摘要

车载语音交互已经成为继通讯社交、智能家居之后的第三大应用场景。随着深度学习的发展,作为智能语音交互的核心,自动语音识别和自然语言理解技术取得了重大进展。然而,深度学习模型的高精度运行需消耗庞大的计算资源,而

受制于车身结构和研发成本,当前车载语音对话系统均采用"云—端"方式运行,

存在着一定的数据安全隐患。为解决这一问题,论文提出了基于残差分组线性变换解码器的自动语音识别模型和基于标签感知图交互的自然语言理解模型,在满足模型性能要求的前提下降低了运行所需的计算资源,并搭建了面向嵌入式设备的本地智能语音对话系统,实现了数据安全、自然实时的离线智能语音交互。论

文的主要工作及创新点如下:

1.针对基于深度编—解码器的自动语音识别模型参数量庞大的问题,提出了一种基于残差分组线性变换的解码器结构。该结构关键模块为"钻石"型缩放单元,其内部采用稀疏连接,同一组神经元共享相同的权重矩阵。所提模型在AISHELL-1数据集上的参数量和计算量分别为20.4M 和5.3B,错误率为6.67%;在 TED-LIUM2数据集上的参数量和计算量分别为20.3M 和4.6B,错误率为

11.86%,在模型识别性能和轻量化程度方面优于其他对比方法。

2.针对基于显式联合建模的自然语言理解模型交互能力不足的问题,提出了一种基于标签感知的图交互模型,其中标签映射模块可以获取原始话语与标签语义之间的相关性以提供丰富的先验知识,全局图交互模块可以对语句级别的意图—槽位交互过程进行建模以提供全局优化。所提模型在 MixATIS 和 MixSnips 数据集上的整体准确率分别为49.9%和77.3%,优于其他对比方法。

3.针对"云—端"方式运行的车载智能语音对话存在数据安全隐患的问题,搭建了面向车载嵌入式设备的本地智能语音对话系统。首先选取了 Nvidia Jetson TX2作为车载嵌入式设备,然后根据实际应用场景收集、创建了驾驶数据集

CQUPT-DS,接着将上述提出的两个模型在驾驶数据集上进行了训练,最后集成、

移植网络模型至 TX2并围绕搭建了全套硬件平台,实现了数据安全、自然实时的离线智能语音对话。整套系统的对话通过率为97%,平均响应时间为0.87s。

关键字:车载语音交互,自动语音识别,自然语言理解,嵌入式设备,对话系统

ABSTRACT

In-vehicle voice interaction is the third largest application scenario after social communication and smart home. With the development of deep learning, as the core of intelligent voice interaction, automatic speech recognition and natural language understanding technology has made significant progress. However, the high-precision operation of the deep learning model consumes vast computing resources. Due to the structure of the vehicle body and the cost of research and development, the current in-vehicle voice dialogue system is operated in a "cloud-end" mode, which has certain hidden dangers in data security. To solve this problem, the paper proposes an automatic speech recognition model based on a residual group linear transform decoder and a natural language understanding model based on label-aware graph interaction, which reduces the computing resources required for operation while meeting the model performance requirements. The paper also builds a local intelligent voice dialogue system based on embedded devices, realizing data security, natural real-time offline intelligent voice

interaction. The main work and innovations of the paper are as follows:

1. A decoder structure based on a residual group linear transform is proposed to address the problem of many parameters in the automatic speech recognition model based on a deep encoder-decoder. The key module of this structure is a "diamond" type scaling unit, which uses sparse connections inside, and the same group of neurons shares the same weight matrix. The parameters and calculations of the proposed model on the AISHELL-1 dataset are 20.4M, and 5.3B, respectively, and the error rate is 6.67%; the parameters and calculations on the TED-LIUM2 dataset are 20.3M and 4.6B, respectively, with an error rate of 11.86%, outperforming other comparative methods in terms of model recognition performance and lightweight.

2. Aiming at the insufficient interaction ability of natural language understanding models based on explicit joint modeling, a label-aware graph interaction model is proposed, which contains label injection module and global graph interaction module. The former can obtain the correlation between the original utterance and label semantics to provide rich prior knowledge, and the latter can model the intent-slot interaction process at the sentence level to provide global optimization. The overall accuracy of the

proposed model on the MixATIS and MixSnips datasets is 49.9% and 77.3%, respectively, outperforming other comparison methods.

3. A local intelligent voice dialogue system based on in-vehicle embedded device was built to aim at the hidden dangers of data security in the in-vehicle intelligent voice dialogue operating in the "cloud-end" mode. First, Nvidia Jetson TX2 was selected as the in-vehicle embedded device, and then the driving data set CQUPT-DS was collected and created according to the actual application scenario. Then the two models proposed above were trained on the driving data set, and finally, the network was integrated and transplanted. The model is up to TX2, and a complete hardware platform is built around it, realizing data security and natural real-time offline intelligent voice dialogue. The dialogue pass rate of the whole system is 97%, and the average response time is 0.87s.

Keywords: In-vehicle Voice Interaction, Automatic Speech Recognition, Natural

Language Understanding, Embedded Device, Dialogue System

目 录

摘 要 I ABSTRACT II

目 录 IV 第 1 章 绪论 1

1.1 研究背景及意义 1

1.2 国内外研究现状 2

1.2.1 自动语音识别技术 2

1.2.2 自然语言理解技术 4

1.3 论文研究内容 6

1.4 论文组织结构 7

第 2 章 对话系统基础理论 9

2.1 对话系统基本组成结构 9

2.2 神经网络基础理论 10

2.2.1 前馈神经网络 10

2.2.2 卷积神经网络 11

2.2.3 循环神经网络及其变种 11

2.3 基于编—解码器结构的自动语音识别 13

2.4 基于显式联合建模的自然语言理解 15

2.5 本章小结 16

第 3 章 基于残差分组线性变换解码器的自动语音识别 17

3.1 引言 17

3.2 Transformer 模型及其组件 17

3.2.1 自注意力机制 18

3.2.2 前馈网络层 19

3.2.3 位置编码 19

3.3 基于残差分组线性变换的"钻石"型缩放单元 20

3.3.1 分组线性变换 20

3.3.2 "钻石"型缩放单元 21

3.4 基于"钻石"型缩放单元的改进 Transformer 模型 23

3.4.1 轻量级前馈网络层 23

3.4.2 轻量级解码器 23

3.4.3 改进的 Transformer 模型结构 24

3.5 实验结果及分析 25

3.5.1 实验环境、数据集和评价指标 25

3.5.2 数据预处理 28

3.5.3 训练配置信息及过程 29

3.5.4 对比实验结果及分析 30

3.5.5 消融实验结果及分析 32

3.6 本章小结 34

第 4 章 基于标签感知图交互的自然语言理解 35

4.1 引言 35

4.2 基于最佳线性逼近的标签映射模块 35

4.2.1 最佳线性逼近 35

4.2.2 标签映射模块 36

4.3 基于图注意力网络的全局图交互模块 37

4.3.1 图注意力网络 37

4.3.2 全局图交互模块 39

4.4 基于标签感知的图交互模型 41

4.4.1 共享编码器 41

4.4.2 意图解码器 41

4.4.3 槽位解码器 42

4.5 实验结果及分析 42

4.5.1 实验环境、数据集和评价指标 42

4.5.2 数据预处理 43

4.5.3 训练过程及配置信息 44

4.5.4 对比实验结果及分析 46

4.5.5 消融实验结果及分析 48

4.6 本章小结 50

第 5 章 面向车载嵌入式设备的本地智能语音对话系统 51

5.1 引言 51

5.2 嵌入式设备运行环境搭建 51

5.3 驾驶数据集收集 53

5.3.1 数据收集平台 53

5.3.2 数据收集过程 54

5.4 模型训练与移植 55

5.4.1 自动语音识别模型训练 55

5.4.2 自然语言理解模型训练 57

5.4.3 模型集成与移植 58

5.5 系统硬件平台搭建与测试 59

5.5.1 平台搭建 59

5.5.2 系统测试 60

5.6 本章小结 60

第 6 章 总结与展望 61

6.1 总结 61

6.2 展望 62

参考文献 63

作者简介 71

1. 基本情况 71

2. 教育和工作经历 71

3. 攻读学位期间的研究成果 71

3.1 发表的学术论文和著作 71

3.2 申请(授权)专利 71

3.3 参与的科研项目及获奖 71

致 谢 73

1.1研究背景及意义

语音交互作为人机通信中最自然、直接的交互方式,具有天然的优势[1]。随着智能汽车的兴起,车载语音交互成为继通讯社交、智能家居之后的第三大应用场景。截止2020年9月,我国乘用车车载语音装配率为64.8%,预计2025年国内前装车载语音市场规模约32亿元,智能语音交互俨然成为汽车的标准化配置之

一[2]。科大讯飞、亚马逊和谷歌等企业也相继推出了自主研发的车载语音对话系

统,对车载语音市场的发展进程产生了显著影响[3]。自动语音识别(Automatic

Speech Recognition,ASR)和自然语言理解(Natural Language Understanding,NLU)是智能语音交互的核心技术。在深度学习的驱动下,当前的 ASR 和 NLU

技术取得了重大进展,模型性能得到极大提升[4]。

然而,高精度的深度学习模型需要消耗庞大的计算资源才能快速运行。受限于车身结构和研发成本,当前主流的车载语音对话系统均采用"云—端"方式运行,即通过互联网将本地收集到的音频数据传输至配置有高性能图形处理器(Graphics Processing Unit,GPU)的云服务器进行处理,再将处理结果通过网络反馈到本地端,最后由汽车中控系统进行相应操作。事实上,此种运行方式在传输过程和云端处理过程中存在着一定的数据安全隐患,例如特斯拉间谍车、滴滴地图数据外泄等事件。为解决上述问题,实现深度学习下的 ASR 和 NLU 技术在车载平台上的高可靠性、强实时性应用,研发离线条件下的智能语音对话是有效的技术途径。

论文面向计算资源有限的车载嵌入式设备,立足于语音识别和自然语言处理领域中已有的先进成果,重点研究针对 ASR 和 NLU 的技术创新应用方法,开发相应的轻量级别神经网络模型,在满足性能要求的前提下降低其运行所需的计算资源,探索以离线方式运行的车载智能语音对话平台搭建,在本地实现数据安全、自然实时的语音对话。论文提出的语音对话方法可以推广到智能家居、智慧教育和智慧医疗等其他智能语音交互领域,丰富相关领域的技术理论,实现语音交互的价值。特别地,除了具体的 ASR 和 NLU 技术外,论文中涉及的一些关键共性技术,如模型轻量化、嵌入式移植等,亦可为研发其他依靠深度神经网络的设备提供成熟的案例支撑。

重庆邮电大学硕士学位论文

1.2国内外研究现状

1.2.1自动语音识别技术

自动语音识别技术的发展过程可以分为三个阶段:第一阶段主要使用高斯混合模型-隐马尔可夫模型(Gaussian Mixture Model-Hidden Markov Model,GMM-

HMM)进行建模,基于 GMM-HMM 的语音识别框架在上世纪得到广泛应用[5];

直至2009年,Hinton 等人首次将深度信念网络(Deep Belief Neiwork,DBN)应

用于声学建模并提出了深度信念网络-隐马尔可夫模型(DBN-HMM)[6],在

TIMIT 的核心测试集上达到了23.0%的音素错误率(Phoneme Error Rate,PER),

这种方法促进了深度学习在 ASR 领域的应用,大量研究人员投入到深度学习研究

中,并取得较大的成果[7]。至此,GMM-HMM 框架被打破,进入到以深度神经网

络-隐马尔可夫模型(Deep Neural Network- Hidden Markov Model,DNN-HMM)为主要方法的第二阶段;第三阶段是端到端(End-to-End,E2E)的时代,基于

E2E 的语音识别普遍采用深度学习的方法,它不需要提前进行语音帧对齐,而是直接采用带标签的语音进行训练。与经典方法相比,它更加简洁且具有较强的通用性,能够减少对专业语音、语言知识的依赖,大大降低了系统搭建难度。总体可分为两类:一类是基于联结时序分类(Connectionist Temporal Classification,CTC)[8]的 E2E 模型,另一类是基于注意力机制(Attention)的序列到序列(Sequence-to-Sequence,S2S)模型[9],二者模型结构如图1-1所示。CTC 方法不

像经典方法需要对语音数据进行对齐操作,只需计算输出序列和真实序列的差距;而 S2S 方法在2017年被 Google 用于语音识别领域,并取得了非常好的效果。

图1-1基于 CTC 的 E2E 模型(左)和基于 Attention 的 S2S 模型(右) Fig.1-1 CTC-based E2E model (left) and Attention-based S2S model (right)

基于 CTC 的 E2E 模型结构如图1-1左侧所示。在无先验性对齐情况下,该模

型能够度量输入和输出序列的相似度,并且能刻画语音特征和字符序列的相关性。

基于联结时序分类的语音识别由卷积神经网络(Convolutional Neural Network,CNN)/循环神经网络(Recurrent Neural Network,RNN)编码模块和 CTC 损失函数模块组成。2006年 Graves 等人利用空白字符对不同长度的序列进行对齐,首次提出 CTC 模型来解决 MINIST 数据集上的手写数字识别和 TIMIT 数据集上的语料

库音素分类的问题[10],并在 TIMIT 数据集上实现了30.51±0.19的标签错误率

(Label Error Rate,LER)。2012年 Graves 对 CTC 语音识别模型进一步改进,将RNN 作为转化模型[11],在 TIMIT 数据集上实现了23.2%的 PER。2013年,Graves等人利用多层的长短时记忆(Long Short-Term Memory,LSTM)神经网络进行建

模,进一步提升 TIMIT 语料库上的识别效果[12],达到了17.7%的 PER。

基于 Attention 的 S2S 模型结构如图1-1右侧所示。该模型主要由编码器(Encoder)、注意力网络和解码器(Decoder)三个部分组成,在2013年由

Graves 首次提出,并被用于 MINIST 数据集的手写数字识别任务中[13]。2014年,Bahdanau等人在此基础上进一步完善,提出基于Attention的编—解码模型[14]。同

年,受到上述两项工作的启发,Chorowski 等人将注意力模型进一步应用于语音

识别任务中[15],使用注意力机制建立输入和输出序列之间的对齐关系,并在目标

函数中加入约束,最后在 TIMIT 的验证集和测试集上的 PER 分别为16.88%和

18.57%。至此,大量的研究人员开始投入到基于 Attention 的语音识别研究中,并提出多种改进模型。根据改进方法可分为三类:对 Encoder 的改进、对 Attention的改进和对语言模型的改进。对 Encoder 的改进方面,Bahar 等人采用更复杂的二维长短时记忆网络(Two-Dimensional Long Short Term Memory,2DLSTM)作为

编码模型,从多维的角度增加语音时序信息的辨认能力[16]。近几年,研究者们热

衷于研究基于深度编—解码器(Transformer)的 ASR 系统,Transformer 最先由

Vaswani 等人提出[17],并被用于建立语言模型和机器翻译领域,由于 Transformer

采用全连接网络,因而相较于 RNN 等结构,其训练效率高,模型收敛效果好[18];

对 Attention 的改进方面,Merboldt 等人提出局部注意力机制,对其添加约束,同时还考虑最大注意力得分,之后采用启发式搜索对模型进行训练,最终在

SwitchBoard 和 LibriSpeech 数据集上取得较好的效果[19];对语言模型的改进方面,

除了传统的语言模型,Zeyer 等人采用 LSTM 作为语言模型,并采用字节对编码(Byte Pair Encoding,BPE)输出不同的识别基元,相比于传统语言模型,取得

了较好的效果[20]。

重庆邮电大学硕士学位论文

综上所述,基于 CTC 的 E2E 模型存在解码过程复杂、速度慢等缺点,而基于Attention 的 S2S 模型具有模型简单、联合训练、直接输出和无需强制数据对齐等优点。但是,该类模型存在参数量庞大、计算复杂、难以部署等缺点。例如,谷歌提出的 Conformer 模型参数量达到了1.19亿[21]。这些因素为其在实际工程中的应用带来了巨大的挑战。因此,如何降低模型的参数量和计算复杂度、提高模型

运行速度,在计算资源有限的车载平台上实现高效的语音识别是亟待解决的问题,也是学术界当前研究的热点。

1.2.2自然语言理解技术

自然语言理解通常包含意图检测(Intent Detection,ID)和槽位填充(Slot

Filling,SF)两个子任务,前者属于文本分类(Text Classification)领域,而后者属于序列标注(Sequence Labeling)领域。自然语言理解技术的发展过程从建模

