

数智创新 声至未来

DEEP IN DIALECTS, FOR FUTURE WAVE

第八届信也科技杯算法大赛

THE 8TH FINVOLUTION DATA SCIENCE COMPETITION

鲜衣怒马少年时

赵国梁（西安交通大学）



目录

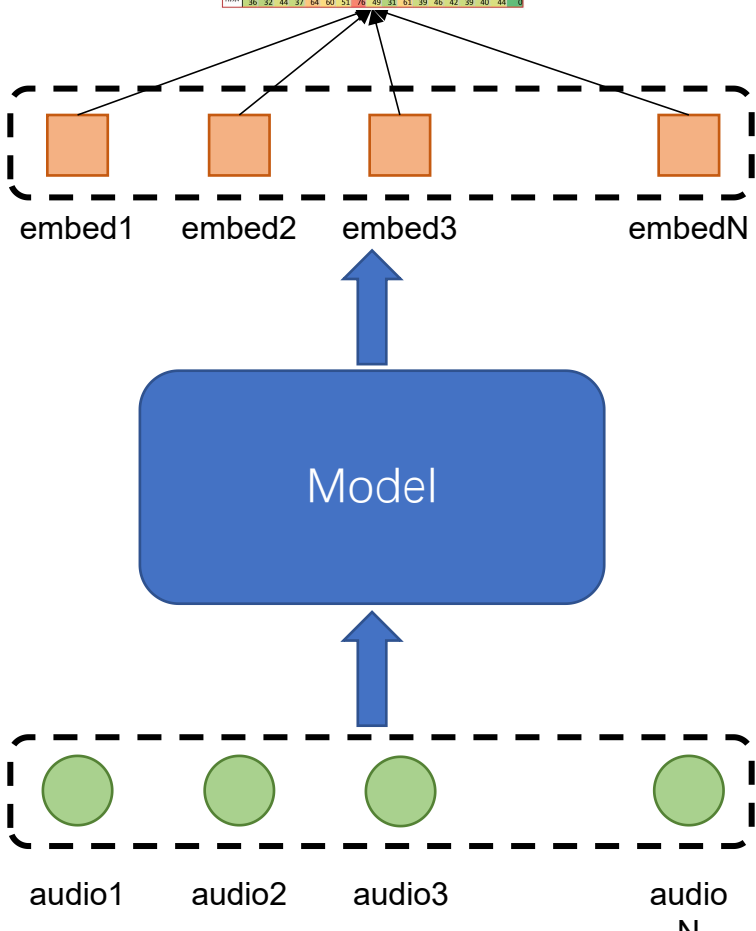
1. 团队介绍
2. 赛题理解
3. 算法设计
4. 实验结果
5. 总结思考

CONTENTS



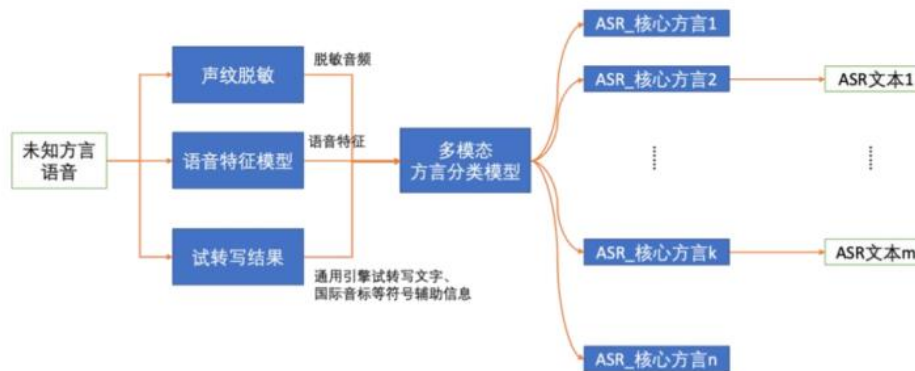
赛题理解

	北	成	潮	广	上	厦	长	贵	湘	晋	川	天	津	石	陕	新	藏
北京	32	23	34	69	68	57	78	52	68	53	30	15	16	19	36		
成都	32	38	25	66	61	50	77	44	32	62	33	39	38	34	28	32	
潮州	23	38	0	40	69	71	62	79	54	41	70	32	32	32	22	23	44
武汉	34	25	40	0	66	63	54	77	34	31	63	34	40	41	36	37	41
广州	69	66	69	66	0	67	68	71	68	69	64	67	70	73	68	69	64
上海	68	61	71	63	67	0	41	78	64	66	24	66	69	67	67	68	70
杭州	57	50	62	54	68	41	0	76	57	57	42	57	61	60	57	59	61
厦门	79	77	79	77	71	78	76	0	77	79	76	79	80	81	79	79	76
长沙	52	44	56	34	68	64	57	77	0	47	64	51	53	52	52	53	55
