信也科技杯 FINVOLUTION BIG DATA COMPETITION | 信也 FINVOLUTION

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

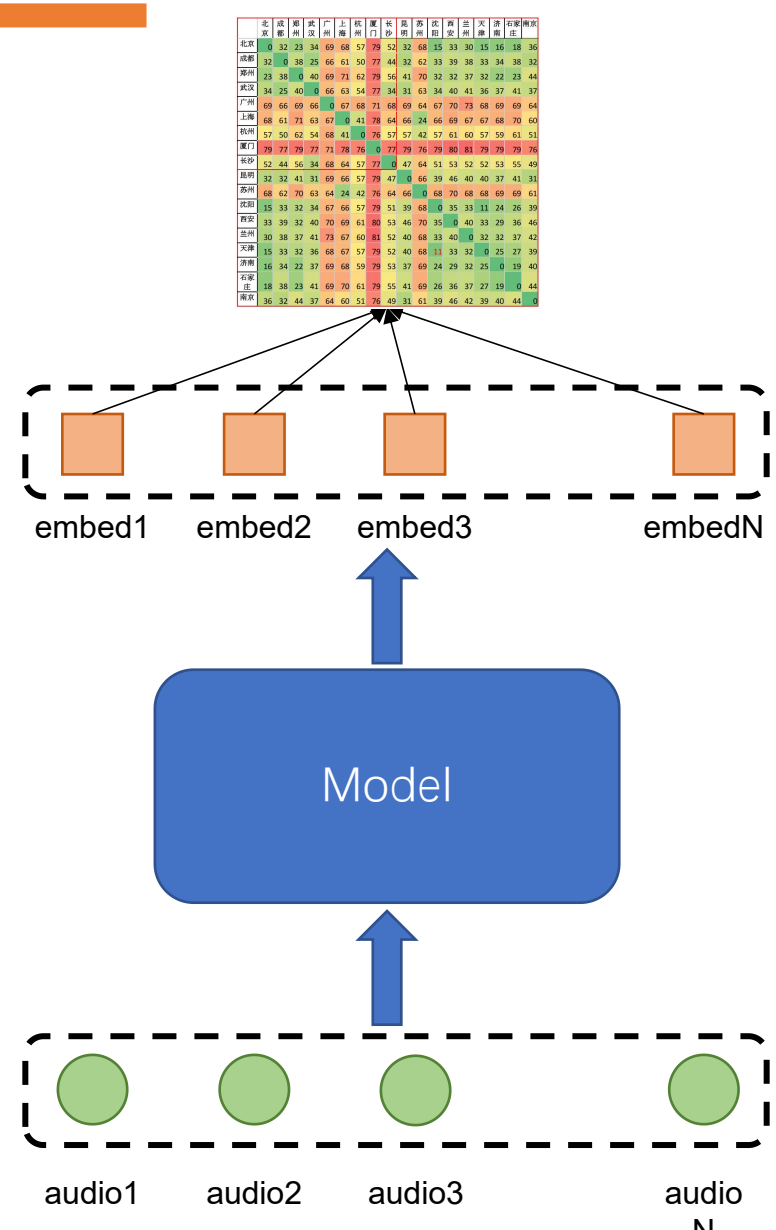
鲜衣怒马少年时

赵国梁（西安交通大学）

目录

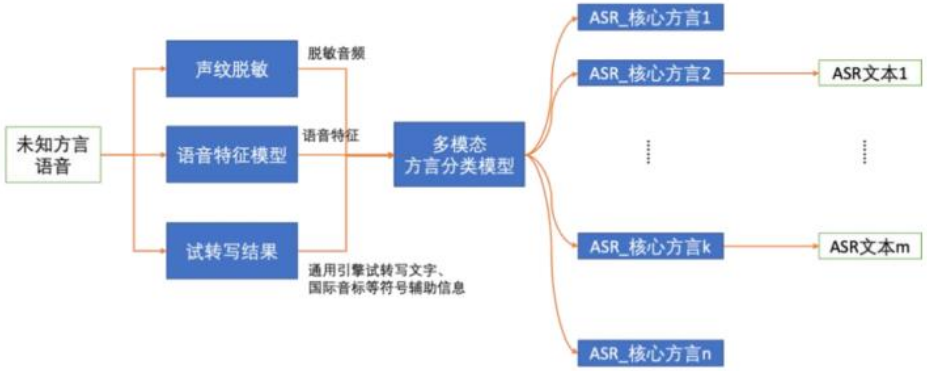
CONTENTS

赛题理解



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>=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。