方法上可以分为两个类别:基于独立建模的方法和基于联合建模的方法[22]。

基于独立建模的方法如图1-2(a)所示。在意图检测方面,Ravuri 和 Stolcke 在

2015年成功地将 RNN 和 LSTM 应用到意图检测中,表明序列特征有利于提升准

确率[23];在槽位填充方面,常用的方法包括条件随机场(Conditional Random

Fields,CRF)、RNN 和基于 RNN 的衍生模型。2013年,Yao 采用基于 RNN 的

语言模型来预测槽位标签,此外还研究了命名实体、句法特征和词类信息等[24]。

2013年,Mesnil 研究了不同的用于 NLU 的 RNN 衍生模型,包括 Elman RNN、Jordan RNN 和双向 Jordan RNN[25]。下一年,Mesnil 利用 Viterbi 编码方式和循环

CRF 层消除了标签的预测偏差问题[26]。基于独立建模的方法虽然取得了不错的效果,但由于单独训练,单个模型的意图检测任务和槽位填充任务之间没有交互作用,存在共享信息泄漏,导致模型性能不足。

图1-2 NLU 模型的三种建模方式。(a)独立建模;(b)隐式联合建模;(c)

显式联合建模

Fig.1-2 NLU modeling methods.(a) Independent modeling;(b) Implicit joint

modeling;(c) Explicit joint modeling

基于联合建模的方法从两个子任务的交互方式可分为隐式交互和显式交互,如图1-2(b)和(c)所示。隐式联合建模采用共享编码器(Shared Encoder)来捕获特征。例如,Zhang 和 Wang 在2016年引入了共享 RNN 模型来获取意图和插槽之

间的相关性[27]。同年,Hakkani-Tur 等人提出了一种用于联合建模的共享 RNN-

LSTM 架构[28]。隐式联合建模是一种直接整合共享信息的方法,但后续未对交互关系进行显式建模,导致模型可解释性不足,性能无法达到预期;而显式联合建模在捕获特征后依旧通过交互模块进行交互,可以充分共享信息,同时具有控制交互过程的特性,所以近年来越来越多的研究者提出基于显式联合建模的方法。Goo 等人提出 Slot-Gated 模型[29],该模型允许槽位填充可以根据学习到的意图设

置条件,在ATIS和Snips两个数据集上的整体准确率分别达到了82.6%和74.6%。Li 等人提出了一种新的 Self-Attention 模型[30],通过意图来引导槽位填充。Qin 等人提出 Stack-Propagation 模型[31],直接使用意图检测结果来指导槽位填充,并使

用字符级别(Token-level)信息来缓解二者间的误差传递,在 ATIS 和 Snips 两个数据集上分别达到了86.5%和86.9%的整体准确率。Niu 等人提出了一种新颖的

SF-ID 模型[32],为意图和槽位提供了双向关联机制,在 ATIS 和 Snips 两个数据集

上实现了86.90%和80.57%的整体准确率。Zhang 等人引入动态路径胶囊网络

(Capsule-NLU)[33],将两个任务之间的分层和相关关系纳入其中,在 ATIS 和

Snips-NLU 两个数据集上分别实现了83.4%和80.9%的整体准确率。Liu 等人提出

了一种新型的协同记忆网络(CM-Net)用于联合建模意图检测和槽位填充[34],

在 ATIS 和 Snips 两个数据集上的意图准确率、槽位 F1值分别为99.10%、96.20%、

99.29%和97.15%。Zhang 等人将 Graph(图)-LSTM 结构引入 NLU 任务中[35],在 ATIS 和 Snips 两个数据集上取得了很好的性能,整体准确率分别为87.57%和

89.71%。Qin 等人通过建立两个任务之间的双向连接,提出了一种考虑交叉影响的 Co-interactive Transformer 模型[36],在 ATIS 和 Snips 两个数据集上达到了87.4%和90.3%的整体准确率。随着计算资源的不断增强,以 BERT(Bidirectional Encoder Representation from Transformers)[37]为代表的各种预训练模型(Pre-

trained Language Models,PLMs)在 NLP 任务中取得了振奋人心的结果。例如,Chen 等人使用 BERT 提取共享文本特征用于意图检测和槽位填充,在原始模型的

基础上取得了显著的性能提升[38]。

综上所述,基于隐式联合建模的方法只是通过共享字符编码来隐式地考虑两个任务之间的相互连接,而基于显式联合建模的方法构建了交互模块,子任务间存在信息交互通道,使得模型性能有了进一步的提高。但是该类模型在交互模块中没有高效的交互信息融合手段,导致模型交互能力不足。虽然可以在特征提取

重庆邮电大学硕士学位论文

阶段引入 PLMs 来补充丰富的语义特征,但 PLMs 的参数量往往非常惊人。以谷

歌推出的基础模型Bert-base-uncased[37]为例,其网络层数为12,隐藏层维度为768,

拥有12个注意力头,参数量达到1.1亿,如此庞大的模型难以部署于车载平台(嵌入式设备)上。因此,如何在不引入 PLMs 的前提下,构建意图和槽位的双向交互通道,高效融合所有的交互信息,深入挖掘跨任务的语义特征,提升模型的性能,在计算资源有限的车载平台上实现高效的对话理解是亟待解决的问题,也是学术界当前研究的热点。

1.3论文研究内容

针对上述问题,论文面向计算资源有限的车载嵌入式设备,重点研究针对ASR 和 NLU 的技术创新应用方法。首先提出了一种基于残差分组线性变换解码器的 ASR 模型,通过在解码器中引入残差分组线性变换,实现模型参数量和计算复杂度的大幅降低;然后提出了一种基于标签感知图交互的 NLU 模型,通过在交互模块中引入标签映射模块和全局图交互模块,实现模型交互能力和预测精度的提高;最后将 ASR 模型和 NLU 模型集成、移植至车载嵌入式设备上,根据实际

用车环境围绕搭建全套硬件平台,实现数据安全、自然实时的离线智能语音对话。

具体涉及以下研究内容:

(1)基于残差分组线性变换解码器的自动语音识别

针对基于深度编—解码器的自动语音识别模型参数量庞大的问题,提出了一种基于残差分组线性变换的解码器结构,该结构关键模块为"钻石"型缩放单元,其内部采用稀疏连接,同一组神经元共享相同的权重矩阵,实现模型参数量和计算复杂度的降低。

(2)基于标签感知图交互的自然语言理解

针对基于显式联合建模的自然语言理解模型交互能力不足的问题,提出了一种基于标签感知的图交互模型,其中标签映射模块可以获取原始话语与标签语义之间的相关性以提供丰富的先验知识,全局图交互模块可以对语句级别的意图—槽位交互过程进行建模以提供全局优化,实现模型交互能力和预测精度的提高。

(3)面向车载嵌入式设备的本地智能语音对话系统

针对"云—端"方式运行的车载智能语音对话存在数据安全隐患的问题,搭建了面向车载嵌入式设备的本地智能语音对话系统。具体地,首先选取 Nvidia Jetson TX2作为车载嵌入式设备并进行刷机、配置环境等操作,然后根据实际应用场景收集、创建驾驶数据集,接着将研究内容(1)和(2)的模型在驾驶数据集上进行训练,最后集成、移植网络模型至 TX2并围绕搭建全套硬件平台,实现数据安全、自然实时的离线智能语音对话。

1.4论文组织结构

论文共分为六章,如图1-3所示,各章节安排如下:

图1-3论文组织结构

Fig.1-3 Structure of the thesis

第一章,绪论。首先阐述论文的研究背景及意义,然后介绍国内外对于自动语音识别和自然语言理解领域的研究现状,针对当前存在的一些关键科学问题,

列出论文的研究内容并展示组织结构;

第二章,对话系统基础理论。首先介绍对话系统基本组成结构,同时引入深度学习领域内相关基础理论,然后在此基础上详细叙述自动语音识别和自然语言

理解的主流建模方法,为后续章节内容进行铺垫;

第三章,基于残差分组线性变换解码器的自动语音识别。首先基于残差分组线性变换构建"钻石"型缩放单元,然后基于缩放单元搭建改进的编—解码器模

重庆邮电大学硕士学位论文

型,实现参数量和计算复杂度的大幅度降低,最后通过一系列对比实验和消融实验论证所提模型的有效性。

第四章,基于标签感知图交互的自然语言理解。首先基于最佳线性逼近思想构建标签映射模块,然后基于图注意力网络搭建全局图交互模块,接着融合上述两个模块形成基于标签感知的图交互模型,实现交互能力和预测精度的提高,最后通过一系列对比实验和消融实验论证所提模型的有效性。

第五章,面向车载嵌入式设备的本地智能语音对话系统。首先对车载嵌入式设备进行选型,然后根据实际应用场景收集、创建驾驶数据集,接着将前文提出的自动语音识别模型和自然语言理解模型在驾驶数据集上进行训练,最后将训练好的网络模型集成、移植至车载嵌入式设备上并围绕搭建硬件平台,并完成系统测试以论证其有效性。

第六章,总结与展望。总结论文研究内容,并对后续研究工作做出展望。

Equation Chapter 2 Section (Next)

第2章对话系统基础理论

2.1对话系统基本组成结构

对话系统历经半个多世纪的发展,取得了很大的研究成果。现有对话系统按类型主要可分为闲聊型对话系统(Chatbot)和任务驱动型对话系统(Task-oriented Dialogue System)两类[39]。此外,按照对话系统回复时需考虑的轮数还可以将其分为单轮对话系统和多轮对话系统。值得注意的是,不管是闲聊对话系统还是任务驱动型对话系统都可以考虑单轮或多轮对话,而且在实际应用中,二者经常混合出现,并非是截然分离的[40]。

早年的对话系统大多使用手动特性或人工设计的规则来表示用户状态,这种方法不仅耗费人力,而且限制了对话系统的扩展应用。近年来,研究人员热衷于研究基于深度学习的对话系统,其通过学习特征的高维表示来缓解这些问题。目前,基于深度学习的任务型对话系统有两种方法:流水线(Pipeline)方法端到端

(End-to-End)方法[41]。前者是目前业界的主流方法,而后者由于存在不可控性

和不可解释性等问题,当前还处于探索阶段[42]。

基于 Pipeline 方法的对话系统如图2-1所示,主要包含自动语音识别(ASR)、

自然语言理解(NLU)、对话状态跟踪(Dialogue State Tracking,DST)、对话策略学习(Dialogue Policy Learning,DPL)、自然语言生成(Nature Language Generation,NLG)、语音合成(Text-to-Speech,TTS)等6个模块。其中,ASR将用户语音转换为系统可接收的文本输入,NLU 将文本输入解析为包含意图和槽位的结构化数据。它们为下游模块提供了必要的信息,同时也是对话系统的重要支撑,故论文主要研究针对 ASR 和 NLU 的技术创新应用方法。

图2-1基于 Pipeline 方法的对话系统

Fig.2-1 Dialogue system based on pipeline method

重庆邮电大学硕士学位论文

2.2神经网络基础理论

常见的 ASR 和 NLU 建模方法是基于深度学习,使用不同类型的神经网络进行各种拓扑组合,从而提升模型的各种性能。鉴于此,在介绍常见的ASR和NLU建模方法之前,本节先介绍在其中常用的前馈神经网络、卷积神经网络和循环神经网络及其变种。

2.2.1前馈神经网络

前馈神经网络(Feedforward Neural Network,FNN)[43]是最简单的神经网络之一,同时也是卷积神经网络和循环神经网络的基本组成单元。以经典的三层前馈神经网络为例,其结构如图2-2所示。所有神经元按层组织排列,第一层称为输入层,中间层称为隐藏层,最后一层称为输出层。前一层的每个神经元会向后一层所有与之相连接的神经元传输信号。由于前、后层的神经元都是"全连接"的,故前馈神经网络也称为全连接神经网络。

图2-2三层前馈神经网络结构

Fig.2-2 Structure of three-layer FNN

FNN 进行信息前向传播的方式如下:

(2-1)

式中,表示当前层数;表示第层神经元的输出向量;表示第层到第层的权重矩阵;表示第层神经元的激活函数;表示第层的偏置向量。如果引入更多层数,FNN 能更好地对"输入—输出"关系进行建模。本质上,

对神经网络进行训练就是通过一些特定的算法不断更新网络参数和,使得其

在某一组参数配置下对于输入的拟合能力达到最优。目前最常见的参数更新算法是反向传播(Back Propagation,BP)算法[44]。BP 算法首先根据输出结果与参考结果的偏差计算损失函数的值,然后使用链式法则从输出层往输入层方向逐层求取各参数梯度,最后将学习率与所求梯度相乘以更新参数。

2.2.2卷积神经网络

卷积神经网络[45]是一种特殊的 FNN,具有局部连接和权值共享等特点。相较

于 FNN,它能以较少的参数量实现对局部特征的提取,其中一个简单的卷积操作如图2-3所示。其中卷积滤波器又称为卷积核,是一组可学习权值的集合,主要作用是将输入图像中的一个小区域像素加权平均为输出图像中的每个对应像素。

首先卷积核在输入图像上按照特定的方向移动,扫描视野范围内的像素信息进而提取特征。每个卷积核提取得到的特征是尺寸更小的单通道图片,即为特征图;然后在非线性激活函数的作用下,对每张特征图进行处理并对得到的激活特征图按顺序进行通道维度上的堆叠,即可得到最后的输出结果。通常,在进行卷积操作后,还会使用池化操作对特征图进行裁剪,进一步压缩特征图大小。通过对卷积核数、尺寸等超参数进行合适的设置,可实现CNN对图像信息特征的扩展与丰富。

图2-3卷积神经网络中的卷积操作

Fig.2-3 Convolution operation in CNN

2.2.3循环神经网络及其变种

循环神经网络[46]是一种处理序列数据或时序数据的递归神经网络,其网络拓

扑结构在时间维度上以递归形式展开,如图2-4所示。

重庆邮电大学硕士学位论文

图2-4循环神经网络拓扑结构

Fig.2-4 Topology of RNN

RNN 的一个主要特点是:在前一时间步计算得到的隐藏状态会作为当前时间步的输入。在时间步,RNN 接收当前时间步的输入和前一时间步的隐藏状态

,产生当前时间步的输出,计算过程如下:

(2-2)

(2-3)

上式中,为作用于前一时间步隐藏状态的权重矩阵;为作用于输入的权重矩阵;为作用于输出的权重矩阵;和均为非线性激活函数。 RNN 的结构决定了其在处理时序信息上具有天然的优势。然而,RNN 在反向传播过程中产生的梯度会随着时间推移逐渐放大或衰减。这一现象在RNN处理具有较长时间依赖性的时间序列尤为明显,其梯度放大或衰减的程度趋近于指数级,即梯度爆炸和梯度消失。为了缓解这一问题,改善RNN的性能,近年来许多研究人员对此提出了一些改进措施并取得了不错的效果。

长短时记忆神经网络[47]是一种改进的 RNN,其中引入的门控机制能够有效地

缓解 RNN 存在的梯度消失和梯度爆炸问题,其基本结构如图2-5所示。结合LSTM 前向计算过程说明遗忘门(Forget Gate)、输入门(Input Gate)和输出门(Output Gate)三个门控单元的作用。

遗忘门决定当前时间步哪些输入将被丢弃。具体地,它将前一时间步的输出和当前时间步的输入拼接作为真正的输入,通过非线性激活函数将输出范围固定为,越接近1说明遗忘门越倾向于保留该输出,计算过程如下:

(2-4)

图2-5长短时记忆神经网络基本结构

Fig.2-5 Basic structure of LSTM

输入门产生输入的映射,计算过程如下:

(2-5)

随后会将输入更新到网络状态中,即状态更新,其计算过程如下:

(2-6)

(2-7)

输出门产生输出,并将其与前一时间步的单元状态交互得到输出,其计

算过程如下:

(2-8)

(2-9)

上式中,、、和分别表示遗忘门、输入门、候选状态和输出门;表示 Sigmoid 函数;和分别表示权重矩阵和偏置向量;表示按元素相乘。

2.3基于编—解码器结构的自动语音识别

编—解码器结构最先应用于机器翻译领域[48]。在该结构中,编码器首先将输入文本编码成向量序列,然后解码器根据该向量序列逐步输出文本。在每一个输出步骤中,解码器基于注意力机制为向量序列分配不同的权重,且下一个输出将由历史输出序列和该向量序列的加权和决定。编—解码器结构非常适合用于自动语音识别领域,这是由该结构中的注意力机制和自动语音识别任务本身特点共同决定的,具体如下:本质上自动语音识别任务和机器翻译任务都属于"序列—序