福州	32	32	41	31	69	66	57	79	47	0	66	39	46	40	37	41	31
贵阳	68	62	70	63	64	24	42	76	64	66	0	68	70	68	68	69	69
西安	15	33	32	34	67	66	57	79	51	39	68	0	35	33	24	26	39
西宁	33	39	32	40	70	69	61	60	53	46	70	15	0	40	33	29	36
兰州	30	38	37	41	73	67	60	81	52	40	68	33	40	0	32	32	37
天津	15	33	32	36	68	67	57	79	52	40	68	11	33	32	0	25	27
济南	16	34	22	37	69	68	59	79	53	37	69	24	29	32	25	0	19
石家庄	18	38	23	41	69	70	61	79	15	41	69	26	36	37	27	18	0
南京	36	32	44	37	64	60	51	76	49	31	63	39	46	42	39	40	41



背景

对于国内仍广泛使用、大量存在且种类繁多的方言语音，商业解决方案还不能满足大部分方言的转写。如果考虑对所有方言单独建ASR模型，其成本是不可接受的；一个相对可行的方案：设置一系列核心方言并建ASR模型，对未知的方言进行鉴别，确定距离其最近的 $m(m \geq 1)$ 种核心方言，再尝试用这 m 种ASR引擎转写该未知方言，转写的不完美结果可用于支持下游任务：



因此，度量不同方言之间的距离是问题的关键，有助于进一步探索如何从语音层面建模方言、进行方言特征抽取、分析方言形成和演化的机理，其结果也可以和传统方言分类方法做合理性的相互印证，以及服务于更为广泛的研究目标。

