheduled sampling for sequence prediction with recurrent neural networks [J]. *Advances in Neural Information Processing Systems* 28, 2015: 1171-1179.
- [139] Zheng Y, Tao J, Wen Z, et al. Forward-backward decoding sequence for regularizing end-to-end tts [J]. *IEEE Transactions on Audio, Speech, and Language Processing*, 2019, 27(12): 2067-2079.
- [140] Mimura M, Sakai S, Kawahara T. Forward-backward attention decoder [J]. *Proc. Interspeech* 2018, 2018: 2232-2236.
- [141] Zhou L, Zhang J, Zong C. Synchronous bidirectional neural machine translation [J]. *Transactions of the Association for Computational Linguistics*, 2019, 7: 91-105.
- [142] Liu L, Utiyama M, Finch A, et al. Agreement on target-bidirectional neural machine translation [C]//*Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2016: 411-416.
- [143] Zhang Z, Wu S, Liu S, et al. Regularizing neural machine translation by target-bidirectional agreement [C]//*Proceedings of the AAAI Conference on Artificial Intelligence: volume 33*. 2019: 443-450.
- [144] Taylor W L. “cloze procedure” : A new tool for measuring readability [J]. *Journalism Bulletin*, 1953, 30(4): 415-433.
- [145] Mousa A, Schuller B. Contextual bidirectional long short-term memory recurrent neural network language models: A generative approach to sentiment analysis [C]//*Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*. 2017: 1023-1032.
- [146] Baevski A, Edunov S, Liu Y, et al. Cloze-driven pretraining of self-attention networks [J/OL]. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019: 5360-5369. DOI: [10.18653/v1/D19-1539](https://doi.org/10.18653/v1/D19-1539).
- [147] Rosenfeld R, Chen S F, Zhu X. Whole-sentence exponential language models: a vehicle for linguistic-statistical integration [J]. *Computer Speech & Language*, 2001, 15(1): 55-73.
- [148] Chen S F. Shrinking exponential language models [C]//*Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. 2009: 468-476.
- [149] Amaya F A, Benedí J M. Improvement of a whole sentence maximum entropy language model using grammatical features [C]//*Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*. 2001: 10-17.

- [150] Wang B, Ou Z, Tan Z. Learning trans-dimensional random fields with applications to language modeling [J]. *IEEE transactions on pattern analysis and machine intelligence*, 2017, 40(4): 876-890.
- [151] Arisoy E, Sethy A, Ramabhadran B, et al. Bidirectional recurrent neural network language models for automatic speech recognition [C/OL]//2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2015: 5421-5425. DOI: [10.1109/ICASSP.2015.7179007](https://doi.org/10.1109/ICASSP.2015.7179007).
- [152] He T, Zhang Y, Droppo J, et al. On training bi-directional neural network language model with noise contrastive estimation [C/OL]//2016 10th International Symposium on Chinese Spoken Language Processing (ISCSLP). 2016: 1-5. DOI: [10.1109/ISCSLP.2016.7918423](https://doi.org/10.1109/ISCSLP.2016.7918423).
- [153] Chen X, Ragni A, Liu X, et al. Investigating bidirectional recurrent neural network language models for speech recognition [C]//Proceedings of Interspeech 2017. International Speech Communication Association (ISCA), 2017: 269-273.
- [154] Zhang Z, Han X, Liu Z, et al. Ernie: Enhanced language representation with informative entities [C]//Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019: 1441-1451.
- [155] Bai Y, Yi J, Tao J, et al. Listen Attentively, and Spell Once: Whole Sentence Generation via a Non-Autoregressive Architecture for Low-Latency Speech Recognition [C/OL]//Proc. Interspeech 2020. 2020: 3381-3385. <http://dx.doi.org/10.21437/Interspeech.2020-1600>.
- [156] Gu J, Bradbury J, Xiong C, et al. Non-autoregressive neural machine translation [C]//International Conference on Learning Representations. 2018.
- [157] Lee J, Mansimov E, Cho K. Deterministic non-autoregressive neural sequence modeling by iterative refinement [J]. 2018.
- [158] Wang Y, Tian F, He D, et al. Non-autoregressive machine translation with auxiliary regularization [C]//Proceedings of the AAAI Conference on Artificial Intelligence: volume 33. 2019: 5377-5384.
- [159] Guo J, Tan X, He D, et al. Non-autoregressive neural machine translation with enhanced decoder input [C]//Proceedings of the AAAI Conference on Artificial Intelligence: volume 33. 2019: 3723-3730.
- [160] Ma X, Zhou C, Li X, et al. Flowseq: Non-autoregressive conditional sequence generation with generative flow [J]. 2019: 4273-4283.
- [161] Chen N, Watanabe S, Villalba J, et al. Listen and fill in the missing letters: Non-autoregressive transformer for speech recognition [J]. *arXiv preprint arXiv:1911.04908*, 2019.
- [162] Lavrenko V, Manmatha R, Jeon J, et al. A model for learning the semantics of pictures. [C]//Nips: volume 1. Citeseer, 2003.