重庆邮电大学硕士学位论文

列"任务,即将输入序列识别为输出序列;编—解码器结构使用的注意力机制可以隐式学习输入序列与输出序列之间的软对齐关系,而不需要将数据提前对齐;编码器产生的向量序列不再是固定长度的单个向量,因此可以处理较长的语音输入序列。

基于以上原因,编—解码器结构在自动语音识别领域受到了广泛关注,其结构如图2-6所示。其中,编码器将输入特征序列映射为抽象的高级表征并传递给解码器,解码器根据接收信息预测输出序列。通常,预测特定所需的信息仅依赖于少数关键输入帧,因此在解码时不需要考虑每一个输入帧。注意力机制通过为每一对输入帧和分配相应的匹配分数来完成输入序列和输出序列的对齐,匹配分数的高低代表特定输入帧与的相关程度,解码器对每个输入帧的关注程度进行决定进而预测输出序列。

图2-6编—解码器结构

Fig.2-6 Structure of Encoder-Decoder

具体地,以加性注意力机制为例,首先使用解码器前一时间步的状态和

编码器得到的某一时刻状态计算注意力得分,计算过程如下:

(2-10)

式中,和分别为可训练的权值向量和偏置向量;和均为可训练的权重矩

阵。然后使用 Softmax 函数进行归一化,计算过程如下:

(2-11)

再加权求和得到输出向量:

(2-12)

最后解码器根据、前一时间步输出向量和状态向量得到当前时间步的状态向量和输出向量,经过 Softmax 函数之后得到预测的输出序列。

(2-13)

(2-14)

除了上述加性注意力机制外,目前使用较为广泛的注意力机制还有缩放点积注意力(Scaled Dot-Product Attention,SDPA)机制[49]、多头注意力(Multi-Head Attention,MHA)机制[17]和局部注意力(Local Attention,LA)机制[50]。通常,为了从不同角度获取各种层次的信息,编—解码器结构中会使用不同的注意力机制,所以目前出现了各种类型的编—解码器网络,如LAS模型[51]、Transformer模

型等。其中,Transformer 模型以极高的识别准确度和便捷的建模方式受到广泛关

注[52]。论文第三章将详细探讨 Transformer 模型,并在此基础上进行改进。

2.4基于显式联合建模的自然语言理解

在自然语言理解领域,考虑到意图检测和槽位填充两个子任务之间的密切相关性,主要采用联合建模的方式来利用两个子任务间的共享特征,主要可分为隐式联合建模和显式联合建模两种。前者仅采用共享编码器直接整合共享信息,而后者捕获特征后依旧通过交互模块进行交互,可以充分共享信息,从而提升模型性能,同时明确其交互过程有助于提高模型可解释性[22]。

以文献[53]提出的模型为例,其结构如图2-7所示,主要包含两个相互连接的双向 LSTM(Bi-LSTM),分别用于意图检测和槽位填充。每个 Bi-LSTM 双向读取输入序列并生成一系列隐藏状态,其中对应用于意图检测的网络,对应用于槽位填充的网络。

具体地,在意图检测的网络中,隐藏状态与来自另一个 Bi-LSTM()

产生的隐藏状态连接,计算时间步上的状态,最后输出预测的意图标

签,其计算过程如下:

(2-15)

重庆邮电大学硕士学位论文

(2-16)

式中,表示所有意图标签在最后一个时间步上的预测概率。

图2-7简单的显式联合建模

Fig.2-7 Simple explicit joint modeling

在槽位填充的网络中,计算过程与上式类似。区别之处在于,由于属于序列

标注领域,在每个时间步都有一个输出,计算过程如下:

(2-17)

(2-18)

式中,表示在时间步上的预测槽位标签。上述例子是一个经典的显式联合建模方法实例,论文第四章将在此方法基础上进行改进。

2.5本章小结

本章作为论文的相关基础理论章节,主要包含三部分。第一部分主要介绍了对话系统的基本组成结构,然后阐述了常用的几种深度神经网络;第二部分主要介绍了基于编—解码器结构的自动语音识别方法;第三部分主要介绍了基于显式

联合建模的自然语言理解方法。这一章节为后续第三章、第四章的内容做下铺垫。

Equation Chapter (Next) Section (Next)

第3章基于残差分组线性变换解码器的自动语音识别

3.1引言

Transformer 模型是近年较为流行的一种深度编—解码器模型,因其对语音输入—输出序列有强大的建模能力和优秀的识别能力,在自动语音识别任务中显现了良好的性能,受到了广泛的关注。随着研究的深入,逐渐衍生出了各式各样的变种模型。然而,Transformer 模型的性能往往与编码器块和解码器块的深度呈正相关。此外,模型维度越大,模型的性能、识别准确率和泛化能力也越高,但这些都会引发模型中权重矩阵维度的增加,进而导致模型参数量和计算复杂度的剧烈增长。例如,Pham 等人构建了一个包含48个编码器块和48个解码器块的Transformer 模型,其参数量达到了252M[54];Synnaeve 等人搭建了一个包含24个编码器块的 Transformer 模型,在模型维度为1024的情况下,其参数量达到了

270M[55]。为了降低基于 Transformer 模型的参数量,研究人员做出了各种尝试。

Luo 等人提出了一种简化的自注意力机制[56],使用前馈顺序记忆网络

(Feedforward Sequential Memory Network,FSMN)[57]代替映射层实现状态序列的转换,使得模型参数量下降了20%。但是,这种简单的替换并不能使得模型表现出良好的性能,而且很少有相关研究将这种轻量级网络应用于实际。

为解决上述问题,本章工作在原始 Transformer 模型的基础上,侧重于以识别准确率的小幅度降低为代价,实现模型参数量和计算复杂度的降低,具体地,通过将机器翻译领域提出的 DeLighT 模块[58]改进、嵌入至解码器中,使得模型在训练过程中可以适应向量维度和深度的变化,从而在增加模型宽度和深度的同时能减少参数量和计算量。

3.2 Transformer 模型及其组件

Transformer 模型如图3-1所示,可以看出是一个典型的编—解码器结构。左侧为编码器,由若干个块(Block)堆叠而成,每个块中包含两个子层,分别为多头注意力层和前馈网络层(Feedforward Network Layer,FNL);右侧为解码器,其结构与编码器类似。不同之处在于,解码器块将多头注意力层替换为掩膜(Masked)多头注意力层,并在之后级联了一个多头注意力层用于获取预测字符与语音特征间的相关性。每个子层之间都使用残差连接进行耦合,并在之后使用层归一化(Layer Normalization)对输出进行调整。下面对 Transformer 模型的几

重庆邮电大学硕士学位论文

个重要组件进行简介。值得注意的是,在较新的文献中,常用线性层指代全连接层[59]。因此,后续对这两种说法不加区分。

图3-1 Transformer 模型结构

Fig.3-1 Structure of Transformer

3.2.1自注意力机制

Transformer 模型中的编码器和解码器通过频繁地使用自注意力机制,从不同表示子空间中获取丰富的信息。自注意力机制使用点积运算来获取序列中任意位置元素间的相关性。

假设一个长度为、向量维度(模型维度)为的序列为自注意力在表示子空间上的输入。在中,自注意力首先使用由三个不同的权重矩阵

形成的全连接层将映射到向量维度为的三个序列,计算过程如下:

(3-1)

式中,,分别表示查询(Query)向量序列、键(Key)向量序列和值( Value )向量序列;,分别为对应的全连接网络权重矩阵。

然后将和的转置进行点积运算并使用进行缩放,便于产生的梯度能顺利地进行反向传播,接着使用 Softmax 函数进行概率化表示,进而得到二者之间的注意力权重矩阵,最后将该矩阵作用于即可得到的输出,计算

过程如下:

(3-2)

在此基础上,将至中各个输出序列按维度拼接,通过一个全连接层即可

得到多头注意力的输出,计算过程如下:

(3-3)

式中,表示拼接操作;,表示输出权重矩阵。

3.2.2前馈网络层

在 Transformer 模型中,模型维度通常指各模块输出向量的维度,取值常为256、512或1024(2的整数幂)。前馈网络层本质上是两层全连接网络,第一层全连接网络将扩张到更大的维度,第二个全连接网络再将维度还原。前馈网

络层的主要作用是增强模型的拟合能力,其前向传播的计算过程如下:

(3-4)

式中,表示激活函数;,分别表示两个全连接网络的权重矩阵;,分别表示两个全连接网络的偏置向量。

3.2.3位置编码

输入的语音序列和预测的字符序列都属于时间序列,具有特定的排列顺序,

且富含上下文语义信息,因此引入位置信息有助于增强表示序列内部关系的能力。

由于 Transformer 模型内部不存在与 RNN 类似的循环结构,无法利用其天然包含

重庆邮电大学硕士学位论文

的时序位置信息,注意力机制就会在不同上下文环境中输出相同的状态序列,进而导致位置信息完全丢失。可见在序列中附加各个向量的位置信息,对模型的训练学习是非常必要的。

因此,Transformer 模型中引入了位置编码(Positional Encoding,PE)来表示序列内部向量间的位置信息。将 PE 与输入序列进行加性组合,使得输入序列附带关于序列内部的位置信息,使得模型自身具备学习位置信息的能力。

具体地,假设序列长度为,表示第个时间步上维度为的位置编

码向量,其中每个元素的计算过程如下:

(3-5)

可以看出,每个元素正、余弦值的频率随位置的值的增加而减少。对于输入序列中的每个向量,线性加上对应的位置编码向量即可得到含有位置信息的模型向量。

3.3基于残差分组线性变换的"钻石"型缩放单元

本节首先介绍分组线性变换的特点,给出其计算过程,然后在此基础上加入残差连接,构建包含维度扩张和维度收缩两个阶段的"钻石"型缩放单元,同时给出其数学定义和拓扑描述。

3.3.1分组线性变换

与普通全连接层使用的线性变换不同,分组线性变换允许下层网络中的一部分神经元与上一层网络的部分神经元全连接,如图3-2所示。在不同的网络层,通过设定神经元被划分的组数,可以控制整个网络的参数量。具体地,对于一个层全连接网络,与分别表示第层的输入和输出,表示第层的神经元的划分组数,那么其参数量为,若采用分层分组线性变换,参数量为

。可见,在网络层数和每层神经元个数相同的情况下,引入分组线性变换后,参数量可下降至原始的。

分组线性变换第层中的每个神经元,都能通过多种途径到达输入向量中的某个输入神经元,以降低流动过程中的信息损失。具体地,假设输入向量和输出向量分别为和,分组线性变换首先根据维度和构建个维度逐渐增加的中间层。因此,第层的输出向量比第层拥有更大的维度。若需要进行变换的向量维度能够被设置的最大组数整除,某一层的输出向量的

计算过程如下:

(3-6)

式中,表示分组线性变换操作;表示第层的可训练的权重矩阵;表示第层的划分组数,值为。

图3-2分组线性变换

Fig.3-2 Group linear transformation

3.3.2"钻石"型缩放单元

在分组线性变换的基础上,加入残差连接,可以构建包含维度扩张和维度收缩两个阶段的"钻石"型缩放单元,如图3-3所示。缩放单元由5个配置参数决定:深度(层数)、宽度因子、输入维度、输出维度和分组线性变换的最大组数。

如图3-3(b)所示,缩放单元首先将输入向量扩张到更高维度(扩张阶段),

使网络具有比较强大的学习能力,再将维度收缩以匹配模型的维度(收缩阶段),

进而减少整个模型的参数量。具体地,在扩张阶段,维度为的输入向量在前

层被逐层映射到最高维度为的向量;在收缩阶段,最高维度向量在剩下的层变换为输出维度。

缩放单元的残差分组线性变换计算步骤如图3-3(a)所示,包含切分混合和分组线性变换两个主要操作,分组线性变换的输入为原始输入序列或中间层输出结果,表示切分混合函数。

重庆邮电大学硕士学位论文

图3-3"钻石"型缩放单元的结构。(a)残差分组线性变换;(b)拓扑结

构;(c)逐块缩放策略

Fig.3-3 Structure of the "diamond" scaling unit.(a) Residual grouping linear

transformation;(b) Topology;(c) Block-wise scaling strategy

以第2层为例,首先将上一层的输出和原始输入根据该层组数按照相同规律进行切分,得到、、和4个中间向量,前两者来自于,后两者来自于;然后按特定顺序合并和、和,得到的两个向量即为的输

出;最后将结果输入得到最终输出。的计算过程如下:

(3-7)

(3-8)

上式中,和分别表示第层的权重矩阵和偏置向量。

对于某些层数较深的网络,可以在其组件中嵌入含有逐块缩放策略的"钻石"型缩放单元,如图3-3(c)所示。每一个六边形表示一个缩放单元,对于处于位置

上的缩放单元,其配置参数和的计算过程如下:

(3-9)

(3-10)

上式中,和均为超参数,分别表示最小深度和最大深度;表示缩放单元总个数。

3.4基于"钻石"型缩放单元的改进 Transformer 模型

本节在原始 Transformer 模型基础上,首先将普通前馈网络层改进为轻量级前馈网络层,然后将上节所述"钻石"型缩放单元嵌入到解码器中,形成轻量级解码器,最终得到改进的、轻量级 Transformer 模型。

3.4.1轻量级前馈网络层

轻量级前馈网络层的结构如图3-4右侧所示,类似于普通前馈网络层,也由两层全连接网络组成。区别之处在于不需要对输入序列的维度进行扩张。

图3-4普通前馈网络层(左侧)和轻量级前馈网络层(右侧)的结构 Fig.3-4 Structure of the normal FNL (left) and the lightweight FNL (right)

假设收缩因子为,轻量级前馈网络层的全连接层1将维度为(和模型维度一致)的序列转变成维度为的序列,全连接层2再将序列维度还原至。假定普通前馈网络层需要先将序列维度扩张倍,那么同等条件下,轻量级前馈网络层的参数量是普通前馈网络层的。另外,轻量级前馈网络层的激活函数为最近提出的 MISH 激活函数[60]。相比于 ReLU 激活函数,MISH 求得的梯度更加平滑,更有利于模型的训练。

3.4.2轻量级解码器

通常,提升 Transformer 模型的表达能力主要有以下三种方法[61]:

(1)宽度缩放:增加模型中向量维度和各个映射维度;

(2)深度缩放:增加模型深度(层数);

(3)综合考虑宽度缩放和深度缩放。

然而,上述方法在某些体量较小的数据集上的表现差强人意。例如,针对THCHS-30数据集[62],将 Transformer 模型的模型维度从512增加至1024时,其

重庆邮电大学硕士学位论文

参数量扩大了4倍,但性能变化不大。而在 Transformer 模型各模块中均匀地增加模型维度和深度,同样会导致模型参数量的急剧增加。

为了减少 Transformer 模型的参数量,基于"钻石"型缩放单元的逐块缩放策略,在不同的解码器块中嵌入深度、宽度不均匀的缩放单元,并将普通的前馈网络层替换为改进的轻量级前馈网络层,形成轻量级解码器块,可以使得模型在宽度和深度增加的同时进一步减少参数量,其结构如图3-5所示。

图3-5轻量级解码器结构

Fig.3-5 Structure of the lightweight Decoder

在每一个块中,首先通过一个缩放单元完成输入序列的深度缩放和宽度缩放,配置参数中的深度和宽度因子将逐块增加,越靠近输出端,缩放单元的结构会变得越深、越宽;解码器中的自注意力机制和普通自注意力机制有所不同,普通自注意力机制使用的三个权重矩阵通常将输入序列维度减小,而在轻量级解码器块中,由于缩放单元的输出向量维度较低,因此掩膜自注意力的权重矩阵不

必再次减小输入序列的维度,只需将计算得到的序列通过线性层映射至模型维度;

之后的编—解码器自注意力也是如此;最后使用轻量级前馈网络层代替普通前馈网络层进一步降低参数量。

3.4.3改进的 Transformer 模型结构

在原始 Transformer 模型的基础上,采用上述内容对解码器进行优化,得到如图3-6所示的改进 Transformer 模型。其中,解码器由若干个轻量级解码器块堆叠而成,用于实现整个模型的轻量化。