数据

本赛题训练数据（复赛）为18种方言的语音片段，其中前九种方言是10小时，后九种为1小时，测试仅包含了“集内-集外”与“集外-集外”两类

表：复赛测试数据集构成

样本对	file_2	训练语音集	测试语音集
file_1		方言 1-18	方言 19-...
测试语音集	方言 19-...	集内-集外	集外-集外

评价指标

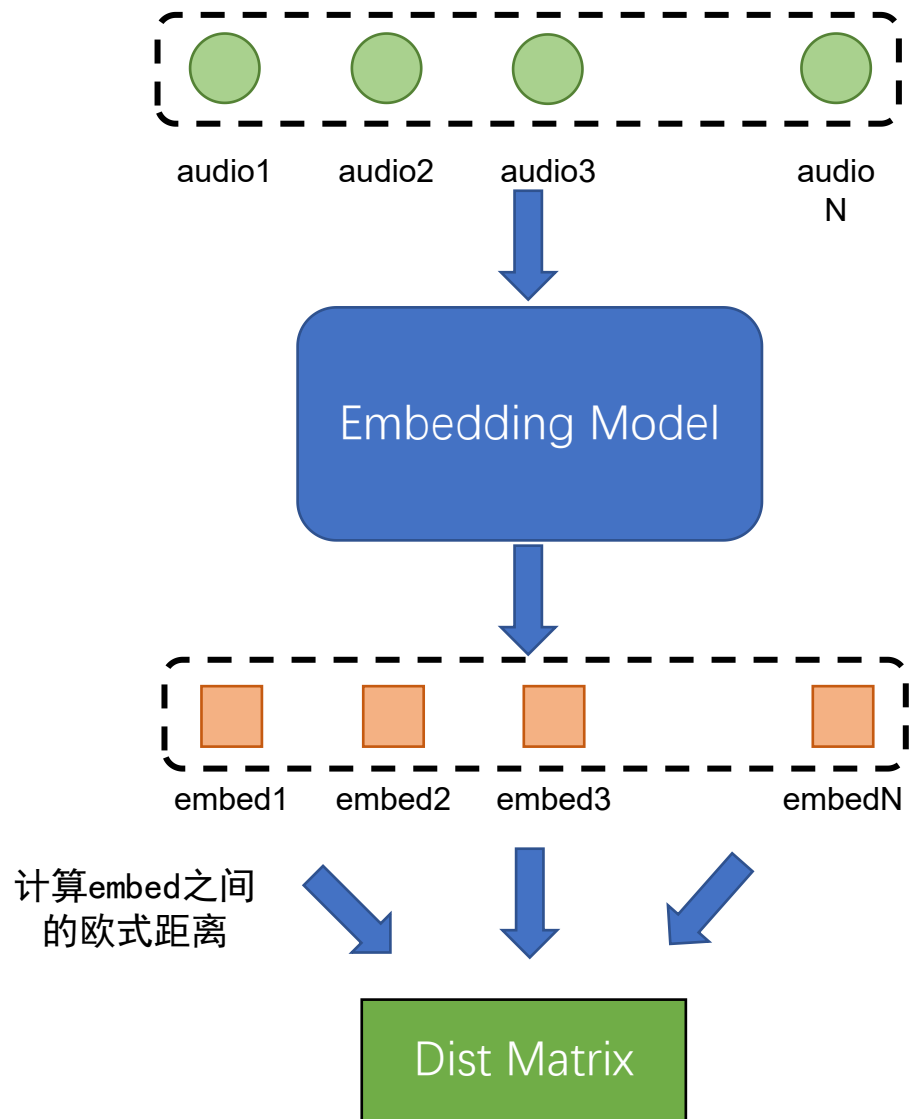
假定用于预测任务的样本对数为 Q 。计算真实样本对距离 \hat{T}_k 与预测样本对距离 T_k 的L1距离作为评价指标：

$$t = |\hat{T} - T|_{l1} = \frac{1}{Q} \sum_{k=1}^Q |\hat{D}(\text{audio}_{k,1}, \text{audio}_{k,2}) - D(\text{audio}_{k,1}, \text{audio}_{k,2})|$$