- [163] Jeon J, Lavrenko V, Manmatha R. Automatic image annotation and retrieval using cross-media relevance models [C]//Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval. 2003: 119-126.
- [164] Jiang Q Y, Li W J. Deep cross-modal hashing [C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 3232-3240.
- [165] Fan M, Wang W, Dong P, et al. Cross-media retrieval by learning rich semantic embeddings of multimedia [C]//Proceedings of the 25th ACM international conference on Multimedia. 2017: 1698-1706.
- [166] Yang Z, Lin Z, Kang P, et al. Learning shared semantic space with correlation alignment for cross-modal event retrieval [J]. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 2020, 16(1): 1-22.
- [167] Zhen L, Hu P, Wang X, et al. Deep supervised cross-modal retrieval [C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 10394-10403.
- [168] Audhkhasi K, Rosenberg A, Sethy A, et al. End-to-end asr-free keyword search from speech [J]. IEEE Journal of Selected Topics in Signal Processing, 2017, 11(8): 1351-1359.
- [169] Salazar J, Kirchhoff K, Huang Z. Self-attention networks for connectionist temporal classification in speech recognition [C/OL]//ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2019: 7115-7119. DOI: [10.1109/ICASSP.2019.8682539](https://doi.org/10.1109/ICASSP.2019.8682539).
- [170] Ding F, Guo W, Dai L, et al. Attention-based gated scaling adaptive acoustic model for ctc-based speech recognition [C]//ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2020: 7404-7408.
- [171] Zagoruyko S, Komodakis N. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer [C]//International Conference on Learning Representations. 2017.
- [172] Cho J H, Hariharan B. On the efficacy of knowledge distillation [C]//Proceedings of the IEEE International Conference on Computer Vision. 2019: 4794-4802.
- [173] Phuong M, Lampert C. Towards understanding knowledge distillation [C]//International Conference on Machine Learning. PMLR, 2019: 5142-5151.
- [174] Cheng X, Rao Z, Chen Y, et al. Explaining knowledge distillation by quantifying the knowledge [C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 12925-12935.
- [175] Prabhavalkar R, Sainath T N, Wu Y, et al. Minimum word error rate training for attention-based sequence-to-sequence models [C]//2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018: 4839-4843.

作者简历及攻读学位期间发表的学术论文与研究成果

作者简历

白焱，男，甘肃兰州人，1993年10月出生，中国科学院自动化研究所博士研究生。

2012.09-2016.06 中国农业大学信息与电气工程学院，通信工程专业，获得工学学士学位。

2016.09-2021.06 中国科学院自动化研究所，模式识别与智能系统专业，硕博连读，攻读博士学位。

以第一作者身份发表(或正式接受)的学术论文:

1. **Ye Bai**, Jiangyan Yi, Jianhua Tao, Zhengkun Tian, Zhengqi Wen, Shuai Zhang. Fast End-to-End Speech Recognition via Non-Autoregressive Models and Cross-Modal Knowledge Transferring from BERT. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, DOI: 10.1109/TASLP.2021.3082299. (语音、自然语言处理领域顶级期刊)
2. **Ye Bai**, Jiangyan Yi, Jianhua Tao, Zhengqi Wen, Zhengkun Tian, Shuai Zhang. Integrating Knowledge into End-to-End Speech Recognition from External Text-Only Data. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, DOI: 10.1109/TASLP.2021.3066274. (语音、自然语言处理领域顶级期刊)
3. **Ye Bai**, Jiangyan Yi, Jianhua Tao, Zhengqi Wen, Cunhang Fan. A Public Chinese Dataset for Language Model Adaptation. *J Sign Process Syst*, Vol. 92, 839–851, doi: 10.1007/s11265-019-01482-5. (*SCI* 期刊)
4. **Ye Bai**, Jiangyan Yi, Jianhua Tao, Zhengkun Tian, Zhengqi Wen, Shuai Zhang. Listen Attentively, and Spell Once: Whole Sentence Generation via a Non-Autoregressive Architecture for Low-Latency Speech Recognition. *Proc. Interspeech 2020*, 3381–3385, DOI: 10.21437/Interspeech.2020-1600. (语音领域顶级会议)
5. **Ye Bai**, Jiangyan Yi, Jianhua Tao, Zhengkun Tian, Zhengqi Wen. Learn Spelling

