说话者验证中的文本适应使用说话者-文本因子分解嵌入

Yexin Yang†, Shuai Wang†, Xun Gong, Yanmin Qian, Kai Yu

MoE Key Lab of Artificial Intelligence
SpeechLab, Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai, China
{yangyexin, feixiang121976, gongxun, yanminqian, kai.yu}@sjtu.edu.cn

ABSTRACT

预收集数据（无论是训练数据还是注册数据）与实际测试数据之间的文本不匹配会严重影响基于文本的说话人验证（SV）系统的性能。虽然这个问题可以通过仔细收集目标语音内容的数据来解决，但这样的数据收集可能会非常昂贵且缺乏灵活性。在本文中，我们提出了一种新颖的文本适应框架来解决文本不匹配问题。这里提出了一个说话人文本分解网络，将输入语音分解为说话人嵌入和文本嵌入，并在后期阶段将其整合为单一表示形式。给定少量与说话人无关的适应话语，可以提取目标语音内容的文本嵌入，并用于将与文本无关的说话人嵌入适应为定制化的说话人嵌入。在 RSR2015 上的实验表明，文本适应显著提高了文本不匹配条件下的性能。

Index Terms— 说话人验证，文本依赖性，文本不匹配，适应性

1. 介绍

说话人验证旨在根据其语音验证客户所声称的身份。考虑到对语音内容的限制，说话人验证可以分为两类：基于文本的和非基于文本的。前者任务要求注册和测试话语具有相同的语音内容，而后者则没有这样的要求，为用户提供了更大的灵活性。

对于与文本无关的说话人验证任务，说话人嵌入提取器通常是在大量不受限制的语音数据上进行训练的，文本信息被隐式规范化，这是有益的，因为最终的说话人嵌入应该去除音素变异性。尽管在与文本无关的任务 [1, 2, 3, 4, 5] 上表现良好，但直接将同一模型应用于与文本相关的任务存在问题，因为在这种情况下文本信息非常重要。解决此类性能下降的常用方法是收集与评估数据具有相同语音内容的训练数据，这种方法通常被公司用于基于唤醒词的说话人验证 [6, 7, 8, 9, 10]。然而，重新收集特定应用的训练数据可能会非常昂贵且缺乏灵活性。

在实际应用中，挑战不仅来自训练数据和评估数据之间的文本不匹配，还来自评估过程中注册数据与测试数据之间的文本不匹配。例如，在实际应用中，用户通常希望使用多个关键词来唤醒智能设备。例如，谷歌设备允许使用 “OK Google” 和 “Hey Google” [10, 11]。某些应用程序甚至涉及更多的不同关键词。

在本文中，为了避免重新收集大量特定应用的训练数据，我们考虑了 “文本适应框架” 中的这两种文本不匹配情况，在该框架中，与文本无关的说话人嵌入被调整为定制的与文本相关的说话人嵌入以适应特定输入。我们提出了一种说话人-文本分解网络，它包含四个部分：一个通用特征学习器、用于提取与文本无关的 “spk” 嵌入的说话人子网、用于提取 “text” 嵌入的文本子网以及一个基于 “text” 嵌入提供的信息来学习文本适应表示的 “组合” 子网。

不同的文本不匹配类型评估集来源于 RSR2015[12] 数据集，对于传统的基于文本的任务，其中文本不匹配仅存在于训练和评估数据之间，“文本” 嵌入是从与 “spk” 嵌入相同的语音中计算得出的。当注册和测试数据之间的文本不匹配也出现时，我们从任意说话人（不同于评估集）处收集了少量语音来计算文本嵌入并适应注册说话人的嵌入。实验结果表明两种情况下性能均有显著提升。此外，从无文本适配的说话人子网中提取的 “spk” 嵌入在标准 Voxceleb 评估集中也优于原始 x 向量基线。

2. 相关工作

2.1. x-向量

x-向量 [1, 13] 是一种基于时间延迟神经网络（TDNN）的说话人嵌入学习框架。该模型包含几个帧级的时间延迟层，随后是一个统计池化层，它将帧级表示聚合为单个段级表示。可以在池化层后的段级层中加入一个或多个嵌入层以提取说话人嵌入。