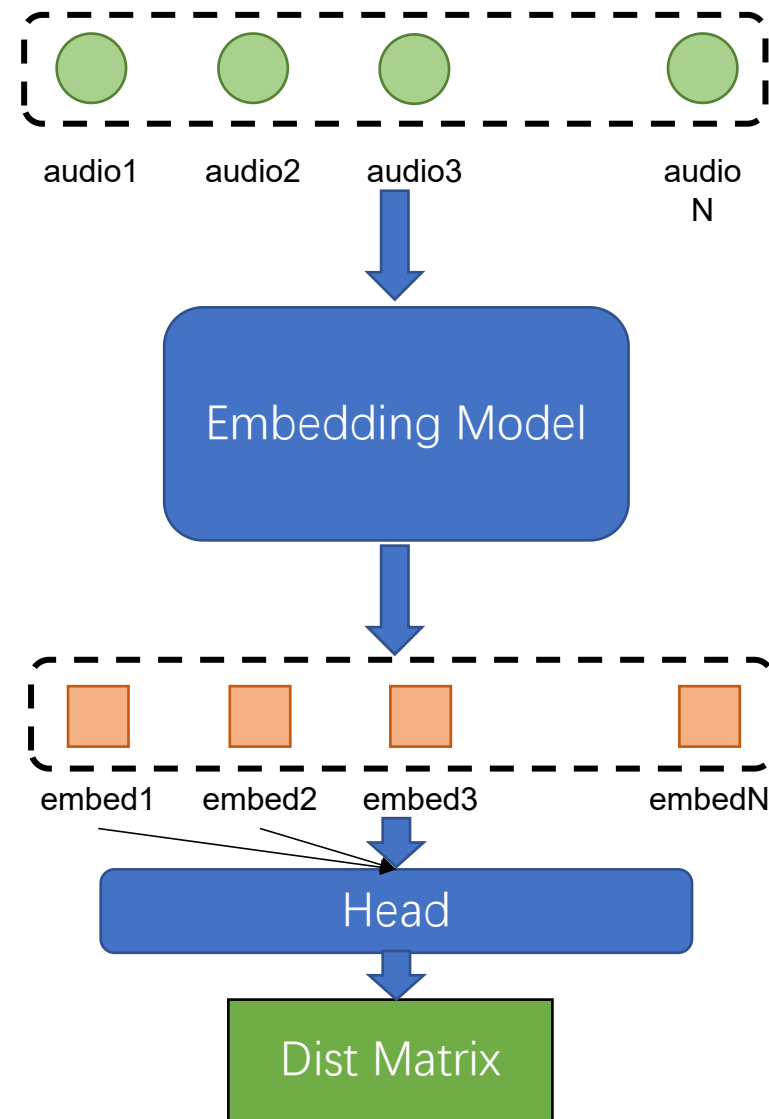
赛题难点

- 如何根据已知的方言数据集训练出一个可以同样适用于“集内-集外”与“集外-集外”的测试数据的模型
- 数据样本不平衡的问题
- 训练样本中distance为0的样本过多的问题