图3-6轻量级 Transformer 模型结构

Fig.3-6 Structure of the lightweight Transformer

3.5实验结果及分析

本节首先介绍实验环境、使用的数据集和评价指标,然后阐述语音信号的频域特征提取方法,并给出详细的语音信号及标签文本预处理流程,接着描述网络模型的训练配置信息及过程,最后通过对比实验结果分析改进模型的性能,同时设计一系列消融实验论证所提方法的有效性。

3.5.1实验环境、数据集和评价指标

1)实验环境

实验环境包含硬件和软件两部分。在硬件方面,主要使用一台高性能服务器完成所有模型的训练,其主要配置为:处理器(Central Processing Unit,CPU)/2×Intel(R) Xeon(R) CPU E5-2680 v4@2.40GHz;内存(Memory)/8×ECC Registered DDR4@32GB;显卡(GPU)/4×NVIDIA Tesla P100@16GB;在软件方面,所有模型的训练都在 Ubuntu 18.04系统下完成,编程语言为 Python 3.6.9,深度学习框架为 PyTorch 1.3。详细信息见表3-1。

重庆邮电大学硕士学位论文

表3-1实验环境信息

Table3-1 Experimental environment

类型型号规格

硬件

CPU Intel(R) Xeon(R) CPU E5-2680 v4@2.40GHz 2

GPU NVIDIA Tesla P10016GB×4

内存 ECC Registered DDR4@32GB 32GB×4硬盘西部数据 SSD 阵列21TB 麦克风 Actins ATS3605D 1

软件

操作系统 Ubuntu 18.04

编程语言 Python 3.6.9

深度学习框架 PyTorch 1.6

2)数据集

实验使用在自动语音识别领域内应用较为广泛的 AISHELL-1数据集[63](开源,中文)和 TED-LIUM2数据集[64](开源,英文)。其中,AISHELL-1包含近

170小时的语音数据,语音段数约14万,来自于400个来自中国不同口音区域的发言人,以44.1KHz(高保真麦克风)和16kHz(Android 或 iOS 系统手机)的采

样率进行录制,录制过程在安静室内环境中进行,内容涉及智能家居、无人驾驶、工业生产等11个领域;TED-LIUM2包含约207小时的语音数据,语音段数约9万,来自于1242个发言人的1495段演讲的音频和文字稿,演讲视频全部来自于TED 网站。

AISHELL-1和 TED-LIUM2都按照一定的比例被划分为训练集(Train)、验证集(Val)和测试集(Test)三个子集,前者的划分比例参照文献[63],后者的划分比例参照文献[65],具体信息见表3-2。另外,AISHELL-1中发言人的年龄、性别和方言信息见表3-3。

表3-2 AISHELL-1和 TED-LIUM2的详细信息 Table3-2 Detail of AISHELL-1 and TED-LIUM2

数据集子集语音段数时长(小时)

AISHELL-1

Train 120098150

Val 1432610

Test 72005

TED-LIUM2

Train 91262203

Val 5801.5

Test 11342.5

表3-3 AISHELL-1中发言人的年龄、性别和方言信息

Table3-3 The age, gender, and accent of speakers in AISHELL-1

年龄、性别信息方言信息

年龄段人数

比例

(%)

男性人数女性人数区域人数

比例

(%)

16—25岁31679140176北方33383

26—40岁71183635南方3810

40岁以上133103闽、贵、粤及其他297

合计400100186214400100

3)评价指标

在模型准确度方面,论文选用常用的标签错误率作为评价指标[66]。对于采用

不同建模单位的数据集,评价指标的计算方式类似。对于 AISHELL-1,因其以汉

字单字符作为基本建模单位,故对应的评价指标为字错误率(Character Error Rate,

CER);对于 TED-LUM2,因其以一元模型(Unigram)作为基本建模单位,故

对应的评价指标为词错误率(Word Error Rate,WER)。二者定义如下:

(3-11)

式中,分别表示相较于参考文本替换/增加/删除的字符(单词)数;表示参考文本的字符(单词)数目。

在模型轻量化程度方面,论文选用参数量(Parameters)和计算量作为评价指标。对于前者,深度学习框架 PyTorch 中集成了成熟的计算方式,只需加载模型即可得到具体结果;对于后者,使用常用的乘加累积操作数(Multiplication and

Accumulations,MACs)[67]进行表示。对于同一个网络,在其他条件相同的情况

下,其 MACs 值约是另一个常见的反映网络复杂度的指标——浮点运算数(Floating Point Operations,FLOPs)的。对于一层将输入维度映射为输

出维度的全连接网络,其 MACs 值的计算过程如下:

(3-12)

对于较为复杂的卷积操作,MACs 值的计算过程如下:

(3-13)

式中,分别表示卷积核的高度和宽度;分别表示输入和输出的通道数;分别表示输出图像的高度和宽度。

重庆邮电大学硕士学位论文

在模型实际应用(推理)方面,论文使用 GPU 内存占用率(GPU Memory-Usage Rate,GMUR)和实时因子(Real Time Factor,RTF)[68]分别反映模型的资源占用和识别速度。对于前者,可使用 Nvidia 官方提供的显卡驱动自带工具

(nvidia-smi,NVSMI)监控 GPU 状态得到,为了保证不受某些特殊因素的影响,

对于每个模型的 GMUR 值,论文共设计5轮测试并计算其均值作为最终的结果;

对于后者,其定义如下:

(3-14)

式中,表示模型识别所有音频的处理时间;表示所有音频的总时间。

RTF 值越小表示模型的识别速度越快,小于1才能达到实时效果,通常在

0.2~0.3。同样地,对于每个模型的 RTF 值,论文共设计5轮测试并计算其均值作为最终的结果。

3.5.2数据预处理

1)频域特征提取

论文使用应用较为广泛的 FBank 特征作为语音信号的频域特征[66]。具体地,

首先对分帧、加窗后的语音帧做快速傅里叶变换(Fast Fourier Transform,FFT)

得到频谱,对频谱取模得到振幅;然后使用公式

将实际频率转化为梅尔(Mel)频域,以便模拟人耳对声音的感知;最后使用三角滤波器组进行滤波,再取对数即可得到 FBank 特征。

对于 AISHELL-1,论文使用窗口长度为25ms、帧移为10ms 的80维 FBank特征作为模型输入;对于 TED-LIUM2,在80维 FBank 特征的基础上额外增加了

3维 Pitch 特征[69]。在上述原始特征基础上,论文还使用卷积模块对其进一步处理。

其中,卷积核为3×3、步长为2、输出通道数为256、激活函数为 ReLU。在卷积模块的作用下,原始特征的时间步长度被压缩至。

2)标签文本处理

对于 AISHELL-1,论文使用不同的4336个字符来构成词汇表(字典)。其中4333个字符来自于本身的标注文本,剩下3个为增加的用于辅助模型训练的特殊字符,分别为<EOS/BOS>、<PAD>和<UNK>。<EOS/BOS>可指示模型开始或停止输出字符序列;<PAD>可在训练阶段将一个批处理(Batch)大小的数据中不同长度的语音特征序列或字符序列填充到相同的长度;<UNK>可在输出时代替某些在词汇表中未出现的字符,避免模型训练时出错;对于 TED-LUM2数据集,

论文使用 SentencePiece 工具[70]生成建模单位为一元模型的词汇表,共包含500个字符。

3.5.3训练配置信息及过程

论文选择领域内近年来一些性能表现优秀的同类模型作为对比组,具体为:TDNN-Chain[71]、LAS[72]、SSAN[56]、Speech-Transformer(非官方开源)[73]、HA-

Transformer(非官方开源)[74]和 STBD(非官方开源)[75]。由于提出这些方法的

文献中未涉及评价指标中的 MACs、GMUR 和 RTF 值,且只在单一语种的数据集上进行训练,所以论文根据这些开源方法,逐一搭建了网络模型并在两个目标数据集上完成了训练。表3-4展示了各个模型主要超参数的配置信息,包含编码器激活函数(E-AF)、解码器激活函数(D-AF)、优化器(OP)、热身步数(WU)、学习率(LR)、Dropout 概率(DR)、批处理大小(BS)和迭代次数(EP)。

改进的 Transformer 模型由12个编码器块和6个解码器块堆叠而成,表示子空间的个数均为4,模型维度为256,其中缩放单元的参数、、、、和 DR 分别为4、12、6、2、8和0.03。

论文使用Kullback-Leibler散度作为模型训练的损失函数,标签平滑度(Label

Smoothing)设为1,其定义如下:

(3-15)

式中,和分别表示第个字符的真实分布和预测分布。

表3-4各个模型主要超参数的配置信息

Table3-4 Configuration of the main hyper-parameters of each model

模型数据集 E-AF D-AF OP WU LR DR BS EP

Speech-

Transformer[73]

AISHELL-1 ReLU ReLU

Adam

2.5e41.00.10128100

TED-LIUM2 GLU GLU 1.2e40.8e-30.15 DBS 150

HA-

Transformer[74]

AISHELL-1

GLU GLU Adam

2.5e41.00.10128100

TED-LIUM21.2e40.2e-20.20 DBS 150

STBD[75] AISHELL-1 ReLU ReLU Adam 1.6e41.00.1096100

Proposed

AISHELL-1

Swish ReLU Adam

2.5e41.00.10128100

TED-LIUM21.2e40.8e-30.10 DBS 150注:STBD 模型在 TED-LIUM2等英文数据集上无法收敛。

重庆邮电大学硕士学位论文

与在 AISHELL-1上不同,在 TED-LIUM2上的批处理大小设置为动态的(Dynamic Batch Size,DBS)。在 AISHELL-1/TED-LIUM2上,模型在迭代训练

100/150次后,将最后30/50次迭代的参数值进行平均形成最终训练好的模型[74]。论文使用了近期提出的 SpecAugment[76]对训练集数据进行必要的增强。另外,为

了保证实验结果对比的公平性,在模型构建过程中没有使用任何例如以 C++语言进行编程的加速技巧。训练过程中各个模型在验证集上的损失(Loss)曲线如图

3-7所示(从第10次迭代开始)。

参照大多数自动语音识别系统,论文构建了外部语言模型(Language Model,

LM)。在推理阶段,将原始模型和语言模型的预测结果以浅融合[77]的方式综合,

可以进一步提高识别精度。论文针对 AISHELL-1和 TED-LIUM2构建了两个相同

的具有1024个隐藏单元的两层 LSTM 网络模型作为 LM,BS 分别为512和128,训练语料来自于各自训练集中的标签文本。使用单个GPU进行训练,LMs在迭代训练60次后,将最后10次迭代的参数值进行平均形成最终训练好的模型。

图3-7训练过程中 AISHELL-1(左侧)和 TED-LIUM2(右侧)验证集

上各个模型的损失曲线

Fig.3-7 Loss curves for each model on the validation subset of AISHELL-1

(left) and TED-LIUM2(right) during training

3.5.4对比实验结果及分析

将改进的 Transformer 模型和其他基于 Transformer 的同类模型在 AISHELL-1和 TED-LIUM2上的实验结果进行对比,具体信息分别见表3-5和表3-6。表中标"\*"表示未开源的方法,MACs 值是在编码序列长度为500、解码序列长度为30的条件下计算得出,"↓"表示该项指标越低越好,粗体数字表示最优结果,下划线数字表示次优结果。

从表3-5中可以看出,在 AISHELL-1上,与对比组中识别效果最好的 HA-Transformer相比,改进模型在测试集上的CER上升了0.71%,RTF值上升了0.015(在实际应用中影响较小),但参数量和计算量分别相对下降了48.35%和77.16%,

同时 GMUR 也下降了24.94%,引入语言模型后 CER 可进一步降低至6.38%;与对比组中参数量和计算量都最小的 Speech-Transformer 相比,引入语言模型后,验证集和测试集上的 CER 分别下降了0.97%和1.09%,参数量和计算量分别相对下降了27.66%和70.72%,同时GMUR也下降了25.06%。改进模型在TED-LIUM2上的结果与在 AISHELL-1上类似。

表3-5 AISHELL-1上各个模型的实验结果

Table3-5 Experimental results of each model on AISHELL-1

模型

CER (%)↓ Parameters (M)

↓

MACs (B)

↓

GMUR (%)

↓

RTF

↓ Val Test

TDNN-Chain (Kaldi)\*[71]7.45

LAS\*[72]10.56

SSAN\*[56]6.8436.0

Speech-Transformer[73]6.917.4728.218.175.470.059

HA-Transformer[74]5.585.9639.523.275.350.067

STBD[75]7.438.0353.859.364.340.083

Proposed 6.186.6720.45.350.410.082

Proposed (With LM)5.946.38

表3-6 TED-LIUM2上各个模型的实验结果

Table3-6 Experimental results of each model on TED-LIUM2

模型

WER (%)↓ Parameters (M)

↓

MACs (B)

↓

GMUR (%)

↓

RTF

↓ Val Test

Speech-Transformer\*[73]10.9511.4527.316.740.460.090

HA-Transformer\*[74]10.8111.4139.421.542.390.091

Proposed 11.8011.8620.34.634.070.115

Proposed (With LM)10.7611.11

为了更加直观地看出所提方法的效果,论文将上述结果进行可视化处理。具体地,选择在测试集上的 CER/WER、参数量 Parameters、计算量 MACs 和 GPU内存占用率 GMUR 等指标,将 CER/WER 作为纵轴、其他三项分别作为横轴绘制

散点图,如图3-8所示。可以看出,改进模型的对应点在三幅图上都位于最左侧,除了在 AISHELL-1上的 CER 略高于 HA-Transformer,在参数量、计算量和 GPU

重庆邮电大学硕士学位论文

内存占用率等方面都达到了最优,进一步证明了所提方法的有效性,即在保证一定识别准确率的情况下,实现模型参数量和计算复杂度的降低。

图3-8各个模型实验结果的可视化处理。(a) CER/WER 与 Parameters 的关系;(b) CER/WER 与 MACs 的关系;(c) CER/WER 与 GMUR 的关系

Fig.3-8 Visualization of experimental results for each model.(a) Relation between CER/WER and parameters;(b) Relation between CER/WER and

MACs;(c) Relation between CER/WER and GMUR

3.5.5消融实验结果及分析

论文围绕解码器类型和注意力表示子空间个数设计消融实验,探究它们对于整个改进模型性能提升的贡献程度。具体地,在本章提出的改进模型的基础上,将轻量级解码器替换为普通解码器、将轻量级解码器的注意力表示子空间个数从

4降低为1,从而得到两个变种模型,分别用ND(Normal Decoder)和SH(Single

Head)表示,观察它们在 AISHELL-1和 TED-LIUM2上的评价指标变化情况,具体信息见表3-7。表中,Base 指本章提出的改进模型,"↓"表示该项指标越低越好,括号内数字表示该项指标值与 Base 模型的差异,红色代表结果变好,绿色代表结果变差。除了变化的模块,所有变种模型的其他部分(包括网络结构、实验环境和训练配置等)均与 Base 模型一致,同时移除所有的外部语言模型。

表3-7消融实验结果

Table3-7 Ablation study results

数据集

模

型

CER/WER (%)↓ Parameters

(M)↓

MACs (B)

↓

GMUR (%)

↓

RTF

↓ Val Test

AISHELL-1

Base 6.186.6720.45.350.410.082

ND

6.00

(-0.18)

6.47

(-0.20)

25.5

(+5.1)

15.6

(+10.3)

63.24

(+12.83)

0.064

(-0.018)

SH

6.36

(+0.18)

6.79

(+0.12)

20.4

(0)

5.3

(0)

52.37

(+1.96)

0.073

(-0.009)

TED-LIUM2

Base 11.8011.8620.34.634.070.115

ND

11.05

(-0.75)

11.45

(-0.41)

24.9

(+4.6)

14.2

(+9.6)

42.73

(+8.66)

0.087

(-0.028)

SH

11.41

(-0.39)

12.07

(+0.21)

20.3

(0)

4.6

(0)

35.43

(+1.36)

0.109

(-0.006)

从上表中可以看出,所有变种模型的RTF值相较于Base模型都有略微降低,均在0.03以内,在实际应用中影响较小。在 AISHELL-1上,若使用普通解码器(ND),验证集和测试集上的 CER 分别下降了0.18%和0.20%,但参数量和计算量分别相对上升了25.00%和194.34%,同时 GPU 内存占用率也上升了12.83%;若使用含有单个注意力表示子空间的轻量级解码器(SH),验证集和测试集上的CER 绝对值分别上升了0.18%和0.12%,参数量和计算量没有变化,GPU 内存占用率上升了1.96%。在 TED-LIUM2上的结果与在 AISHELL-1上类似。