2.2. 段级音素标签定义

研究人员已经探讨了将语音信息整合到说话人建模过程中的方法，其中大多数遵循帧级多任务学习范式 [14, 15]。在我们之前的工作中，我们提出了一种框架，在段级别考虑语音信息，这与段级别训练的 x 向量更为兼容。关键点是如何定义一个包含多个音素的语音片段的语音标签。我们采用了如下一种简单的方法：对于给定的包含 N 帧的片段 $\boldsymbol{x}$，对应的段级音素标签 $\boldsymbol{y}^t$ 表示为

$$\boldsymbol{y}^t = \{y_1, y_2, \ldots, y_C\}, \quad y_c = \frac{N_c}{N}$$

其中 C 是选定音素集的大小。N_c 表示 $\boldsymbol{x}$ 中第 c 个音素的出现次数。

3. 文本适应框架

一个典型的深度说话人验证任务包括三个阶段：

- 训练：说话人嵌入提取器使用大量预先收集的数据进行训练。
- 评估
  - 注册：新说话人通过使用训练良好的提取器生成说话人嵌入来进行注册。

†: These authors have contributed equally to this work

Yanmin Qian and Kai Yu are the corresponding authors

Authors would like to thank Johan Rohdin for providing phoneme labels

This work has been supported by the China NSFC project No. U1736202. Experiments have been carried out on the PI supercomputer at Shanghai Jiao Tong University

– 测试：每个测试话语都使用所声称身份的注册模型进行评估以做出验证决策。

对于与文本无关的任务，我们不对训练和评估数据之间或注册和测试数据之间的文本匹配提出任何要求。

对于传统的依赖文本的任务，直接应用训练用于非依赖文本说话人验证任务的系统通常会由于训练数据和评估数据之间的文本不匹配而获得非常差的表现。当前最先进的依赖文本的说话人验证系统与非依赖文本的方法相同，但训练数据是为特定应用程序收集的。例如，[6, 7] 和 [8] 中的训练语音片段是从大量共享相同内容 “OK Google” 或 “Hey Cortana” 的说话人中收集的。尽管使用这种方法取得了良好的结果，但对于每个不同的短语重新收集训练数据是昂贵且不灵活的。

此外，现实世界的应用通常不遵循标准的文本依赖制度。还存在注册语音和测试语音之间也存在文本不匹配的情况。例如，在某些条件下，用户可能会希望使用多个关键词来唤醒智能设备。例如，谷歌设备同时支持 “OK Google” 和 “Hey Google” [10, 11]。一些应用程序可能需要更多的不同关键词。是否允许用户只注册其中一个并在其他不同的关键词上进行测试?

3.1. 说话人嵌入的文本适应

总结上述提到的问题，我们希望解决以下两个文本不匹配条件：

- 训练和评估数据之间的文本不匹配存在于传统的基于文本的任务中。
- 文本不匹配不仅存在于训练和评估数据之间，还存在于注册和测试数据之间。

在预收集的训练数据与评估数据共享相同文本的情况下，文本信息被隐式地建模在说话人提取器中。然而，为了处理上述两种类型的文本不匹配问题而无需回忆大量的训练数据，应显式考虑文本信息建模。本文提出了一种框架，在该框架中可以将与文本无关的说话人嵌入适应为与文本相关的说话人嵌入，同时可以根据输入自定义文本信息。

3.2. 说话人-文本因子分解网络

如图 1 所示，所提出的模型包含四个部分：通用特征提取器 M_f、两个并行子网 M_s 和 M_t 分别用于说话人区分和音素分布学习，以及 “组合” 子网 M_c 用于整合说话人和语音信息。根据 [2]，音素分类器 M_t 预测输入段落中音素的归一化类别出现次数，而说话人分类器 M_s 是一个标准分类器，用于预测说话人类别。说话人嵌入 ebd_s 和基于音素的文本嵌入 ebd_t 从 M_s 和 M_t 中提取后，作为输入拼接进入组合网络 M_c，旨在恢复说话人的身份和语音信息。模型是联合训练的。给定一个训练段对 $[\boldsymbol{x}_s, \boldsymbol{x}_t]$ 及其对应的 speaker 标签 $\boldsymbol{y}^s$ 和音素标签 $\boldsymbol{y}^t$，损失被定义为 $\mathcal{L}_{total} = \mathcal{L}_{s1} + \mathcal{L}_{t1} + \mathcal{L}_{s2} + \mathcal{L}_{t2}$，其中