- from Teachers: Transferring Knowledge from Language Models to Sequence-to-Sequence Speech Recognition. Proc. Interspeech 2019, 3795-3799, DOI: 10.21437/Interspeech.2019-1554. (语音领域顶级会议)
6. **Ye Bai**, Jiangyan Yi, Jianhua Tao, Zhengqi Wen, Zhengkun Tian, Cunhang Fan. A Time Delay Neural Network with Shared Weight Self-Attention for Small-Footprint Keyword Spotting. Proc. Interspeech 2019, 2190-2194, DOI: 10.21437/Interspeech.2019-1676. (语音领域顶级会议)
 7. **Ye Bai**, Jiangyan Yi, Jianhua Tao, Zhengqi Wen, Bin Liu. Voice Activity Detection Based on Time-Delay Neural Networks. Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 1173-1178, doi: 10.1109/APSIPAASC47483.2019.9023262. (语音、信号处理领域国际会议)
 8. **Ye Bai**, Jianhua Tao, Jiangyan Yi, Zhengqi Wen, Cunhang Fan, CLMAD: A Chinese Language Model Adaptation Dataset. The 11th International Symposium on Chinese Spoken Language Processing (ISCSLP), 2018, 275-279, doi: 10.1109/ISCSLP.2018.8706600. (语音领域国际会议)
 9. **Ye Bai**, Jiangyan Yi, Hao Ni, Zhengqi Wen, Bin Liu, Ya Li, Jianhua Tao. End-to-end keywords spotting based on connectionist temporal classification for Mandarin. The 10th International Symposium on Chinese Spoken Language Processing (ISCSLP), 2016, pp. 1-5, doi: 10.1109/ISCSLP.2016.7918460. (语音领域国际会议)
 10. **白焯**, 易江燕, 陶建华, 温正棋. 基于随机时频掩蔽的 DNN-HMM 声学模型数据扩增. 第十五届全国人机语音通讯学术会议 (NCMMSC), 2019, 248-253 (语音领域国内会议)

申请或已获得的专利:

1. 温正棋, 白焯, 一种基于知识迁移的序列到序列语音识别模型训练方法. 公开号: 110459208A
2. 白焯, 温正棋, 基于前馈神经网络的低延时语音识别模型及训练方法. 公开号: 112133304A

参加的研究项目及获奖情况:

1. 国家重点研发计划项目, 音视频检测技术, 编号: 2017YFC0820602
2. 国家重点研发计划项目, 多模态自适应选择的高鲁棒协同感知, 编号: 2017YFB1002802
3. 国家重点研发计划项目, 移动办公模式下多模态深度融合的协同交互与反馈, 编号: 2018YFB1005003
4. 院双边合作项目, 基于小数据的强噪声语音识别声学模型研究, 编号: 173211KYSB20190049
5. 横向项目, 智能语音交互系统
6. 2018-2019 学年中科院三好学生
7. ISCSLP 2018 最佳学生论文提名
8. 2018 京东金融对话语音识别大赛第一名

致 谢

不知不觉，学位论文竟然已经写到了最后。回想 2015 年夏天，骑车来到自动化研究所参加夏令营申请读博，却又不知道做研究是什么意思。五年时间的博士研究生生涯即将结束，没有想法时的茫然、想法没有效果时候的无奈、获得新发现时候的喜悦，以及调试 bug 时的抓耳挠腮、修改论文时的字句斟酌、与人讨论时的精神振奋都一一倏忽而过，留下来的是对自我的发现和重新认识。我衷心地感谢所有帮助过我的老师，同学，朋友，和亲人。

感谢我的导师陶建华研究员。六年前，陶老师指引我进入语音领域的大门。入学后，研究方向的选择以及毕业论文的选题写作，陶老师投入了大量心血。陶老师创造了一个宽松自由的环境来培养我，还提供机会让我能够参加会议与国际同行交流。生活上，陶老师的关心和照顾让我能够将精力投入学术研究中。陶老师高效的科研管理风格、勤奋的工作态度、儒雅的人格气度潜移默化地影响着我，并将使我受益终身，而陶老师开阔的视野、前瞻的目光、敏锐的洞察则是我一生学习的榜样。在此，向陶老师表达我最衷心的感谢。

感谢模式识别国家重点实验室的宗成庆老师，赵军老师、刘文举老师、张家俊老师、刘康老师，感谢他们为我的开题和中期报告提出了宝贵的建议。感谢所研究部的曹娟、张志琳、郭静、郑璐、赵瑾等老师，以及实验室综合办公室的连国臻、赵微、王爱华、周晓旭等老师，感谢他们在学习和生活中给予我的帮助。感谢自动化所的领导，特别是牟克雄书记在 2021 年给滞留北京学生的年夜饭让我倍感温暖。