赛题理解

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

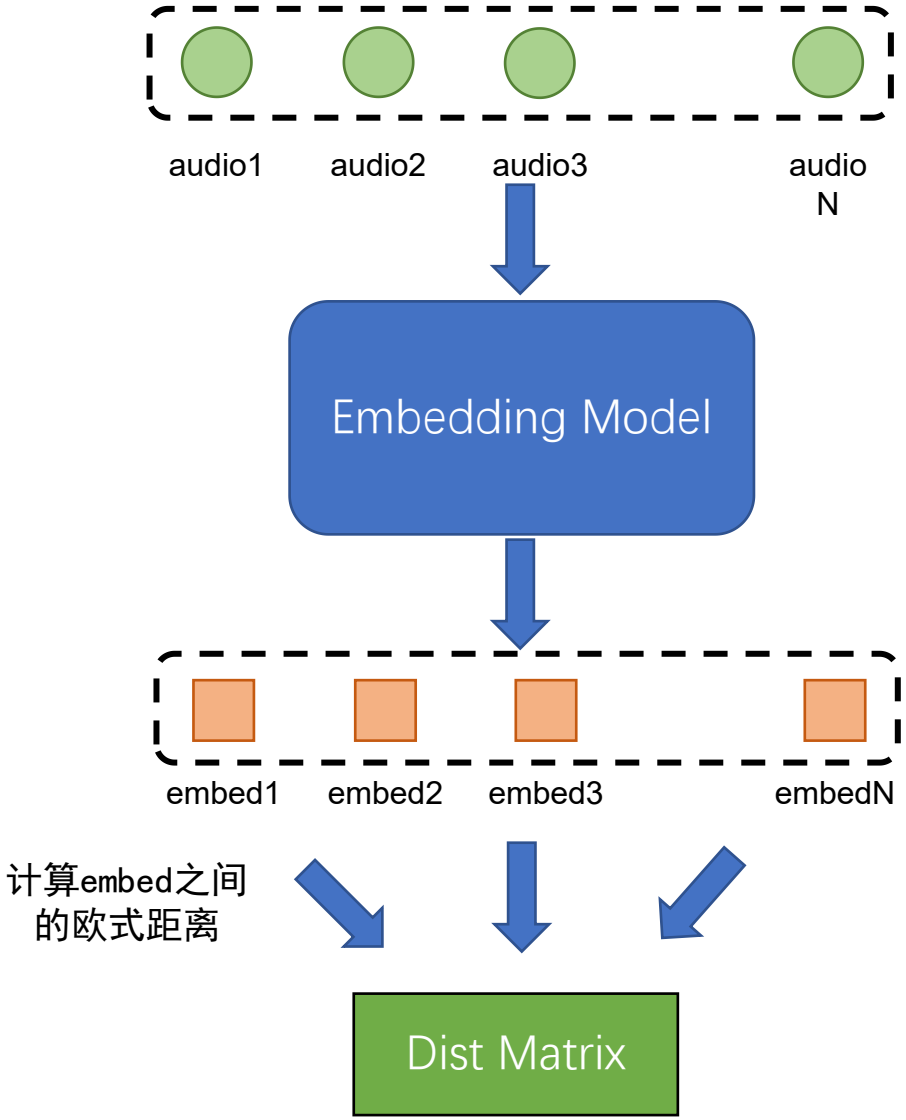
假定用于预测任务的样本对数为Q。计算真实样本对距离$\hat{T}_k$与预测样本对距离T_k的L1距离作为评价指标：