$$\mathcal{L}_{s1} = \text{CE}(M_s(M_f(\boldsymbol{x}_s)), \boldsymbol{y}^s)$$
$$\mathcal{L}_{t1} = \text{KLD}(M_t(M_f(\boldsymbol{x}_t)), \boldsymbol{y}^t)$$
$$\mathcal{L}_{s2} = \text{CE}(M_c([\text{ebd}_s, \text{ebd}_t]), \boldsymbol{y}^s)$$
$$\mathcal{L}_{t2} = \text{KLD}(M_c([\text{ebd}_s, \text{ebd}_t]), \boldsymbol{y}^t)$$

“spk” 嵌入 ebd_s 通过 speaker 子网从 $\boldsymbol{x}_s$ 计算得到，而 “text” 嵌入 ebd_t 则通过 text 子网从 $\boldsymbol{x}_t$ 得到。为了更好地分离说话人和语音信息，训练对 $[\boldsymbol{x}_s, \boldsymbol{x}_t]$ 是从训练数据中随机采样的，可以是相同的发音或是来自两个不同说话人的两次发音。

从说话人子网络中提取的 “spk” 嵌入直接用于与文本无关的任务，因为不需要进行文本适应。对于传统的与文本相关的说话人验证任务，在注册和测试数据中可以准确计算出文本嵌入的情况下，“text” 嵌入来自与 “spk” 嵌入相同的语音片段。对于另一种存在注册和测试语音片段之间文本不匹配的目标试验场景，我们使用少量预先收集的数据（例如，其他说话人的 10 个语音片段）来计算 “text” 嵌入，并用其适应从注册语音片段中计算出的 “spk” 嵌入，而真实的 “text” 嵌入则用于测试语音片段。

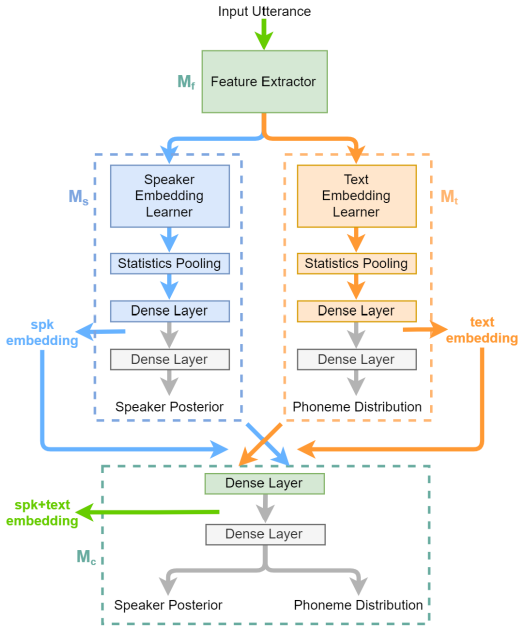

Fig. 1. 提出的说话人-文本因子分解网络。

4. 实验设置

4.1. 数据

我们在实验中使用了 Voxceleb 和 RSR2015 数据集。所有的音素标签都是由一个音素识别器生成的。更多细节可参考 [2]。关于文本适应性和文本独立性评估的数据准备详情将在下文给出，试验定义可在网络上获取以重现我们在定制的 RSR2015 评估集上的结果 [1]。

4.1.1. 训练集

在我们的实验中，使用 Voxceleb2 开发集来训练神经网络和概率线性判别分析（PLDA）后端。该集合包含 5994 位说话者和 1092009 个 utterances。为了训练神经网络，我们遵循 Kaldi Voxceleb 配方中的数据准备过程，将 utterances 切割成长度在 2 秒到 4 秒之间的片段。需要注意的是，与该配方不同，我们没有使用任何数据增强。

4.1.2. 文本适应性评估集

两个独立的评估集被创建对应于不同的文本不匹配情况：**训练数据与评估数据不匹配**：当训练和评估数据之间的不匹配仅存在时，它就成了传统的依赖文本的任务。评估集来源于 RSR2015[12] 第一部分的评估部分。包含来自 106 名说话者（57 名男性，49 名女性）的 30 个固定短语发音，每个短语持续时间为 3-4 秒。每位说话者每句话念 9 次，其中 3 次用于注册，其余用于测试。如表 1 所示，在一个标准的依赖文本任务中，有四种可能的试验类型，其中 TC 代表目标试验，TW、IC、IW 表示三种非目标条件。