为了更加直观地看出各变种模型和 Base 模型的区别,论文将上述结果进行可视化处理。具体地,选择在测试集上的 CER/WER、参数量 Parameters、计算量MACs 和 GPU 内存占用率 GMUR 等指标,将 CER/WER 作为纵轴、其他三项分别作为横轴绘制散点图,如图3-9所示。可以看出,Base 模型的对应点在三幅图上都位于最左侧,表明引入轻量级解码器,可以实现模型参数量、计算复杂度和GPU 内存占用率的显著降低,而只牺牲小部分的识别性能,达到识别性能和模型轻量化的平衡状态,进一步论证了所提方法的有效性。此外,轻量级解码器中注意力表示子空间的个数在很大程度上会影响到模型的识别准确率,采用单一表示子空间时模型的识别性能较差。

重庆邮电大学硕士学位论文

图3-9各个变种模型实验结果的可视化。(a) CER/WER 与 Parameters 的关系;(b) CER/WER 与 MACs 的关系;(c) CER/WER 与 GMUR 的关系

Fig.3-9 Visualization of experimental results for each variant model.(a) Relation between CER/WER and parameters;(b) Relation between CER/WER

and MACs;(c) Relation between CER/WER and GMUR

3.6本章小结

本章工作在原始 Transformer 模型的基础上,侧重于以识别准确率的小幅度降低为代价,实现模型参数量和计算复杂度的降低。首先提出基于残差分组线性变换的"钻石"型缩放单元,然后将缩放单元嵌入至原始 Transformer 模型的解码器中形成轻量级解码器,最终得到改进的、轻量级 Transformer 模型。在 AISHELL-

1和 TED-LIUM2数据集上的实验结果论证了本章所提方法的有效性。

Equation Chapter (Next) Section (Next)

第4章基于标签感知图交互的自然语言理解

4.1引言

基于显式联合建模的自然语言理解是近年来较为流行的一种建模方式,因其交互模块能充分共享捕获到的两个子任务间的共享特征信息,且能明确两个子任务间的交互过程以提高模型可解释性,在自然语言理解任务中展现了良好的性能,

受到了广泛的关注。然而,当前的研究方法需要进一步讨论以下问题:

(1)过于简化的标签特征。捕获特征只关注于原始话语和特征(One-hot)编码之间的相关性,而忽略了直观的原始标签特征。而标签语义可以通过评估话语中的字符和标签中的字符之间的语义相似性来提高自然语言理解模型的性能;

(2)维度较低的交互模块。交互模块只考虑了在字符级别(Token-level)上的意图—槽位交互,缺少了在语句级别(Sentence-level)上的全局优化,导致上下文特征泄露,从而影响模型性能。

为解决上述问题,本章工作在经典显式联合建模方法的基础上,侧重于对特征捕获和交互方式进行优化,实现模型交互能力和预测精度的提高。具体地,首先提出标签映射模块获取原始话语和标签语义之间的相关性以提供丰富的先验知识,然后提出全局图交互模块对语句级别的意图—槽位交互过程进行建模以提供全局优化,从而提升模型性能。

4.2基于最佳线性逼近的标签映射模块

受文献[78]的启发,论文利用最佳线性逼近(Best Linear Approximation)[79]

思想辅助检测原始话语的意图。具体地,基于最佳线性逼近构建标签映射模块,将意图标签特征自适应地融合到话语特征中,以增强模型的表征能力。

4.2.1最佳线性逼近

线性逼近是一种数学方法,它可以将一个函数在某一点处近似地表示为一个线性函数。具体来说,对于一个可导函数,可以在其某一点处,通过求取其在该点的导数来构造一个线性函数,使得在该点附近可以近似地代替。最佳线性逼近是在线性逼近的基础上,使得可以最优地代替,即在所有一次函数中,与的误差最小。

在多维空间上,最佳线性逼近指的是:通过一个线性函数来最好地拟合给定的数据点集合。具体来说,假设有一组数据点,它们位于多维空间中的某个子空

重庆邮电大学硕士学位论文

间上。需要想要找到一个线性函数,能够最好地拟合这些数据点。这个线性函数可以用一个向量表示,即最佳线性逼近向量。

上述过程用数学符号可描述为:假设表示 Hilbert 空间、表示的一个子空间,对于一个给定向量,需要找到一个离最近的向量,

的解是维空间的一组基向量的线性组合,其

中系数满足,为 Gram 矩阵,。

4.2.2标签映射模块

将上述思想迁移应用至自然语言理解领域,利用意图标签数据构建标签映射模块,其结构如图4-1所示。下面对模块的构建细节进行介绍。

图4-1标签映射模块结构

Fig.4-1 Structure of label injection module

1)标签编码器

构造一个以为基向量的标签空间,其中表示意图标签的个数。为了得到这一组基向量,设计标签编码器,通过堆叠的 Bi-LSTM 和自注意力机制对原始标签数据进行编码。具体地,首先使用 Bi-LSTM 双向读取输入序列

以生成隐藏状态,其中

;然后使用与式(3-2)相同的计算步骤得到自注意力的输出;接着将与连接形成编码向量;最后使用

Softmax 函数得到最终的输出,计算过程如下:

(4-1)

(4-2)

式中,为可训练的权重矩阵;表示偏置向量。

2)标签映射

通过上述编码方式构造标签空间后,对于一个给定的原始话语向量,可

将其映射到上以获取最佳线性逼近向量,计算过程如下:

(4-3)

式中,系数,Gram 矩阵和定义如下:

(4-4)

需要注意的是,每个向量代表一个意图,且是线性独立的,因此 Gram 矩阵是正定的并且有逆。

4.3基于图注意力网络的全局图交互模块

为了高效融合所有的交互信息,深入挖掘两个子任务间的语义特征,论文提出全局图交互模块提供双向交互通道。具体地,基于图注意力网络搭建全局图交互模块,对语句级别的意图—槽位交互过程进行建模以提供全局优化,从而提升模型的交互能力。

4.3.1图注意力网络

图注意力网络(Graph Attention Network,GAT)[80]是一种图神经网络的变体,

融合了图的结构信息和节点特征,其掩膜自注意力层让节点专注于邻域特征并学习不同的注意力权重。与传统的图卷积神经网络(Graph Convolutional Network,

GCN)不同,GAT 不仅考虑节点之间的邻接关系,而且还考虑节点之间的关联性,

并根据关联性对邻节点进行不同程度的加权。具体地,GAT 接收节点特征

作为输入,其中,表示节点个数,表示每个节点的特征个数;输出一组新的节点特征,其中(可能具有不同的基数)。

重庆邮电大学硕士学位论文

为了将输入特征转化为更高层次的特征,首先将由权重矩阵参数化的共享线性变换应用于每个节点;然后在节点上使用自注意力机制计算系数

,计算过程如下:

(4-5)

式中,表示节点的特征与节点的关联性;由权重矩阵得到,并使用 LeakyReLU 激活函数。

接着使用 Softmax 函数对进行归一化处理得到,计算过程如下:

(4-6)

式中,表示节点的所有一阶邻节点(包括本身)的集合。

将上两式综合,自注意力机制计算出的系数可以表示为:

(4-7)

式中,表示转置操作。上述过程如图4-2所示。

图4-2系数的计算过程

Fig.4-2 Calculation process of coefficient

最后使用非线性函数作用于,并引入多头注意力机制得到最终的输出特

征,计算过程如下:

(4-8)

式中,表示多头注意力机制中表示子空间的个数;表示由第个表示子空间计算得到的归一化系数;表示相对应的权重矩阵。上述过程如所示(以为例),其中不同颜色、形式的箭头表示各自独立的注意力计算方式。

图4-3多头注意力机制下的特征融合过程

Fig.4-3 Feature aggregation based on multi-head attention mechanism

4.3.2全局图交互模块

基于上述思想,为了实现语句级别的意图—槽位交互,搭建全局图交互模块,

其中所有的意图预测结果和槽位序列都相互连接,如图4-4所示,橙色方框中的

表示与槽位相关的隐藏状态序列,绿色圆框中的

表示意图预测结果。在数学意义上,图(Graph)可以表示为,其中顶点指的是意图和槽位,边指的是它们之间的相关性。下面将从这些方面对模块的构建细节进行介绍。

图4-4全局图交互模块结构

Fig.4-4 Structure of global graph interaction module

重庆邮电大学硕士学位论文

1)顶点

图中共有个顶点,其中表示原始话语的长度,表示的个数,即

预测出的意图个数。输入的槽位字符特征定义如下:

(4-9)

(4-10)

式中,表示层数;表示可训练的权重矩阵。初始状态。

输入的意图特征定义如下:

(4-11)

式中,表示用于映射的可训练的嵌入参数。

将上述两种特征(节点)组合,形成图的第1层状态向量。

2)边

图中有3种连接关系,分别为:槽位—槽位连接:构造槽位—槽位连接关系,

其中每个槽位节点通过滑动窗口(超参数)连接其他槽位节点,以进一步获取槽位之间的依赖性并合并上下文信息;意图—意图连接:考虑到所有意图节点都出自于同一个原始话语,构造意图—意图连接关系,获取所有意图节点之间的关联性;意图—槽位连接:由于意图和槽位之间的密切联系,构造意图—槽位连接关系,对两个子任务之间的全局交互过程进行建模。具体地,引入缩放点积注意力

机制计算意图和槽位之间的相关程度,计算过程如下:

(4-12)

式中,表示第个状态向量(槽位)和第个意图预测结果的相关程度;表示隐藏单元的维度。

若(超参数),表明该槽位节点和意图节点有足够的关联性。在这种情况下,将该槽位节点与预测意图节点进行连接。

3)特征融合

图中在第层的特征融合过程如下:

(4-13)

式中,和分别表示槽位节点和意图节点的集合;表示可训练的权重矩阵。

4.4基于标签感知的图交互模型

在经典显式联合建模方法(如图2-7所示)的基础上,采用上述模块对交互过程进行优化,得到如图4-5所示的基于标签感知的图交互模型。下面对该模型的一些重要组件进行简介。

图4-5基于标签感知的图交互模型

Fig.4-5 Structure of graph interaction model based on label-aware

4.4.1共享编码器

对于给定的长度为的输入序列,使用与标签编码器(节

4.2.2)相同的编码方式得到和,其中共享向量作为意图检测子网络和槽位填充子网络的输入。此外,在槽位填充子网络中使用 Bi-LSTM 对共享向量进一步编码,得到与槽位相关的隐藏状态序列,其中第个时间步的隐藏状态。

4.4.2意图解码器

受文献[81]的启发,论文引入阈值(超参数)辅助意图的解码预测,的值可以根据不同的数据集进行调整。具体地,在经过标签映射模块(节4.2.2)

重庆邮电大学硕士学位论文

得到最佳线性逼近向量之后,首先将其输入至意图解码器,获得关于所有意图

的概率分布序列,计算过程如下:

(4-14)

式中,表示 Sigmoid 函数;和表示可训练的权重矩阵和偏置向量。

然后逐一比较和,若,则将其作为最终的意图之一。例如,当且时,。

4.4.3槽位解码器

在经过全局图交互模块(节4.3.2)中层的特征融合后,使用 argmax 和

Softmax 函数得到槽位预测结果,计算过程如下:

(4-15)

式中,表示第个字符的槽位;表示可训练的权重矩阵。

4.5实验结果及分析

本节首先介绍实验环境、使用的数据集和评价指标,然后给出对于标签文本的预处理流程及文本特征提取方法,接着描述网络模型的训练配置信息及过程,最后通过对比实验结果分析改进模型的性能,同时设计一系列消融实验论证所提方法的有效性。

4.5.1实验环境、数据集和评价指标

1)实验环境

实验环境与节3.5.1相同,具体信息可见表3-1。

2)数据集

实验使用在自然语言理解领域内应用较为广泛的开源多意图(Multiple Intent)

数据集:MixATIS[81]和 MixSnips[81],二者分别由单意图(Single Intent)数据集ATIS(Airline Travel Information Systems)[82]和 Snips[83]扩充而来,使得每句话语

包含1~3个意图。其中,MixATIS 包含14746句话语,涵盖关于航班的编号

(atis\_flight\_no)、时间(atis\_flight\_time)和餐食(atis\_meal)等17个意图类别,

由录音和相应的人工记录组成,内容涉及用户在航空公司旅行查询系统上的航班询问信息;MixSnips 包含44173句话语,涵盖查询天气(GetWeather)、预定餐厅(BookRestaruant)和播放歌曲(PlayMusic)等7个意图类别。MixATIS 和MixSnips 的子集划分见表4-1,二者均已完成数据清洗,去除了重复的语句。

表4-1 MixATIS 和 MixSnips 的划分信息 Table4-1 Split of MixATIS and MixSnips

数据集训练集(句)验证集(句)测试集(句)

MixATIS 13162756828

MixSnips 3977621982199

3)评价指标

在模型准确度方面,论文选用常用的意图准确率(Intent Accuracy,IAcc)、槽位 F1值和整体准确率(Overall Acc,OAcc)分别作为意图检测、槽位填充和

整体语句的评价指标[52]。其中,OAcc 表示意图和槽位均预测正确的句子所占的

比例,IAcc 和 F1值的定义如下:

(4-16)

(4-17)

式中,TP(True Positive)表示预测正样本正确的个数;TN(True Negative)表示预测负样本正确的个数;FP(False Positive)表示预测正样本错误的个数;FN(False Negative)表示预测负样本错误的个数。

在模型实际应用(推理)方面,论文使用 GPU 内存占用率(GPU Memory-Usage Rate,GMUR)和推理时间(Latency)分别反映模型的资源占用和预测速度。对于前者,与节3.5.1中计算方式一致;对于后者,定义为模型推理一个批次数据的时间,可使用 Python 中自带的工具库 Time 计算得到,为了保证不受某些特殊因素的影响,对于每个模型的 Latency 值,论文共设计5轮测试并计算其均值和标准差作为最终的结果。

4.5.2数据预处理

1)标签文本处理

和节3.5.2类似,在自然语言理解任务中,同样需要建立词汇表(字典)。对于 MixATIS 和 MixSnips,论文分别使用不同的554/1435个字符来构成词汇表,其中551/1432个字符来自于本身的标注文本,另外包含3个用于辅助模型训练的特殊字符,即<EOS/BOS>、<PAD>和<UNK>。

2)文本特征提取

论文使用应用较为广泛的 Word2Vec 中的 Skip-gram 模型提取文本特征[84]。

Skip-gram 是一种用于获取文本特征的模型,它的输入是一个中心字符,而输出是

重庆邮电大学硕士学位论文

与该中心字符相关的字符。具体地,给定一个长度为的文本序列

,其中表示文本中的第个字符。对于任意一个中心字符,Skip-gram 模型的目标是预测它周围的上下文字符。具体地,Skip-gram 模型的损

失函数基于交叉熵损失,定义如下:

(4-18)

式中,表示上下文窗口的大小;表示在给定中心字符的条件下生成上下文字符的概率。

这个概率可以通过对它们之间的向量(One-hot编码)进行点积操作,然后将

结果送入 Softmax 函数得到,计算过程如下:

(4-19)

式中,表示上下文字符对应的向量;表示中心字符对应的向量;表示词汇

表的字符数;

在训练结束后,对于词汇表中的任一字符,可得到以该字符的中心字符向量和上下文字符向量,使用前者作为提取的文本特征。

4.5.3训练过程及配置信息

论文选择领域内近年来一些性能表现优秀的同类模型作为对比组,具体为:

Attention BiRNN[85],Slot-Gated(官方开源)[29]、Bi-Model[53]、SF-ID(官方开源)

[32]、Stack-Propagation(官方开源)[31]、Joint Multiple ID-SF[86]、AGIF(官方开源)

[81]和 SDJN[87]。由于提出这些方法的文献中未涉及评价指标中的 GMUR 和 RTF值,所以论文根据这些开源方法,逐一搭建了网络模型并在两个目标数据集上完成了训练。表4-2展示了本章所提模型主要超参数的配置信息。

对于本章所提模型,论文采用联合训练的模式来综合考虑意图检测和槽位填充两个子任务,联合训练的损失函数由它们各自的损失函数按一定的比例组合而

成,定义如下:

(4-20)

(4-21)

(4-22)