算法设计——模型结构

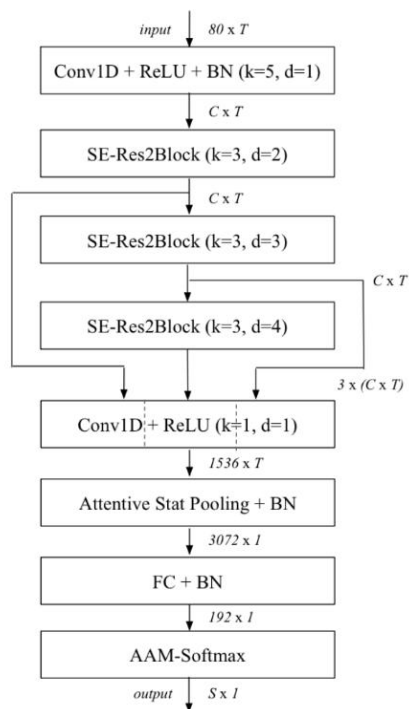


考虑到embed的维度过高，直接用欧式距离作为语音之间的距离，缺乏一定泛化能力，因此考虑设计对应Head网络预测语音之间的距离

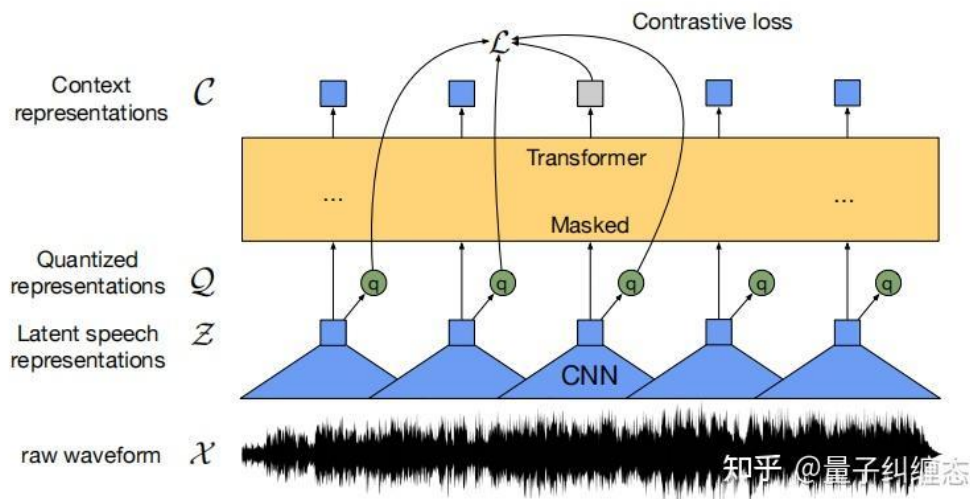


算法设计——模型结构

Embedding Model

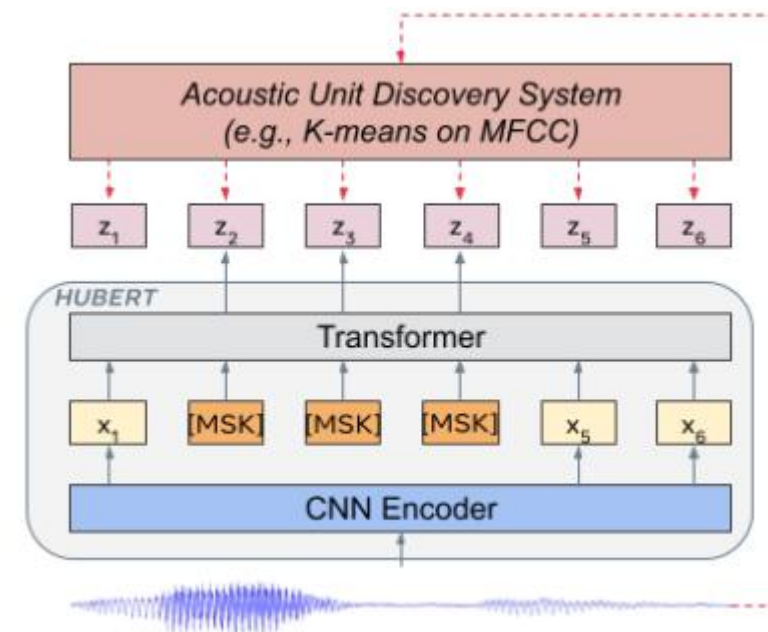


ECAPA-TDNN



Wav2Vec2.0

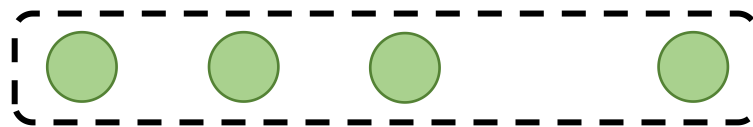
★ TencentGameMate/chinese-hubert-large



Hubert

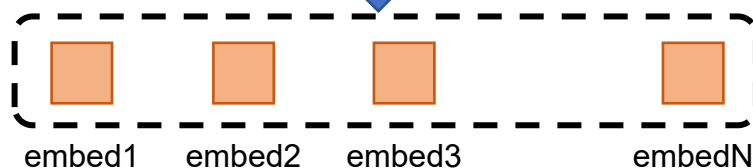
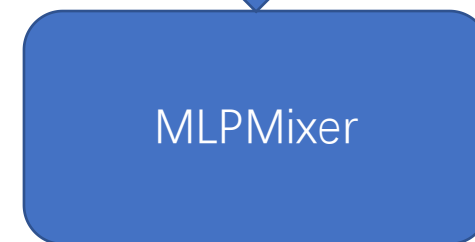
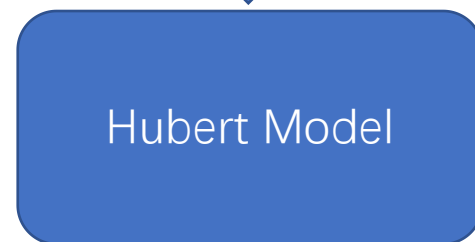
算法设计——模型结构