[1] https://github.com/Xflick/RSR2015_trials

Table 1. 文本依赖任务的试验类型

	Correct Content	Wrong Content
Target	TAR-校正 (TC)	TAR-错误 (TW)
Impostor	_IMP-正确_ (IC)	错误-IMP (IW)

由于在原始试验定义中，大约 90% 的原始试验属于非常简单的 IW 情况，我们按照以下比例生成自己的试验列表：TC:TW:IC:IW=1:3:3:3。

招生数据与测试数据不匹配：如前所述，RSR2015 第一部分评估集中有 30 个不同的固定短语。我们随机选择了其中的十个，并生成了十个评估子集。考虑了两种注册条件，文本无关和文本相关。对于前者条件，三个注册语音的文字是随机选择的，而对于后者，文字在注册语音之间共享且与测试语音的文字没有重叠。用于适应的文本嵌入通过使用与目标文本相同的开发集中的 10 个随机语音计算得出（说话者来自评估集之外）。为了更好地展示我们说话人-文本因子化网络的文本意识，增加了 TC 试验的数量。对于这个特定任务的试验列表遵循以下比例：TC:TW:IC:IW=1:1:1:1。

4.1.3. 独立于文本的评估集

Voxceleb1 **评估集**：为了验证我们基线系统和提议系统的性能，我们首先报告了在官方 Voxceleb1 评估集上的结果。使用了清理过的测试列表：VoxCeleb1（表示为 Voxceleb1-O，O 代表“原始”），Voxceleb1-E（扩展版）以及清理后的 Voxceleb1-H（困难版本）。

RSR2015 **评估集**：为了更好地展示我们提出框架的有效性，我们基于 RSR2015 设计了一个额外的文本无关评估集，所有试验配对与第 4.1.2 节中传统文本相关情况定义的一样，而来自 TW 条件下的试验现在被视为目标。

4.2. 系统配置

标准的 x-vector[1] 系统，带有五层时间延迟层和两层密集层，被用作我们的基线系统。提出的因子化网络系统是基于基线系统修改而来的，其中三层时间延迟层用于提取通用特征。M_s 和 M_t 具有相同的结构，两者都包含两层时间延迟层、一层统计池化层和两层密集层，区别在于一个用于说话人分类，另一个则用于预测语音信息。M_c 拥有两个通用的密集层以及两个输出层来处理两个任务。值得注意的是，在仅提取说话人嵌入时，因子化网络与基线 TDNN 模型具有完全相同的结构。

40 维的 Fbank 特征用于模型训练。神经网络在 4 个 GPU 上使用批量大小为 256 进行训练。使用学习率为 0.01、动量为 0.9 和权重衰减为 1e-4 的随机梯度下降来优化模型。ReLU 激活函数之后应用了批归一化。

PLDA 在 Voxceleb1 测试集上应用以验证我们系统的正确性，对于其他测试集，则消除 PLDA 补偿的影响并专注于学习到的嵌入特性，并使用简单的余弦评分。

所有架构均在 PyTorch[16] 中实现。我们根据等错误率（EER）和最小检测成本（minDCF）报告模型性能，其中将 P_{tar} 设置为 0.01。

5. 结果与分析

5.1. 文本独立任务的验证结果

x-向量被提出用于处理与文本无关的任务，以展示基准模型和所提模型的正确性。我们首先在标准 Voxceleb1 评估集上报告了结果，如表 2 所示。由于我们仅使用了干净的训练数据，因此该基准模型相比文献中的其他模型较为强大，如 [17, 18, 19] 所述。如表 2 所示，从所提模型的说话人子网中提取的嵌入将 Voxceleb_O、Voxceleb_E 和 Voxceleb_H 上的 EER 分别从 2.888%、3.055%和 5.026%降低到了 2.595%、2.784%和 4.703%。在 minDCF 方面也可以观察到类似的改进。对于 RSR2015 与文本无关评估集上的实验，也观察到了说话人嵌入从说话人子网中提取出的类似性能提升。