式中,、和分别表示联合训练任务、意图检测任务和槽位填充任务的损失函数;和均为超参数;表示输入话语序列的长度;表示意图标签个数;表示槽位标签个数;和分别表示真实的和预测的意图结果。在本章所提模型的训练过程中,和分别设置为0.85和0.15。训练过程中各个模型在验证集上的损失(Loss)曲线如图4-6所示(从第10次迭代开始)。在所有对比组实验中,论文选择在验证集上表现最佳的模型作为训练好的模型,然后在测试集上对其进行测试以进行公平比较。此外,在模型构建过程中没有使用任何例如以 C++语言进行编程的加速技巧。

表4-2本章所提模型主要超参数的配置信息

Table4-2 Configuration of the main hyper-parameters of the model proposed in this chapter

名称 MixATIS MixSnips 名称 MixATIS MixSnips 隐藏单元维度256256优化器 Adam Adam 字符向量维度12864学习率 Learning Rate 1e-31e-3

标签向量维度256128权重衰减1e-61e-6

表示子空间个数48 Dropout 概率0.40.4

图层数22滑动窗口大小21

批处理大小 Batch Size 64128阈值0.50.5

迭代次数 Epoch 200200阈值0.50.5

图4-6训练过程中 MixATIS(左侧)和 MixSnips(右侧)验证集上各个

模型的损失曲线

Fig.4-6 Loss curves for each model on the validation subset of MixATIS (left)

and MixSnips (right) during training

重庆邮电大学硕士学位论文

4.5.4对比实验结果及分析

将本章提出的模型和其他同类模型在 MixATIS 和 MixSnips 测试集上的实验

结果进行对比,具体信息分别见表4-3和表4-4。表中标"\*"表示未开源的方法,

"↑"/"↓"分别表示该项指标越高/低越好,粗体数字表示最优结果,下划线数字表示次优结果。

表4-3 MixATIS 上各个模型的实验结果

Table4-3 Experimental results of each model on MixATIS

模型

OAcc (%)

↑

IAcc (%)

↑

F1(%)

↑

GMUR (%)

↓

Latency (ms)

↓

Attention BiRNN\*[85]39.174.686.4

Slot-Gated[29]35.563.987.746.2411.7±1.1

Bi-Model\*[53]34.470.383.9

SF-ID[32]34.966.287.448.3514.2±1.3

Stack-Propagation[31]39.676.286.555.7336.6±2.1

Joint Multiple ID-SF\*[86]36.173.484.6

AGIF[81]40.874.486.753.787.3±0.5

SDJN\*[87]44.677.188.2

Proposed 49.977.888.344.629.5±0.3

表4-4 MixSnips 上各个模型的实验结果

Table4-4 Experimental results of each model on MixSnips

模型

OAcc (%)

↑

IAcc (%)

↑

F1(%)

↑

GMUR (%)

↓

Latency (ms)

↓

Attention BiRNN\*[85]59.595.489.4

Slot-Gated[29]55.494.687.948.2512.3±1.7

Bi-Model\*[53]63.495.690.7

SF-ID[32]59.995.090.645.7714.6±1.2

Stack-Propagation[31]72.496.293.744.5437.7±2.3

Joint Multiple ID-SF\*[86]62.995.190.6

AGIF[81]74.295.194.243.147.2±0.4

SDJN\*[87]75.796.594.4

Proposed 77.397.194.840.6610.2±0.3

从表4-3中可以看出,在 MixATIS 上,与对比组中预测效果最好的 SDJN 相比,本章所提模型的整体准确率 OAcc、意图准确率 IAcc 和槽位 F1值分别上升了

5.3%、0.7%和0.1%;与对比组中 GMUR 最小的 Slot-Gated 相比,GMUR 下降了

1.62%;与对比组中 Latency 值最小的 AGIF 相比,Latency 的均值上升了3ms(在实际应用中影响较小)。本章所提模型在 MixSnips 上的结果与在 MixATIS 上类似。

为了更加直观地看出所提方法的效果,论文将上述结果进行可视化处理。具体地,选择整体准确率 OAcc、GPU 内存占用率和推理时间 Latency 均值等指标,将 OAcc 作为纵轴、其他两项分别作为横轴绘制散点图,如图4-7所示。可以看出,本章所提模型的对应点在两幅图上都位于最高点,除了在 Latency 值方面略高于 AGIF,在整体准确率和 GPU 内存占用率两方面都达到了最优,进一步证明了所提方法的有效性,即优化了特征捕获和交互方式,实现模型交互能力和预测精度的提高。

图4-7各个模型实验结果的可视化处理。(a) OAcc与GMUR的关系;(b)

OAcc 与 Latency 的关系

Fig.4-7 Visualization of experimental results for each model.(a) Relation

between OAcc and GMUR;(b) Relation between OAcc and Latency

另外,为了更好地理解本章所提模型在双向交互过程中捕获的语义特征,分别使用来自 MixATIS 和 MixSnips 数据集的话语输入至模型并将其注意力权重可视化,纵轴为预测意图,横轴为输入话语,标\*表示含有槽位的字符,如图4-8所

示。对于每个权重块,颜色越深,表示意图和槽位的关联性越强。在 MixATIS 上,

例句具有两个意图 atis\_flight 和 atis\_meal,可以看到注意力权重成功地集中在正确的位置上,具体地,在槽位 delta 处正确聚合了 atis\_flight 意图信息,在槽位meal 处正确聚合了 atis\_meal 意图信息,表明模型能很好地关注每个槽位并合并相关意图信息。在 MixSnips 上的结果与在 MixATIS 上类似,在槽位 song 和 siesta

重庆邮电大学硕士学位论文

处聚合了 AddToPlaylist 意图,在槽位 shadow、of、suribachi、five 和 starts 处聚合了 RateBook 意图。

图4-8输入话语在本章所提模型上的注意力权重可视化。(a)来自

MixATIS 的话语;(b)来自 MixSnips 的话语

Fig.4-8 Visualization of the attention weights for input utterances on the models proposed in this chapter.(a) Utterance from MixATIS;(b) Utterance

from MixSnips

4.5.5消融实验结果及分析

论文围绕标签映射模块和全局图交互模块设计消融实验,探究它们对于整个模型性能提升的贡献程度。具体地,在本章所提模型的基础上,移除标签映射模块(Label Injection Module,LIM)、移除全局图交互模块(Global Graph Interaction Module,GGIM),从而得到两个变种模型,分别用 w/o LIM 和 w/o GGIM 表示,观察它们在 MixATIS 和 MixSnips 上的评价指标变化情况,具体信息

见表4-5。表中,Base 指本章所提模型,"↑"/"下"表示该项指标越高/低越好。

除了变化的模块,所有变种模型的其他部分(包括网络结构、实验环境和训练配置等)均与 Base 模型一致。

从表中可以看出,所有变种模型的预测精度相较于 Base 模型都有明显降低。具体地,在 MixATIS 上,若移除标签映射模块(w/o LIM),和 Base 模型相比,GPU 内存占用率 GMUR 下降了3.80%,Latency 值下降了1.2ms,二者在实际应用中均影响较小,但整体准确率 OAcc、意图准确率 IAcc 和槽位 F1值分别下降了

7.6%、1.4%和1.2%;若移除全局图交互模块(w/o GGIM),和 Base 模型相比

GPU 内存占用率 GMUR 下降了2.96%,Latency 值下降了0.8ms,但整体准确率OAcc、意图准确率 IAcc和槽位F1值分别下降了6.8%、0.6%和2.6%。在MixSnips上的结果与在 MixATIS 上类似。

表4-5消融实验结果

Table4-5 Ablation study results

数据集模型

OAcc (%)

↑

IAcc (%)

↑

F1(%)

↑

GMUR (%)

↓

Latency (ms)

↓

MixATIS

Base 49.977.888.344.629.5±0.3

w/o LIM

42.3

(-7.6)

76.4

(-1.4)

87.1

(-1.2)

40.82

(-3.80)

8.3±0.7

(-1.2)

w/o GGIM

43.1

(-6.8)

77.2

(-0.6)

85.7

(-2.6)

41.66

(-2.96)

8.7±0.8

(-0.8)

MixSnips

Base 77.397.194.840.6610.2±0.3

w/o LIM

72.2

(-5.1)

95.5

(-1.6)

93.8

(-1.0)

37.35

(-3.31)

9.4±0.4

(-0.8)

w/o GGIM

74.0

(-3.3)

96.4

(-0.7)

92.5

(-2.3)

36.92

(-3.74)

8.1±1.2

(-2.1)

注:括号内数字表示该模型的该项指标值与 Base 模型的差异。红色表示结果变好,绿色表示结果变差。

为了更加直观地看出各变种模型和 Base 模型的区别,论文将上述结果进行可视化处理。具体地,将意图准确率 IAcc 作为纵轴、槽位 F1值作为横轴绘制散点图,如图4-9所示。

图4-9各个变种模型意图准确率 IAcc 和槽位 F1值关系的可视化

Fig.4-9 Visualization of the relation between intent accuracy and slot F1 for each variant model

从上图中可以看出,在意图准确率 IAcc 方面,w/o LIM 相较于 Base 模型的下降程度大于 w/o GGIM,说明标签映射模块可以获取原始话语和标签语义之间的

重庆邮电大学硕士学位论文

相关性,以辅助意图检测子任务;在槽位 F1值方面,w/o GGIM 相较于 Base 模型的下降程度大于 w/o GGIM,说明全局图交互模块可以对语句级别的意图—槽位交互过程进行建模,并通过全局优化辅助槽位填充子任务,从而显著提高模型性能,进一步论证了所提方法的有效性。

4.6本章小结

本章工作在经典显式联合建模方法的基础上,侧重于对特征捕获和交互方式进行优化,实现模型交互能力和预测精度的提高。首先提出标签映射模块获取原始话语和标签语义之间的相关性以提供丰富的先验知识,然后提出全局图交互模块对语句级别的意图—槽位交互过程进行建模以提供全局优化,从而提升模型性能。在MixATIS和MixSnips数据集上的实验结果论证了本章所提方法的有效性。

Equation Chapter (Next) Section (Next)

第5章面向车载嵌入式设备的本地智能语音对话系统

5.1引言

当前主流的车载语音对话系统均采用"云—端"方式运行,存在着一定的数据安全隐患。为解决这类问题,面向车载嵌入式设备,研发离线条件下的本地智能语音对话是有效的技术途径。具体地,首先选取 Nvidia Jetson TX2作为车载嵌入式设备并进行刷机、配置环境等操作,然后根据实际应用场景收集、创建驾驶数据集,接着将论文第3章、第4章提出的模型分别在驾驶数据集上进行训练,最后集成、移植网络模型至 TX2并围绕搭建全套硬件平台,实现数据安全、自然实时的离线智能语音对话。本章会对系统搭建的细节进行详细介绍并完成相应的系统测试。

5.2嵌入式设备运行环境搭建

嵌入式设备是指将计算能力和控制功能嵌入到设备中的计算机系统,通常具有小巧、低功耗、高效率和低成本等特点,广泛应用于工业自动化、医疗设备和安防监控等领域。英伟达(Nvidia)公司推出的 Jetson TX2是一款高性能、低功耗的嵌入式计算平台,其外观如图5-1(a)所示。它的基于 ARM 架构的 CPU 和Nvidia Pascal 架构的 GPU,能够提供强大的计算和图形处理能力,非常适合于嵌入式设备应用[88]。相较于上一代的 TX1,TX2内存和 eMMC(Embedded Multi

Media Card)提高一倍,CUDA(Compute Unified Device Architecture)架构升级为 Pascal,每瓦性能提高一倍。同时,由于支持 PyTorch、TensorFlow 和 Caffe 等深度学习框架,TX2也是一款非常适合深度学习应用的嵌入式设备,能够处理复杂的神经网络模型和算法,其主要硬件配置信息见表5-1。

表5-1 TX2主要硬件配置信息

Table5-1 Main hardware configuration of TX2

类型规格类型规格

算力1.33TFLOPS 内存

8GB 128-bit

LPDDR459.7GB/s

GPU 256-core NVIDIA PascalTM GPU 存储32GB eMMC 5.1

CPU

Dual-Core NVIDIA Denver 1.564-Bit CPU 和 Quad-

Core ARM®Cortex®-A57 MPCore processor

功率7.5W/15W

USB USB 3.0和 USB 2.0尺寸87mm 50mm

重庆邮电大学硕士学位论文

综上所述,Nvidia Jetson TX2是嵌入式 AI 应用的理想选择,所以论文选择该器件作为搭建本地智能语音对话系统的核心嵌入式设备。下面对 TX2的运行环境搭建步骤进行详细介绍。

首先按照官方教程,将 TX2与 PC 虚拟机(Ubuntu18.04系统)连接,在 PC虚拟机上安装刷机软件SDK Manager并下载一体化软件包 JetPack(版本为4.5.1),

即可开始刷机操作,中间操作界面和完成界面如图5-1(b)和(c)所示;然后需要安装神经网络模型训练和推理所需的深度学习框架和各种第三方库。具体地,打开TX2终端窗口执行教程上的相关安装命令,如图5-1(d)所示。值得注意的是,由于在 TX2上没有 NVSMI 工具,无法使用 nvidia-smi 命令查看 GPU 运行状态,所

以需要额外安装 Jtop 工具作为替代,其安装命令为 sudo -H pip3 install -U jetson-

stats,安装完成后,可直接使用 Jtop 命令查看 CPU 运行状态、GPU 运行状态、内存和软、硬件信息(如图5-1(e)所示),以及使用 pip3 list 命令查看所有已安装的和运行环境有关的软件包,如图5-1(f)所示。至此,TX2的运行环境搭建完成。

图5-1 TX2外观及运行环境搭建。(a) TX2外观;(b)刷机中间操作界面;(c)刷机完成界面;(d)深度学习框架和第三方库的安装命令;(e) Jtop

命令查看软、硬件信息;(f) pip3 list 命令查看所有软件包

Fig.5-1 TX2 appearance and operating environment construction.(a) Appearance of TX2;(b) Intermediate operation interface of flashing OS;(c) Complete interface of flashing OS;(d) Installation command of deep learning framework and third-party library;(e) Jtop command to view software and

hardware information;(f) pip3 list command to view all packages

5.3驾驶数据集收集

由于所搭建的本地智能语音对话系统专用于驾驶员和车载增强现实平视显示器(Augmented Reality Head-Up Display,ARHUD)系统的语音交互,所以需要从使用 ARHUD 的真实场景中收集音频数据。此外,为了增加数据量且降低收集成本,设计以微信小程序为载体的调查问卷,可从移动端(手机端)收集更多数据。下面对数据收集平台和收集过程中的细节进行详细介绍。

5.3.1数据收集平台

数据收集平台包括与 ARHUD 相连的车载收集平台和手机云收集平台,如图

5-2所示。

图5-2数据收集平台。(a)车载收集平台(驾驶室);(b)车载收集平台

(后备箱);(c)手机云收集平台

Fig.5-2 Data collection platform.(a) In-vehicle collection platform (cab);(b) In-vehicle collection platform (trunk);(c) mobile phone cloud collection

platform

重庆邮电大学硕士学位论文

在车载收集平台上,主要硬件单元包括:麦克风阵列、触发器(按钮)和TX2。麦克风和触发器放置于驾驶室,TX2放置于后备箱,它们通过通用串行总线(Universal Serial Bus)进行连接。麦克风阵列包含4个 MEMS(Micro Electro Mechanical System)麦克风单元的,采样率为16bit/44.1Khz。每个麦克风单元都配备了数字信号处理器(Digital Signal Processor,DSP)用于降低车辆噪音,如发动机噪音、传输噪音、轮胎噪音和空气动力噪音等。为了避免在收集过程中与志愿者交流造成的语音干扰,设计了一个触发器来手动控制麦克风阵列接收的语音输入。TX2用于存储收集的音频数据,为了扩展存储空间,TX2上额外外接了一张8G 的 SD 卡(Secure Digital Memory Card)。在手机云收集平台上,所有收集的音频数据存储于后台服务器。

5.3.2数据收集过程

根据 ARHUD 的功能和使用场景,论文按6个功能类别(包含25个意图)收集音频数据,分别是:关机、切换主题、调整高度和亮度、控制驾驶信息、控制导航和控制其他功能。在整个收集过程中,共有570名志愿者参与,年龄范围在

18~45岁,男女比例约为1.54:1,具体信息见表5-2。所有的音频数据都是在志愿者和工作人员共同驾车的过程中收集的。实验地位于重庆市渝北区鱼嘴工业园区的非公共道路,主要包括一些位于不同工业园区区域的专用测试路线,全长约