加入Mel Data



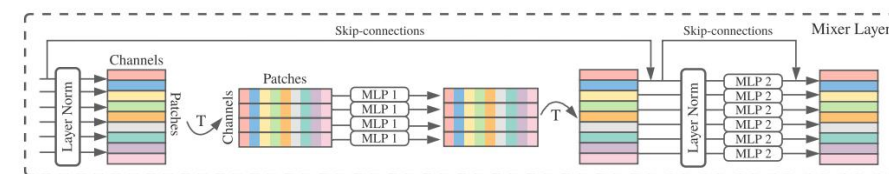
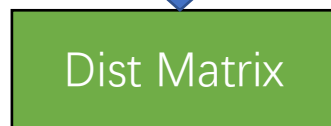
audio1 audio2 audio3 audioN

MelSpectrogram



embed1 embed2 embed3 embedN

- SENet
- Linear
- Sigmoid



算法设计——数据增强

① 固定训练和测试样本的时长（400x80）

固定训练和测试样本长度为指定值，减小样本长度不同带来的推理误差，便于设计模型（MLPMixer、Transformer等）处理序列数据。

② 修改baseline示例中的同类别不同样本的concatenate

由于baseline中，仅仅在训练中设计了同类别不同样本的拼接，然而推理无法进行同类别不同样本的拼接。因此，改为同一条样本重复拼接，直到指定长度400x80。

③ 设计Mixup策略，针对同类别不同样本进行Mixup

设定概率 $p = \text{random.random}()$ ，当 $p > 0.5$ 时，进行mixup操作：

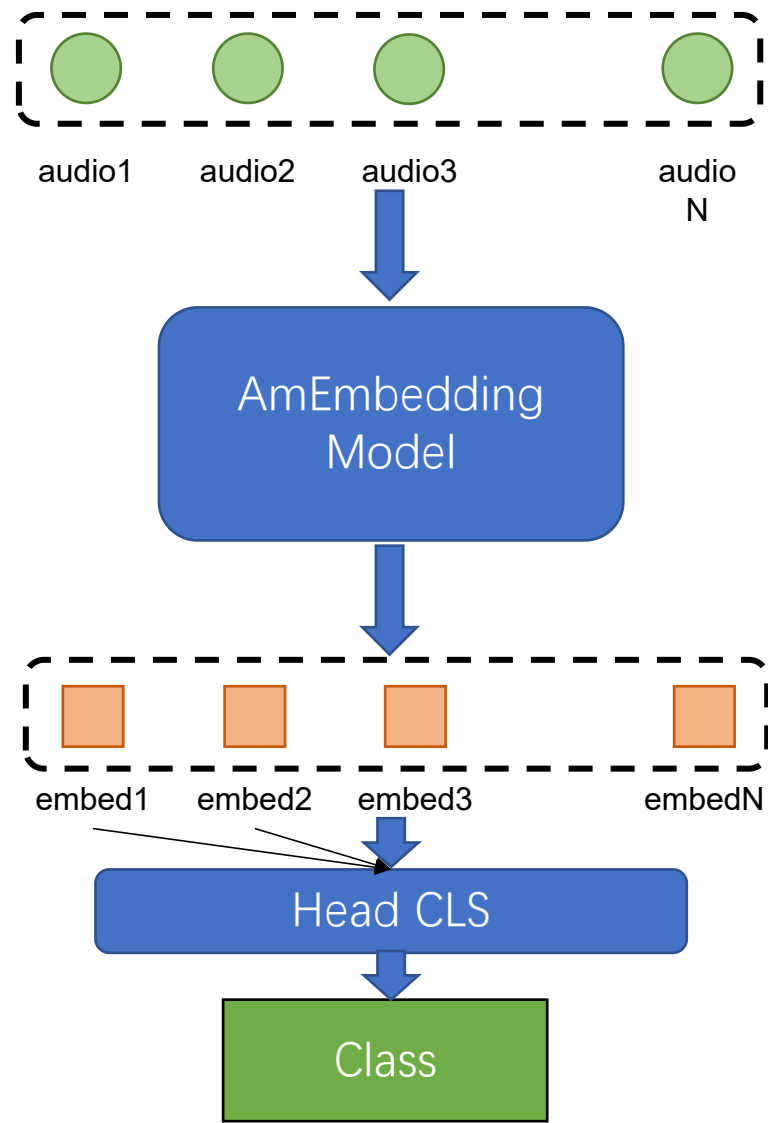
$$audio_{mixup} = \alpha * audio_i + (1 - \alpha) * audio_j$$