感谢易江燕老师对我科研的指导和帮助。每一篇论文与报告，易老师对逻辑的检查、逐字逐句的文字修改都让我受益匪浅，让我知道如何工作，如何表达。第一次投顶会论文时，易老师画满四页纸的红笔印记犹在眼前。易老师在工作时的认真细致、为人处事时的真诚谦和让我受益良多。感谢温正棋老师对我技术上的指导。记得刚刚进组实习时晚上开会，温老师打开投影仪跟我们讲解解码器的源代码，努力时光令人难忘。还要感谢温老师帮我修改第一次写作的英文论文，几乎可以说是全文翻新一遍，最后论文中稿 ISCSLP 会议，我第一次体会中论文的感觉。感谢刘斌老师的帮助，每一次和刘老师讨论问题，无论是行业还是

学术，都能有新的启发。

在智能交互团队的五年里，亲眼见证了团队的茁壮发展。在这个活泼而勤奋、宽松而进取的集体里，我倍感自豪。感谢智能交互团队的柳雪飞、李永伟、张大伟、梁山、聂帅等老师，以及李雅、戚肖克、杨明浩等老师对我学习、工作和生活上的帮助和照顾。感谢刘燕、刘瑞祺、郑雪英等对我学习、工作和生活上的支持。感谢语音/情感小组，每周的学术讨论激荡着思维的火花。五百年修得同船共度，感谢范存航、连政、牛明月，作为同窗一同奋斗直至毕业。感谢田正坤、张帅，一起讨论研究语音识别技术的时光特别快乐，思维的碰撞和高效的执行令人激动。感谢巢林林师兄、丁星光师兄、方硕师姐、方祥师兄、郑艺斌师兄、倪浩师兄、黄健师兄、傅睿博师兄、赵博程师兄，能沿着你们的道路学习我倍感荣幸；感谢汪涛、车飞虎、王成龙、马浩鑫、钟荣秀、孙立才、赵呈昊、赵冬梅、蔡聪等团队所有同学，和你们一同奋斗的读书时光让人倍感难忘。感谢车浩师兄在语音行业与职业方面的指点，让人受益匪浅。

感谢 Daniel Povey 博士和他杰出的 KALDI 语音工具箱，我从中学习到各种语音基础知识和编程技巧。感谢李锦宇老师在学术上和技术上的鼓励与指点。感谢 Chiori Hori 老师，在 Interspeech 会议上的交流让人深受启发。感谢何彦璋、王育军、Edward Lin、张华师姐、Julien Lai、万广鲁、马泽君师兄、何怡、黄申师兄、胡鹏飞师兄、康健、苏牧师兄、王晓瑞师兄、李杰师兄、赵媛媛师姐、黄东延老师等语音领域的前辈、师兄师姐在技术、行业发展方面的指点。

感谢汪定老师在研究和工作上的指点，每一次向汪老师学习与交流都获益颇丰。感谢研究生阶段的张灏、向世明、彭思龙、张煦尧、王晓、高随祥、孟钢等老师让我能领略前人知识的美，提升学术品味。感谢本科生阶段的李国辉、王建平、温勃婴、李俐等老师帮我本科知识的基础。感谢范益民、李万祥、单小虎、邵立强等中学和小学老师教我常识与方法。感谢老师们，是您让我学会学习与做人。

感谢吴雅儒、Iona Gessinger 等 ISCA-SAC 的伙伴，大家一同努力，成功地举办 INTERSPEECH 2020 学生活动是十分难忘的记忆。

感谢我的朋友们。感谢张颐康，和康神在一起玩耍与讨论总能让人感到快乐感到佩服。感谢贾桐、俞剑文，一起学习、玩耍、吹牛神侃特别欢乐。感谢朱玉帛、乔鹏、管宏伟、崔向元，一起爬山喝酒，谈天说地，让人可以忘掉烦恼。感谢

李星熠、罗宗海，一起玩耍、讨论 IT 行业。感谢胡宇星、郑锋、杨文添、吕佳伟，儿时热爱讨论的习惯保持到了现在。感谢郭建珠、隋典伯、曹鹏飞、白赫、张家斌等朋友一起讨论学术与生活。感谢我的室友白桂荣，一块探讨人生 novelty 和 motivation 最欢乐。

感谢我的家人们。感谢舅舅一家对我的照顾和支持。感谢二姨一家对我的挂念。感谢叔叔们姑姑们对我的关心。

需要感谢的人实在太多，片纸不足表达，只能铭记心头。

回头想来，研究像是一场旅行，最开心的是埋头走路的时候偶尔看见了一点儿新的风景。这本博士论文是我的游记，我把它献给我的爸爸妈妈和姥爷，他们无尽的爱让我变得勇敢和坚定，能够走完这段路。学习是一辈子的事情，行囊背起来，又该继续向前走了。