7公里。汽车的行驶速度取决于驾驶员的驾驶水平,但不允许超过道路限速。

表5-2收集的数据集的详细信息

Table5-2 Details of the collected dataset

数据集信息功能类别音频数

来自车载收集

平台的数量

来自手机云收集

平台的数量

男性人数346关机24742187287

女性人数224切换主题24992017482

时长(小时)52调整高度和亮度14198131451053

采样率(Hz)16000控制行车信息70426348694

通道数1控制导航46604092568

比特率(bps)96000控制其他功能27065240533012

帧大小(KB)50合计57983518426096

为保证数据的真实性,需要志愿者根据自己的驾驶习惯完成实验。在每次驾

车中,志愿者在工作人员(副驾驶位)的引导下行驶约15分钟(路程约3公里),

根据所需的 ARHUD 功能说出相应的命令词,负责数据采集其他工作(监控设备运行状态是否正常)的工作人员坐在副驾驶后面。

在 TX2和后台服务器上的初始语音数据集包含近70000个音频文件,总时长约为75小时。首先对其进行数据清理操作,去除噪音过大、音量过小和格式损坏的音频文件,然后聘请专业的数据处理团队对数据集进行文本标记。经过以上数据处理工作,最终得到了完整的驾驶数据集,将其命名为 CQUPT-DS,包含近

58000个音频文件,总时长约51小时,子集划分见表5-3。

表5-3 CQUPT-DS 的划分信息

Table5-3 Split of CQUPT-DS

语音段数/语句数时长(小时)

训练集(Train)5194045.7

验证集(Val)30002.7

测试集(Test)29982.6

5.4模型训练与移植

在完成驾驶数据集的收集工作之后,将第3章和第4章提出的模型分别在CQUPT-DS 上进行训练,并将训练好的模型搭配其他对话系统中的模块进行集成与移植。

5.4.1自动语音识别模型训练

将第3章提出的模型在 CQUPT-DS 上进行训练,同时为了验证所提模型在CQUPT-DS 上的适应性,根据节3.5设置对比实验进行比较,表5-4展示了各个模型主要超参数的配置信息,包含编码器激活函数(E-AF)、解码器激活函数(D-AF)、优化器(OP)、热身步数(WU)、学习率(LR)、Dropout 概率(DR)、批处理大小(BS)和迭代次数(EP)。

表5-4各个模型在 CQUPT-DS 上主要超参数的配置信息

Table5-4 Configuration of the main hyper-parameters of each model on CQUPT-DS 模型 E-AF D-AF OP WU LR DR BS EP Speech-Transformer[73] ReLU ReLU Adam 2.3e41.20.10160100

HA-Transformer[74] GLU GLU Adam 2.3e41.20.10128100

STBD[75] ReLU ReLU Adam 1.4e41.20.1096100

Proposed Swish ReLU Adam 2.3e41.20.15128100

重庆邮电大学硕士学位论文

训练过程中各个模型在验证集上的损失(Loss)曲线如图5-3所示(从第10次迭代开始)。

图5-3训练过程中 CQUPT-DS 验证集上各个 ASR 模型的损失曲线

Fig.5-3 Loss curves for each ASR model on the validation subset of CQUPT-DS during training

对比实验结果见表5-5,表中 MACs 值是在编码序列长度为500、解码序列长度为30的条件下计算得出,RTF 值是在 TX2上计算得出,"↓"表示该项指标越低越好,粗体数字表示最优结果,下划线数字表示次优结果。

表5-5 CQUPT-DS 上各个模型的实验结果

Table5-5 Experimental results of each model on CQUPT-DS

模型

CER (%)↓ Parameters (M)

↓

MACs (B)

↓

GMUR (%)

↓

RTF

↓ Val Test

Speech-Transformer[73]8.578.0628.218.173.560.117

HA-Transformer[74]8.267.6939.523.271.490.121

STBD[75]7.457.3253.859.363.680.182

Proposed 7.267.2620.45.354.070.198

从表中可以看出,第3章所提模型在 CQUPT-DS 上除了 RTF 值外,在测试集上的 CER、参数量、计算量和 GMUR 都达到了最优。具体地,在 CER 方面,相较于 HA-Transformer 下降了0.43%;在参数量和计算量方面,相较于 Speech-

Transformer 分别相对下降了27.66%和70.72%;在 GMUR 方面,相较于 STBD 下降了9.61%。表明所提模型在 CQUPT-DS 上也有较强的可适应性。

5.4.2自然语言理解模型训练

将第4章提出的模型在 CQUPT-DS 上进行训练,同时为了验证所提模型在CQUPT-DS 上的适应性,根据节4.5设置对比实验进行比较,表5-6展示了所提模型主要超参数的配置信息。

表5-6所提模型在 CQUPT-DS 上主要超参数的配置信息

Table5-6 Configuration of the main hyper-parameters of the proposed model

名称值名称值

隐藏单元维度256优化器 Adam 字符向量维度128学习率 Leraning Rate 1e-3

标签向量维度256权重衰减1e-6

表示子空间个数4 Dropout 概率0.4

图层数2滑动窗口大小1

批处理大小 Batch Size 256阈值0.5

迭代次数 Epoch 200阈值0.5

在第4章所提模型的训练过程中,和分别设置为0.9和0.1。训练过程中各个模型在验证集上的损失(Loss)曲线如图5-4所示(从第10次迭代开始)。

图5-4训练过程中 CQUPT-DS 验证集上各个 NLU 模型的损失曲线

Fig.5-4 Loss curves for each NLU model on the validation subset of CQUPT-DS during training

对比实验结果见表5-7,表中"↑"/"↓"分别表示该项指标越高/低越好,粗体数字表示最优结果,下划线数字表示次优结果。从表中可以看出,第4章所提模型在 CQUPT-DS 上除了 Latency 值外,整体准确率、意图准确率、槽位 F1值和GMUR都达到了最优。其中,在整体准确率OAcc、意图准确率 IAcc和槽位 F1

重庆邮电大学硕士学位论文

值方面,相较于 AGIF 分别上升了3.3%、2.1%和0.3%;在 GMUR 方面,相较于SF-ID 下降了1.06%。表明所提模型在 CQUPT-DS 上也有较强的可适应性。

表5-7 CQUPT-DS 上各个模型的实验结果

Table5-7 Experimental results of each model on CQUPT-DS

模型

OAcc (%)

↑

IAcc (%)

↑

F1(%)

↑

GMUR (%)

↓

Latency (ms)

↓

Slot-Gated[29]58.793.989.349.2314.4±1.3

SF-ID[32]61.294.491.642.8313.1±0.4

Stack-Propagation[31]75.496.093.958.8632.6±3.5

AGIF[81]78.294.794.851.579.5±0.4

Proposed 81.596.895.141.7712.8±1.6

5.4.3模型集成与移植

模型训练完成后,根据节2.1并参考其他模块的开源方法,完成对话系统的集成,如图5-5所示。这些开源模块方法包括:DST-as-Prompting[89]、Deep Dyna-

Q[90]、KENLG-Reading[91]和 FastPitch[92]等。

集成完成后,将系统模型移植至已经搭建了运行环境的 TX2上。为了观察TX2运行对话系统时的性能表现,使用 top 和 Jtop 命令查看 TX2的 CPU 占用、GPU 占用和内存占用情况,如图5-6所示。

具体地,系统启动后,经过约10s 所有网络模型均加载完成,系统处于等待音频输入的状态(如图5-6(a)所示);在模型加载和模型推理时,CPU 的最大占用率为74.6%(如图5-6(b)所示);GPU 的最大占用率接近85%,在等待音频输入时为55%左右(如图5-6(c)所示);内存占用为60%(如图5-6(d)所示)。表明对话系统能在嵌入式设备 TX2上较好地运行。

图5-5集成的对话系统

Fig.5-5 Integrated dialog system

图5-6 TX2运行对话系统时的性能表现。(a)运行界面;(b) CPU 占用;

(c) GPU 占用;(d)内存占用

Fig.5-6 Performance of TX2 running the dialogue system.(a) Running

interface;(b) CPU usage;(c) GPU usage;(d) Memory usage

5.5系统硬件平台搭建与测试

5.5.1平台搭建

在完成模型的移植工作之后,围绕 TX2搭建全套硬件平台,如图5-7所示。系统运行的基本流程为:驾驶员在驾驶位上,按住触发器(录音开始)说出命令词,结束后松开触发器(录音结束),麦克风阵列将音频传输至 TX2进行处理,处理结果以 Socket 通信方式(车载内网)传输给车载 ARHUD,并在车前窗上的特定区域显示,同时播放语音给予驾驶员反馈。例如,驾驶员命令词为"我想看看别的行车信息",ARHUD 会在特定区域内切换下一个行车信息,同时播放的语音内容为"好的,已为您切换行车信息,当前信息:XXX"。

值得说明的是,由于触发器的存在,可能会出现驾驶员误触的情况,这时麦克风阵列录制的音频为空白或者音量很小,会影响到后续模块的有效运行。所以在 TX2中(ASR 模块前)加入逻辑控制,当检测到空白和音量很小()

重庆邮电大学硕士学位论文

的音频时进行特殊处理,即直接跳过后续模块并播放语音"说话音量较小,请适当提高音量"以提醒驾驶员。

图5-7系统运行平台。(a)驾驶室;(b)后备箱

Fig.5-7 System operating platform.(a) Cab;(b) Trunk.

5.5.2系统测试

整套系统已在实验用车上完成搭建,支持 ARHUD 包含的25种意图/功能,最大对话轮次达到5轮。整套系统已通过由重庆利龙智能汽车研究院主导的内部项目验收测试,对话通过率为97%,平均响应时间为0.87s,具体结果见表5-8。

表5-8项目验收结果

Table5-8 Project acceptance results

测试项目测试次数测试通过次数通过率判定结果平均响应时间不符合项概述

对话系统626097% OK 0.87s 无

5.6本章小结

本章工作为面向车载嵌入式设备研发离线条件下的本地智能语音对话系统。首先选取 Nvidia Jetson TX2作为车载嵌入式设备并进行刷机、配置环境等操作,然后根据实际应用场景收集、创建驾驶数据集,接着将论文第3章、第4章提出的模型分别在驾驶数据集上进行训练,最后集成、移植网络模型至 TX2并围绕搭建全套硬件平台。由重庆利龙智能汽车研究院主导的内部项目验收测试结果论证了本章工作的有效性。

6.1总结

对话系统是智能语音交互的一种具体技术形式,在车载终端控制、智能客服和智能家居等领域有着广泛的应用。作为对话系统的重要技术,自动语音识别和自然语言理解是近年来备受关注的研究热点。如何解决对话系统由于当前"云—端"运行方式带来的数据安全隐患,实现自动语音识别和自然语言理解技术在车载平台上的高可靠性、强实时性应用,研发离线条件下的智能语音对话也是当前的研究难点。论文首先阐述了自动语音识别和自然语言理解技术的研究现状,然后分析现有方法存在的问题,提出了相应的改进方案,最后基于改进的网络模型搭建了车载对话系统,实现了数据安全、自然实时的离线智能语音交互。论文的

主要研究成果如下:

(1)在对现有自动语音识别技术的改进方面,针对基于深度编—解码器的模型参数量庞大的问题,提出了一种基于残差分组线性变换的解码器结构。该结构关键模块为"钻石"型缩放单元,其内部采用稀疏连接,同一组神经元共享相同的权重矩阵。在 AISHELL-1和 TED-LIUM2数据集上的实验结果表明,所提模型能以识别准确率的小幅度降低为代价,实现参数量和计算复杂度的降低。

(2)在对现有自然语言理解技术的改进方面,针对基于显式联合建模的模型交互能力不足的问题,提出了一种基于标签感知的图交互模型。其中标签映射模块可以获取原始话语与标签语义之间的相关性以提供丰富的先验知识,全局图交互模块可以对语句级别的意图—槽位交互过程进行建模以提供全局优化。在MixATIS 和 MixSnips 数据集上的实验结果表明,所提模型能对特征捕获和交互方式进行优化,实现交互能力和预测精度的提高。

(3)在对现有车载语音对话系统的改进方面,针对"云—端"方式存在数据安全隐患的问题,搭建了面向车载嵌入式设备的本地智能语音对话系统。具体地,首先选取 Nvidia Jetson TX2作为车载嵌入式设备并进行刷机、配置环境等操作,然后根据实际应用场景收集、创建驾驶数据集 CQUPT-DS,接着将研究成果

(1)和(2)中的模型在 CQUPT-DS 上进行训练,最后集成、移植网络模型至TX2并围绕搭建全套硬件平台。由重庆利龙智能汽车研究院主导的内部项目验收测试结果表明,对话系统能实现数据安全、自然实时的离线运行。

重庆邮电大学硕士学位论文

6.2展望

论文对自动语音识别和自然语言理解模型进行了改进,在降低参数量和计算复杂度的同时提高了识别性能,同时搭建了车载智能语音对话系统,实现了数据安全、自然实时的离线智能语音交互。然而论文中仍有一些不足之处,后续研究

工作可以从以下两个方面展开:

(1)流式自动语音识别。论文提出的自动语音识别模型不支持流式识别,即无法在输入语音信号的同时进行识别,这会在一些实时性要求较高的场景中导致语音识别的体验降低。下一步可以考虑使用分块技术对编码器网络中的注意力机制进行优化,以建立在当前输入时刻语音帧和历史信息之间的联系,从而支持流式自动语音识别。

(2)非自回归式解码。论文提出的自动语音识别和自然语言理解模型均采用自回归式解码,即解码器需要逐个输出预测的字符。当序列长度增加,这种方式的计算复杂程度会大大提升,可能会影响模型推理效率。下一步可以考虑采用非自回归的训练方式对模型进行训练,以进一步提高模型的预测速度。

参考文献

[1] SEABORN K, MIYAKE N P, PENNEFATHER P, et al. Voice in human-agent interaction: a survey[J]. ACM Computing Surveys, 2021, 54(4): 1-43.

[2] 中泰证券研究所. Cerence:人工智能助力未来出行—车智能化海外公司系列报告[EB/O

L]. (2021-04-25) [2022-12-25]. https: pdf.dfcfw.com/pdf/H3\_AP202104261487921362\_1.pdf?1619458440000.pdf.

[3] 陈艳华. 基于智能交互的车载语音系统的设计与实现[D]. 北京: 北京交通大学, 2020: 1-2.

[4] DERIU J, RODRIGO A, OTEGI A, et al. Survey on evaluation methods for dialogue systems[J]. Artificial Intelligence Review, 2021, 54(1): 755-810.

[5] RIGOLL G, NEUKIRCHEN C. A new approach to hybrid HMM/ANN speech recognition

using mutual information neural networks[C]. 10th Conference on Neural Information Processing Systems, Denver, USA, 1996: 772-778.

[6] MOHAMED A, DAHL G, HINTON G. Deep belief networks for phone recognition[C]. 23rd

Conference Neural Information Processing Systems Workshop on Deep Learning for Speech Recognition and Related Applications, Vancouver, Canada, 2009: 39-39.

[7] ABDEL-HAMID O, MOHAMED A, JIANG H, et al. Convolutional neural networks for speech

recognition[J]. IEEE-ACM Transactions on Audio, Speech, And Language Processing, 2014, 22(10): 1533-1545.

[8] GRAVES A. Connectionist temporal classification[M]. Springer, Berlin, Heidelberg: Supervised Sequence Labelling with Recurrent Neural Networks, 2012: 61-93.

[9] SUTSKEVER I, VINYALS O, LE Q V. Sequence to sequence learning with neural networks[C]. 28th Conference Neural Information Processing Systems, Montreal, Canada, 2014: 3104-3112. [10] 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, Pittsburgh, USA, 2006: 369-376.

[11] GRAVES A. Sequence transduction with recurrent neural networks[J]. arXiv preprint arXiv:1211.3711, 2012.

[12] GRAVES A, MOHAMED A, HINTON G. Speech recognition with deep recurrent neural

networks[C]. 38th IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, Canada, 2013: 6645-6649.

重庆邮电大学硕士学位论文

[13] GRAVES A. Generating sequences with recurrent neural networks[J]. arXiv preprint arXiv:1308.0850, 2013.

[14] BAHDANAU D, CHO K, BENGIO Y. Neural machine translation by jointly learning to align and translate[J]. arXiv preprint arXiv:1409.0473, 2014.

[15] CHOROWSKI J, BAHDANAU D, CHO K, et al. End-to-end continuous speech recognition using attention-based recurrent NN: first results[J]. arXiv preprint arXiv:1412.1602, 2014.