5.2. 文本适应以解决文本不匹配问题

5.2.1. 训练数据与评估数据不匹配

如表 4 所示，当训练和评估数据之间存在不匹配时，将文本信息整合到嵌入中显著提高了性能。系统的 EER 从 6.671% 降低到了 1.542%，而 minDCF 则从 0.5234 降到了 0.1246。

表 5 显示了当单独分析不同类型的错误时的结果。由错误文本导致的错误 TW 和 IW 按预期大大减少。说话人错误（IC）也从 1.919% 减少到 1.101%。

5.2.2. 报名数据与测试数据不匹配

如表 6 所示，当训练数据和测试数据之间、以及注册数据和测试数据之间发生不匹配时，大多数系统在任务上都会失败。然而，通过使用目标文本嵌入代替真实的文本嵌入来调整注册“spk”嵌入，系统性能得到了实质性的提升，这表明了我们分解网络生成文本定制的说话人嵌入的有效性。

6. 结论与未来工作

训练数据和评估数据之间的文本不匹配会导致基于文本的说话人验证性能大幅下降。一个常见的解决方案是收集与评估数据具有相同文本信息的应用特定训练数据。为了摆脱昂贵且缺乏灵活性的数据收集过程，并利用大量不受限制的语音数据，我们提出了一种“文本适应说话人验证”框架，在该框架中，可以根据特定的适应输入将文本无关的说话人嵌入适应为文本定制的嵌入。提出了一个说话人-文本分解网络，它首先将语音段分解成与文本无关的说话人嵌入和与说话人无关的文本嵌入，然后重新组合它们作为一个包含两种信息的单一嵌入。我们首先在标准文本无关 Voxceleb 评估集上验证了没有文本适应的方法，并观察到所有三个试验列表上的性能一致提升。基于 RSR2015 数据集得到的三个定制化评估集的结果显示，使用文本适应的方法可以大大减少训练和评估数据之间以及注册和测试数据之间的文本不匹配造成的错误。

在未来的工作中，我们将更加努力地让模型能够利用简单的纯文本而不是从特定音频计算出的文本嵌入来进行说话人嵌入的文本适应。

7. REFERENCES

[1] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur, “X-vectors: Robust dnn embeddings for speaker recognition,” in IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018. IEEE, 2018.

[2] Shuai Wang, Johan Rohdin, Lukáš Burget, Oldřich Plchot, Yanmin Qian, Kai Yu, and Jan Černockỳ, “On the usage of phonetic information for text-independent speaker embedding extraction,” Proc. Interspeech 2019, pp. 1148–1152, 2019.

[3] Zili Huang, Shuai Wang, and Kai Yu, “Angular softmax for short-duration text-independent speaker verification.,” in Interspeech, 2018, pp. 3623–3627.

Table 2. 在 Voxceleb 1 评估集上的验证实验

系统配置		Voxceleb1_O		Voxceleb1_E		Voxceleb1_H	
Architecture	Embedding Type	EER	minDCF	EER	minDCF	EER	minDCF
TDNN	spk	2.888	0.3281	3.055	0.3272	5.026	0.4646
Factorization Net	spk	2.595	0.2940	2.784	0.2990	4.703	0.4292

Table 3. 在 RSR2015 文本无关评估集上的实验

系统配置		EER（%）	最小化代价函数
Architecture	Embedding Type		
TDNN	spk	7.220	0.7068
Factorization Net	spk	6.239	0.6721

Table 4. 在 RSR2015 文本适应评估集上的实验（训练数据与评估数据不匹配）

系统配置		EER (%)	最小化代价函数
Architecture	Embedding Type		
TDNN	spk	6.671	0.5234
因子分解网络	spk	6.010	0.5144
	spk+text	1.542	0.1246

Table 5. 不同错误类型在 RSR2015 文本适应评估集上的 EER（训练数据与评估数据不匹配）

系统配置		EER (%)		
Architecture	Embedding Type	TW	IC	IW
TDNN	spk	10.60	1.919	1.007
因子分解网络	spk	10.32	1.385	0.7867
	spk+text	2.454	1.101	0.1573