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

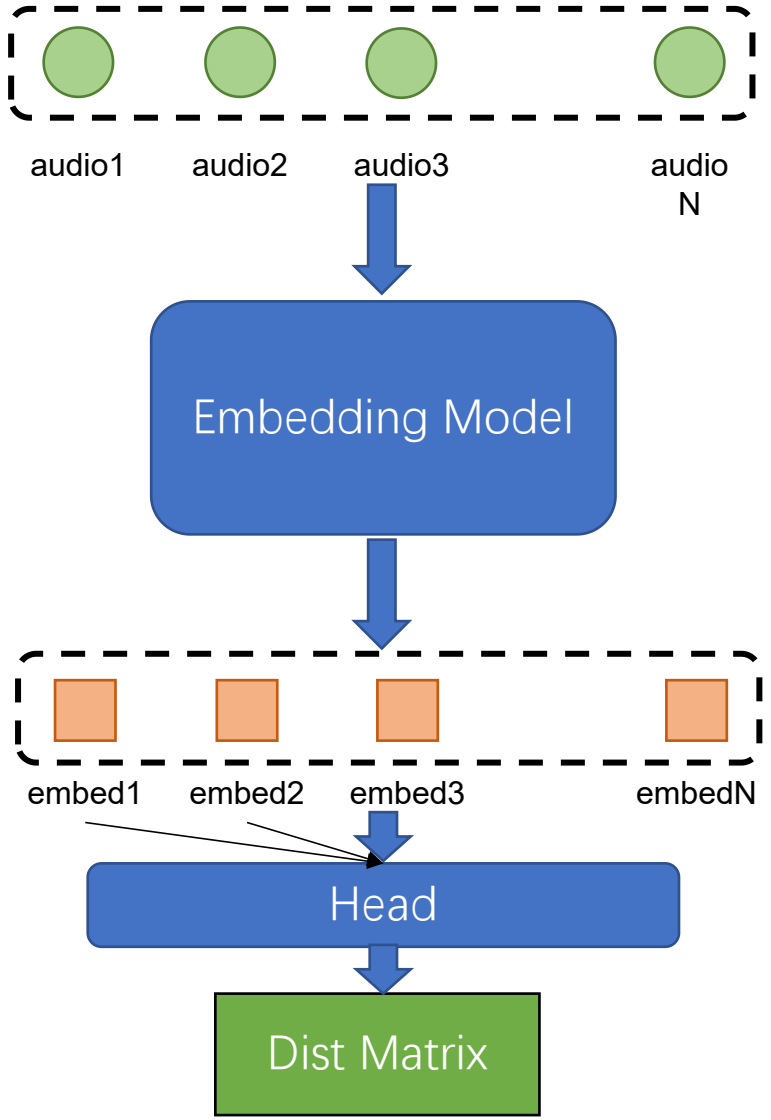
赛题难点

- 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

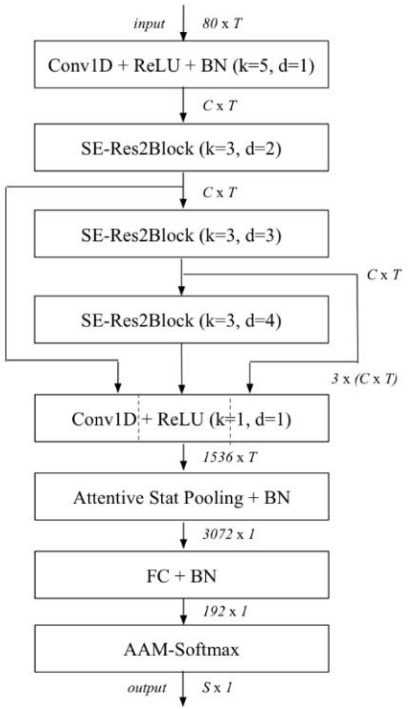


考虑到embed的维度过高，直接用欧式距离作为语音之间的距离，缺乏一定泛化能力，因此考虑设计对应Head网络预测语音之间的距离

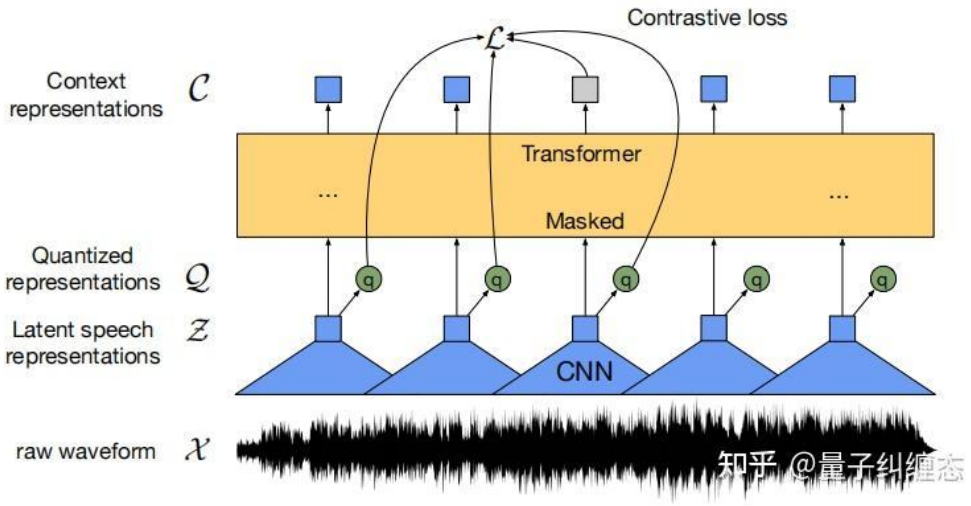


算法设计——模型结构

Embedding Model

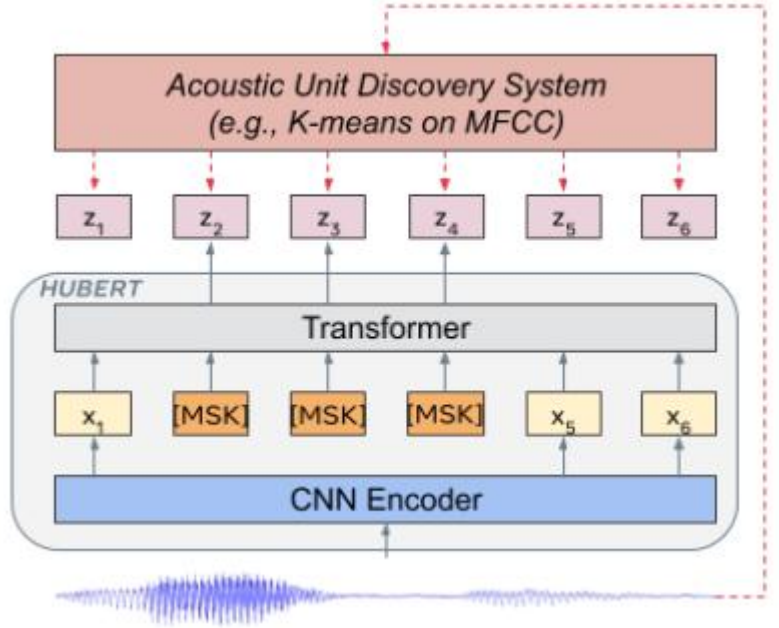


ECAPA-TDNN