④ 进行音量和音速的随机扰动

利用torchaudio.transforms中自带的包对原音频进行0.8-1.2的音速和音量扰动

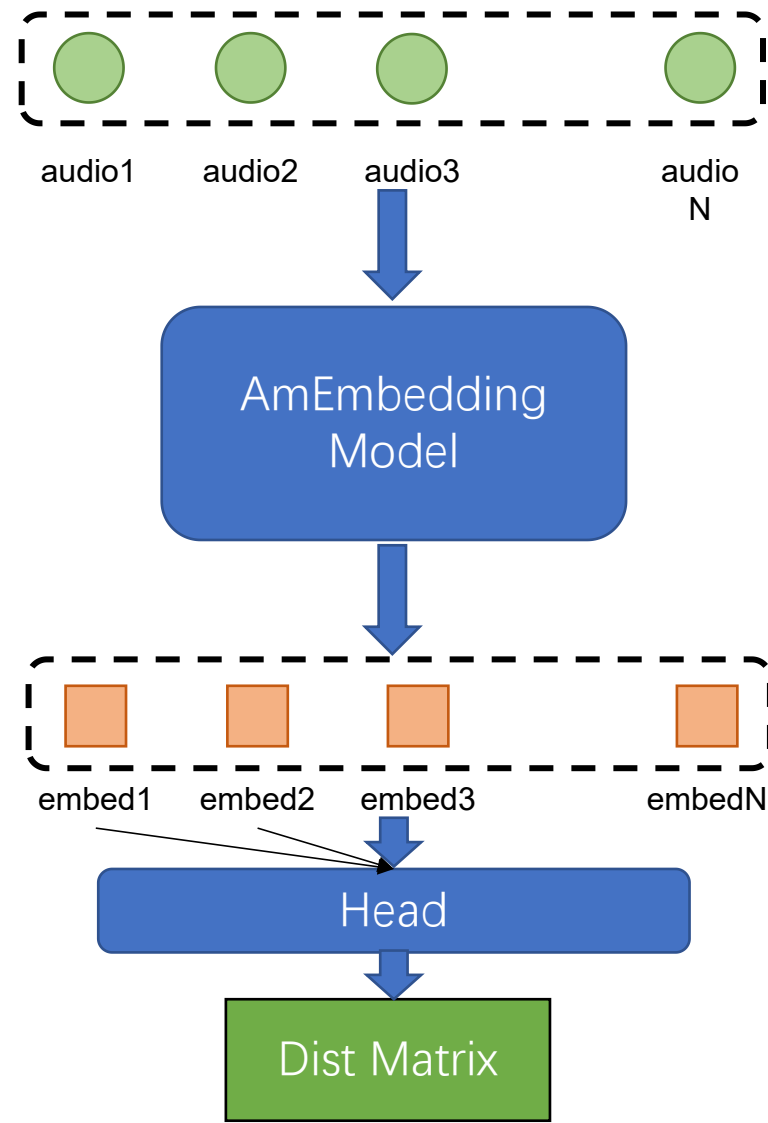
算法设计——训练策略

Pretrain与Finetune



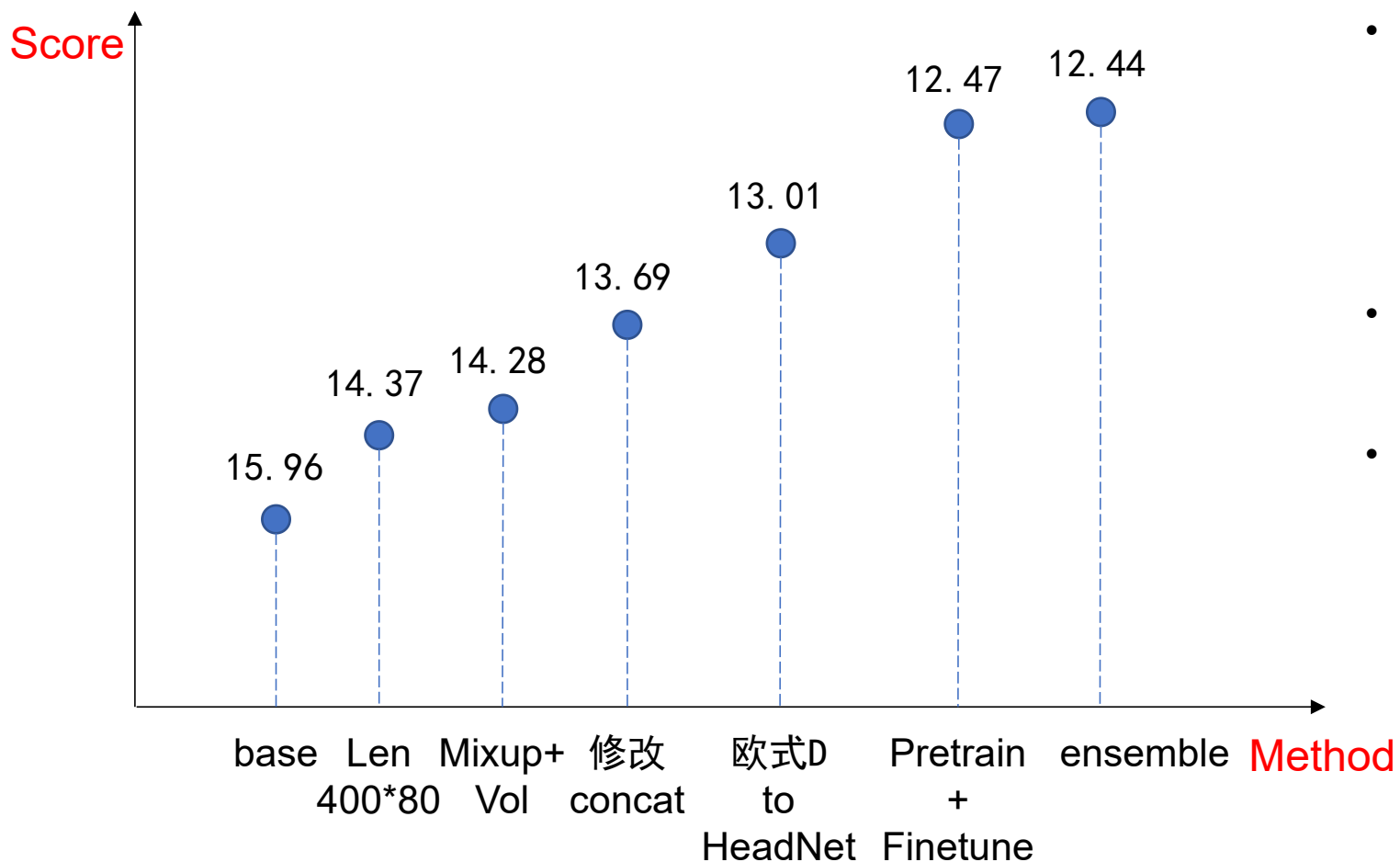
迁移

Delete



实验结果与总结

实验结果



总结和思考

- 对于复赛数据的使用可能存在一定问题，通过实验结果来看，仅仅使用初赛数据的效果略好于初赛数据+复赛数据的效果。可能是数据样本失衡的问题
- 数据loader的方式导致一个batch内distance为0的样本居多，影响模型学习
- 由于赛程问题，赛程结束的前两天才想到Pretrain+Finetune的架构，导致对该架构的调试时间不足，后续可以在这方面继续深挖

**THANK
YOU**