[16] BAHAR P, ZEYER A, SCHLÜTER R, et al. On using 2d sequence-to-sequence models for

speech recognition[C]. 44th IEEE International Conference on Acoustics, Speech and Signal Processing, Brighton, UK, 2019: 5671-5675.

[17] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[C]. 31st Conference Neural Information Processing Systems, Long Beach, USA, 2017: 5998-6008.

[18] ZHOU S, DONG L, XU S, et al. A comparison of modeling units in sequence-to-sequence

speech recognition with the transformer on mandarin Chinese[C]. 25th International Conference on Neural Information Processing, Siem Reap, Cambodia, 2018: 210-220.

[19] MERBOLDT A, ZEYER A, SCHLÜTER R, et al. An analysis of local monotonic attention

variants[C]. 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 2019: 1398-1402.

[20] ZEYER A, IRIE K, SCHLÜTER R, et al. Improved training of end-to-end attention models for speech recognition[J]. arXiv preprint arXiv:1805.03294, 2018.

[21] GULATI A, QIN J, CHIU C C, et al. Conformer: convolution-augmented transformer for speech recognition[J]. arXiv preprint arXiv:2005.08100, 2020.

[22] QIN L, XIE T, CHE W, et al. A survey on spoken language understanding: recent advances and new frontiers[J]. arXiv preprint arXiv:2103.03095, 2021.

[23] RAVURI S, STOLCKE A. Recurrent neural network and LSTM models for lexical utterance

classification[C]. 16th Annual Conference of the International Speech Communication

Association, Dresden, Germany, 2015: 35-40

[24] YAO K, ZWEIG G, HWANG M Y, et al. Recurrent neural networks for language

understanding[C]. 14th Annual Conference of the International Speech Communication Association, Lyon, France, 2013: 2524-2528.

[25] MESNIL G, HE X, DENG L, et al. Investigation of recurrent-neural-network architectures and

learning methods for spoken language understanding[C]. 14th Annual Conference of the International Speech Communication Association, Lyon, France, 2013: 3771-3775.

[26] MESNIL G, DAUPHIN Y, YAO K, et al. Using recurrent neural networks for slot filling in

spoken language understanding[J]. IEEE-ACM Transactions on Audio, Speech, and Language Processing, 2014, 23(3): 530-539.

[27] ZHANG X, WANG H. A joint model of intent determination and slot filling for spoken

language understanding[C]. 25th International Joint Conference on Artificial Intelligence, New York, USA, 2016: 2993-2999.

[28] HAKKANI-TÜR D, TÜR G, CELIKYILMAZ A, et al. Multi-domain joint semantic frame

parsing using bi-directional RNN-LSTM[C]. 17th Annual Conference of the International Speech Communication Association, San Francisco, USA, 2016: 715-719.

[29] GOO C W, GAO G, HSU Y K, et al. Slot-gated modeling for joint slot filling and intent

prediction[C]. Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), New Orleans, USA, 2018: 753-757.

[30] LI C, LI L, QI J. A self-attentive model with gate mechanism for spoken language

understanding[C]. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 2018: 3824-3833.

[31] Qin L, Che W, Li Y, et al. A stack-propagation framework with token-level intent detection for

spoken language understanding[C]. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Hong Kong, China, 2019: 2078-2087.

[32] NIU P, CHEN Z, SONG M. A novel bi-directional interrelated model for joint intent detection and slot filling[J]. arXiv preprint arXiv:1907.00390, 2019.

[33] ZHANG C, LI Y, DU N, et al. Joint slot filling and intent detection via capsule neural

networks[C]. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 2019: 5259-5267.

[34] LIU Y, MENG F, ZHANG J, et al. CM-Net: a novel collaborative memory network for spoken

language understanding[C]. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Hong Kong, China, 2019: 1051-1060.

[35] ZHANG L, MA D, ZHANG X, et al. Graph LSTM with context-gated mechanism for spoken

language understanding[C]. Proceedings of the 34th AAAI Conference on Artificial Intelligence, New York, USA, 2020: 9539-9546.

重庆邮电大学硕士学位论文

[36] Qin L, Liu T, Che W, et al. A co-interactive transformer for joint slot filling and intent

detection[C]. 46th IEEE International Conference on Acoustics, Speech and Signal Processing, Toronto, Canada, 2021: 8193-8197.

[37] DEVLIN J, CHANG M W, LEE K, et al. BERT: pre-training of deep bidirectional transformers

for language understanding[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), Minneapolis, USA, 2019: 4171-4186.

[38] CHEN Q, ZHUO Z, WANG W. Bert for joint intent classification and slot filling[J]. arXiv preprint arXiv:1902.10909, 2019.

[39] CHEN H, LIU X, YIN D, et al. A survey on dialogue systems: recent advances and new frontiers[J]. ACM SIGKDD Explorations Newsletter, 2017, 19(2): 25-35.

[40] MCTEAR M F. Spoken dialogue technology: enabling the conversational user interface[J]. ACM Computing Surveys, 2002, 34(1): 90-169.

[41] 赵新颜. 基于深度学习的对话系统研究与应用[D]. 合肥: 中国科学技术大学, 2022: 13-14. [42] NI J, YOUNG T, PANDELEA V, et al. Recent advances in deep learning based dialogue systems: a systematic survey[J]. Artificial Intelligence Review, 2022: 1-101.

[43] JORGENSEN P E T, SONG M S, TIAN J. Operator theory, kernels, and feedforward neural networks[J]. arXiv preprint arXiv:2301.01327, 2023.

[44] RUMELHART D E, HINTON G E, WILLIAMS R J. Learning representations by back-propagating errors[J]. Nature, 1986, 323(6088): 533-536.

[45] LI Z, LIU F, YANG W, et al. A survey of convolutional neural networks: analysis, applications,

and prospects[J]. IEEE Transactions on Neural Networks and Learning Systems, 2021, 33(2): 6999-7019.

[46] HEWAMALAGE H, BERGMEIR C, BANDARA K. Recurrent neural networks for time series

forecasting: current status and future directions[J]. International Journal of Forecasting, 2021, 37(1): 388-427.

[47] HOCHREITER S, SCHMIDHUBER J. Long short-term memory[J]. Neural Computation, 1997, 9(8): 1735-1780.

[48] YAO S, WAN X. Multimodal transformer for multimodal machine translation[C]. Proceedings

of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 2020: 4346-4350.

[49] DU Y, PEI B, ZHAO X, et al. Deep scaled dot-product attention based domain adaptation model for biomedical question answering[J]. Methods, 2020, 173: 69-74.

[50] MIRSAMADI S, BARSOUM E, ZHANG C. Automatic speech emotion recognition using

recurrent neural networks with local attention[C]. 42nd IEEE International Conference on Acoustics, Speech and Signal Processing, New Orleans, USA, 2017: 2227-2231.

[51] CHAN W, JAITLY N, LE Q, et al. Listen, attend and spell: a neural network for large

vocabulary conversational speech recognition[C]. 41st IEEE International Conference on Acoustics, Speech and Signal Processing, Shanghai, China, 2016: 4960-4964.

[52] KARMAKAR P, TENG S W, LU G. Thank you for attention: a survey on attention-based

artificial neural networks for automatic speech recognition[J]. arXiv preprint arXiv:2102.07259, 2021.

[53] WANG Y, SHEN Y, JIN H. A bi-model based RNN semantic frame parsing model for intent detection and slot filling[J]. arXiv preprint arXiv:1812.10235, 2018.

[54] PHAM N Q, NGUYEN T S, NIEHUES J, et al. Very deep self-attention networks for end-to-end speech recognition[J]. arXiv preprint arXiv:1904.13377, 2019.

[55] SYNNAEVE G, XU Q, KAHN J, et al. End-to-end ASR: from supervised to semi-supervised learning with modern architectures[J]. arXiv preprint arXiv:1911.08460, 2019.

[56] LUO H, ZHANG S, LEI M, et al. Simplified self-attention for transformer-based end-to-end

speech recognition[C]. 8th IEEE Spoken Language Technology Workshop, Shenzhen, China, 2021: 75-81.

[57] ZHANG S, LIU C, JIANG H, et al. Feedforward sequential memory networks: a new structure to learn long-term dependency[J]. arXiv preprint arXiv:1512.08301, 2015.

[58] MEHTA S, GHAZVININEJAD M, IYER S, et al. Delight: deep and light-weight transformer[J]. arXiv preprint arXiv:2008.00623, 2020.

[59] WEN Q, ZHOU T, ZHANG C, et al. Transformers in time series: a survey[J]. arXiv preprint arXiv:2202.07125, 2022.

[60] MISRA D. Mish: a self regularized non-monotonic activation function[J]. arXiv preprint arXiv:1908.08681, 2019.

[61] LIN T, WANG Y, LIU X, et al. A survey of transformers[J]. AI Open, 2022, 1(1): 1-16.

[62] WANG D, ZHANG X. Thchs-30: A free Chinese speech corpus[J]. arXiv preprint arXiv:1512.01882, 2015.

[63] BU H, DU J, NA X, et al. Aishell-1: An open-source mandarin speech corpus and a speech

recognition baseline[C]. 20th Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment, Seoul, South Korea, 2017: 1-5.

重庆邮电大学硕士学位论文

[64] ROUSSEAU A, DELÉGLISE P, ESTEVE Y. Enhancing the TED-LIUM corpus with selected

data for language modeling and more TED talks[C]. Proceedings of the 9th International Conference on Language Resources and Evaluation, Reykjavik, Iceland, 2014: 3935-3939.

[65] WATANABE S, HORI T, KARITA S, et al. ESPNET: end-to-end speech processing toolkit[J]. arXiv preprint arXiv:1804.00015, 2018.

[66] MALIK M, MALIK M K, MEHMOOD K, et al. Automatic speech recognition: a survey[J]. Multimedia Tools and Applications, 2021, 80: 9411-9457.

[67] WHITEHEAD N, FIT-FLOREA A. Precision & performance: floating point and IEEE 754 compliance for NVIDIA GPUs[J]. RN (A+ B), 2011, 21(1): 18749-19424.

[68] HIGUCHI Y, INAGUMA H, WATANABE S, et al. Improved mask-CTC for non-

autoregressive end-to-end ASR[C]. 46th IEEE International Conference on Acoustics, Speech and Signal Processing, Toronto, Canada, 2021: 8363-8367.

[69] GHAHREMANI P, BABAALI B, POVEY D, et al. A pitch extraction algorithm tuned for

automatic speech recognition[C]. 39th IEEE International Conference on Acoustics, Speech and Signal Processing, Florence, Italy, 2014: 2494-2498.

[70] KUDO T, RICHARDSON J. SentencePiece: a simple and language independent subword

tokenizer and detokenizer for neural text processing[C]. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Brussels, Belgium, 2018: 66-71.

[71] POVEY D, PEDDINTI V, GALVEZ D, et al. Purely sequence-trained neural networks for ASR

based on lattice-free MMI[C]. 17th Annual Conference of the International Speech Communication Association, San Francisco, USA, 2016: 2751-2755.

[72] SHAN C, WENG C, WANG G, et al. Component fusion: learning replaceable language model

component for end-to-end speech recognition system[C]. 44th IEEE International Conference on Acoustics, Speech and Signal Processing, Brighton, UK, 2019: 5361-5635.

[73] DONG L, XU S, XU B. Speech-transformer: a no-recurrence sequence-to-sequence model for

speech recognition[C]. 43rd IEEE International Conference on Acoustics, Speech and Signal Processing, Calgary, Canada, 2018: 5884-5888.

[74] XU M, LI S, ZHANG X L. Transformer-based end-to-end speech recognition with local dense

synthesizer attention[C]. 46th IEEE International Conference on Acoustics, Speech and Signal Processing, Toronto, Canada, 2021: 5899-5903.

[75] CHEN X, ZHANG S, SONG D, et al. Transformer with bidirectional decoder for speech recognition[J]. arXiv preprint arXiv:2008.04481, 2020.

[76] PARK D S, CHAN W, ZHANG Y, et al. Specaugment: a simple data augmentation method for automatic speech recognition[J]. arXiv preprint arXiv:1904.08779, 2019.

[77] KANNAN A, WU Y, NGUYEN P, et al. An analysis of incorporating an external language

model into a sequence-to-sequence model[C]. 43rd IEEE International Conference on Acoustics, Speech and Signal Processing, Calgary, Canada, 2018: 1-5828.

[78] WU T W, SU R, JUANG B. A label-aware BERT attention network for zero-shot multi-intent

detection in spoken language understanding[C]. Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Online and Punta Cana, Dominican Republic, 2021: 4884-4896.

[79] DEL PINO G E, GALAZ H. Statistical applications of the inverse gram matrix: a revisitation[J]. Brazilian Journal of Probability and Statistics, 1995: 177-196.

[80] VELIČKOVIĆ P, CUCURULL G, CASANOVA A, et al. Graph attention networks[J]. arXiv preprint arXiv:1710.10903, 2017.

[81] QIN L, XU X, CHE W, ET AL. AGIF: An adaptive graph-interactive framework for joint multiple intent detection and slot filling[J]. arXiv preprint arXiv:2004.10087, 2020.

[82] HEMPHILL C T, GODFREY J J, DODDINGTON G R. The ATIS spoken language systems

pilot corpus[C]. Speech and Natural Language: Proceedings of a Workshop, Hidden Valley, USA, 1990.

[83] COUCKE A, SAADE A, BALL A, et al. Snips voice platform: an embedded spoken language

understanding system for private-by-design voice interfaces[J]. arXiv preprint arXiv:1805.10190, 2018.

[84] MIKOLOV T, CHEN K, CORRADO G, et al. Efficient estimation of word representations in vector space[J]. arXiv preprint arXiv:1301.3781, 2013.

[85] LIU B, LANE I. Attention-based recurrent neural network models for joint intent detection and slot filling[J]. arXiv preprint arXiv:1609.01454, 2016.

[86] GANGADHARAIAH R, NARAYANASWAMY B. Joint multiple intent detection and slot

labeling for goal-oriented dialog[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), Minneapolis, USA, 2019: 564-569.

[87] CHEN L, ZHOU P, ZOU Y. Joint multiple intent detection and slot filling via self-distillation[C].

47th IEEE International Conference on Acoustics, Speech and Signal Processing, Singapore, 2022: 7612-7616.

重庆邮电大学硕士学位论文

[88] SÜZEN A A, DUMAN B, ŞEN B. Benchmark analysis of jetson tx2, jetson nano and raspberry

pi using deep-CNN[C]. 2nd International Congress on Human-Computer Interaction, Optimization and Robotic Applications, Online, 2020: 1-5.

[89] LEE C H, CHENG H, OSTENDORF M. Dialogue state tracking with a language model using

schema-driven prompting[C]. Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Online and Punta Cana, Dominican Republic, 2021: 4937-4949. [90] PENG B, LI X, GAO J, et al. Deep Dyna-Q: integrating planning for task-completion dialogue

policy learning[C]. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Melbourne, Australia, 2018: 2182-2192. [91] YU W, JIANG M, HU Z, et al. Knowledge-enriched natural language generation[C].

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts, Online and Punta Cana, Dominican Republic, 2021: 11-16.

[92] ŁAŃCUCKI A. Fastpitch: parallel text-to-speech with pitch prediction[C]. 46th IEEE

International Conference on Acoustics, Speech and Signal Processing, Toronto, Canada, 2021: 6588-6592.

致谢

在即将完成学位论文之际,我想要借此机会表达我最深切的感激之情。

首先,我要感谢我的指导老师李鹏华教授,您一直是我在学术道路上最坚实的后盾。您给予我耐心指导,帮助我澄清思路,解决问题,并在研究中不断给予鼓励和支持。没有您的帮助和支持,我不可能完成这篇论文。在您的指导下,我不仅获得了专业知识和研究技能,还形成了自己的学术思想和方法论。您的教诲将对我今后的学术生涯产生深远的影响,我将终生铭记。

我也要感谢我的家人和朋友,在我整个学术过程中一直给予我鼓励和支持。没有你们的支持和理解,我不可能取得今天的成果。在我遇到困难和挫折时,你们的支持和鼓励让我能够坚持下来。

我还要感谢我的各位同门,我们一起探讨学术问题,互相帮助,共同进步。你们的意见和建议帮助我不断完善研究内容和论文质量。

最后,我要感谢我的母校和国家,给予我接受高等教育的机会和平台。在我接受教育的过程中,我获得了知识和技能,更加明确了自己的人生方向和责任使命。

在此,我再次向所有支持和帮助过我的人们表示由衷的感谢和敬意。