Table 6. 与文本无关/与文本相关的注册（在第 4.1.2 节中介绍）EERs(%) 在 RSR2015 文本适应评估集上的结果（注册数据和测试数据之间的不匹配）

子集	TDNN	因子分解网络		
	spk	spk	发言人 +文本	说话人 +适应文本
1	27.34/29.21	27.05/28.31	26.42/26.29	12.26/15.07
2	24.61/22.73	26.03/24.09	24.87/23.18	13.71/13.99
3	22.18/24.33	23.41/25.37	20.75/21.49	10.84/14.49
4	22.56/20.19	22.88/19.97	24.87/17.14	6.912/7.227
5	28.16/32.45	26.74/30.99	27.93/34.84	10.51/16.98
6	22.46/21.32	22.76/21.74	21.38/23.40	9.985/10.81
7	22.20/25.09	22.02/25.23	22.02/27.67	7.898/12.52
8	29.97/29.09	29.64/28.88	30.24/30.44	14.21/15.32
9	22.86/24.73	22.43/25.04	22.91/24.66	9.450/14.56
10	22.91/25.50	22.51/25.34	23.15/25.76	9.083/12.35
avg	24.53/25.46	24.55/25.50	24.45/25.49	10.49/13.33

[4] Zili Huang, Shuai Wang, and Yanmin Qian, "Joint i-vector with end-to-end system for short duration text-independent speaker verification," in 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018, pp. 4869–4873.

[5] Chunlei Zhang and Kazuhito Koishida, "End-to-end text-independent speaker verification with triplet loss on short utterances.," in Interspeech, 2017, pp. 1487–1491.

[6] Ehsan Variani, Xin Lei, Erik McDermott, Ignacio Lopez Moreno, and Javier Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2014, pp. 4052–4056.

[7] Georg Heigold, Ignacio Moreno, Samy Bengio, and Noam Shazeer, "End-to-end text-dependent speaker verification," in 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2016, pp. 5115–5119.

[8] Shi-Xiong Zhang, Zhuo Chen, Yong Zhao, Jinyu Li, and Yifan Gong, "End-to-end attention based text-dependent speaker verification," in 2016 IEEE Spoken Language Technology Workshop (SLT). IEEE, 2016, pp. 171–178.

[9] Yichi Zhang, Meng Yu, Na Li, Chengzhu Yu, Jia Cui, and Dong Yu, "Seq2seq attentional siamese neural networks for text-dependent speaker verification," in ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019, pp. 6131–6135.

[10] FA Rezaur rahman Chowdhury, Quan Wang, Ignacio Lopez Moreno, and Li Wan, "Attention-based models for text-dependent speaker verification," in 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018, pp. 5359–5363.

[11] Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno, "Generalized end-to-end loss for speaker verification," in 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018, pp. 4879–4883.

[12] Anthony Larcher, Kong Aik Lee, Bin Ma, and Haizhou Li, "Text-dependent speaker verification: Classifiers, databases and rsr2015," Speech Communication, vol. 60, pp. 56–77, 2014.

[13] David Snyder, Daniel Garcia-Romero, Daniel Povey, and Sanjeev Khudanpur, "Deep neural network embeddings for text-independent speaker verification," Proc. Interspeech 2017, pp. 999–1003, 2017.

[14] Yuan Liu, Yanmin Qian, Nanxin Chen, Tianfan Fu, Ya Zhang, and Kai Yu, "Deep feature for text-dependent speaker verification," Speech Communication, vol. 73, pp. 1–13, 2015.

[15] Yi Liu, Liang He, Jia Liu, and Michael T. Johnson, "Speaker embedding extraction with phonetic information," CoRR, vol. abs/1804.04862, 2018.

[16] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer, "Automatic differentiation in pytorch," 2017.

[17] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman, "Voxceleb2: Deep speaker recognition," arXiv preprint arXiv:1806.05622, 2018.

[18] Xu Xiang, Shuai Wang, Houjun Huang, Yanmin Qian, and Kai Yu, "Margin matters: Towards more discriminative deep neural network embeddings for speaker recognition," arXiv preprint arXiv:1906.07317, 2019.

[19] Weidi Xie, Arsha Nagrani, Joon Son Chung, and Andrew Zisserman, "Utterance-level aggregation for speaker recognition in the wild," in ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019, pp. 5791–5795.