Wav2Vec2.0

TencentGameMate/chinese-hubert-large



Hubert

算法设计——模型结构

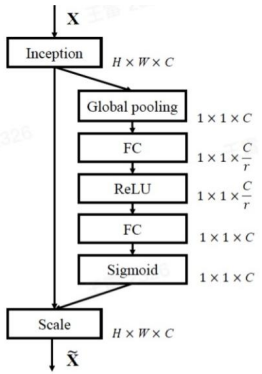
加入Mel Data

audio1 audio2 audio3 audio

MelSpectrogram

Hubert Model

MLPMixer

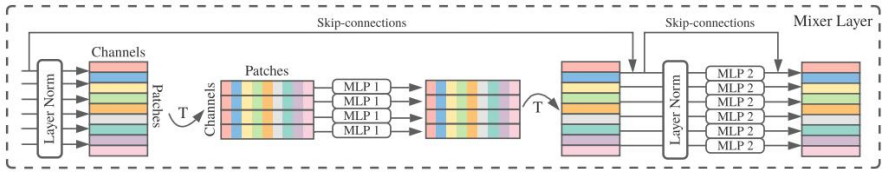


embed1 embed2 embed3 embedN

- SENet
- Linear
- Sigmoid

Head

Dist Matrix



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=random.random()，当p＞0.5时，进行mixup操作：

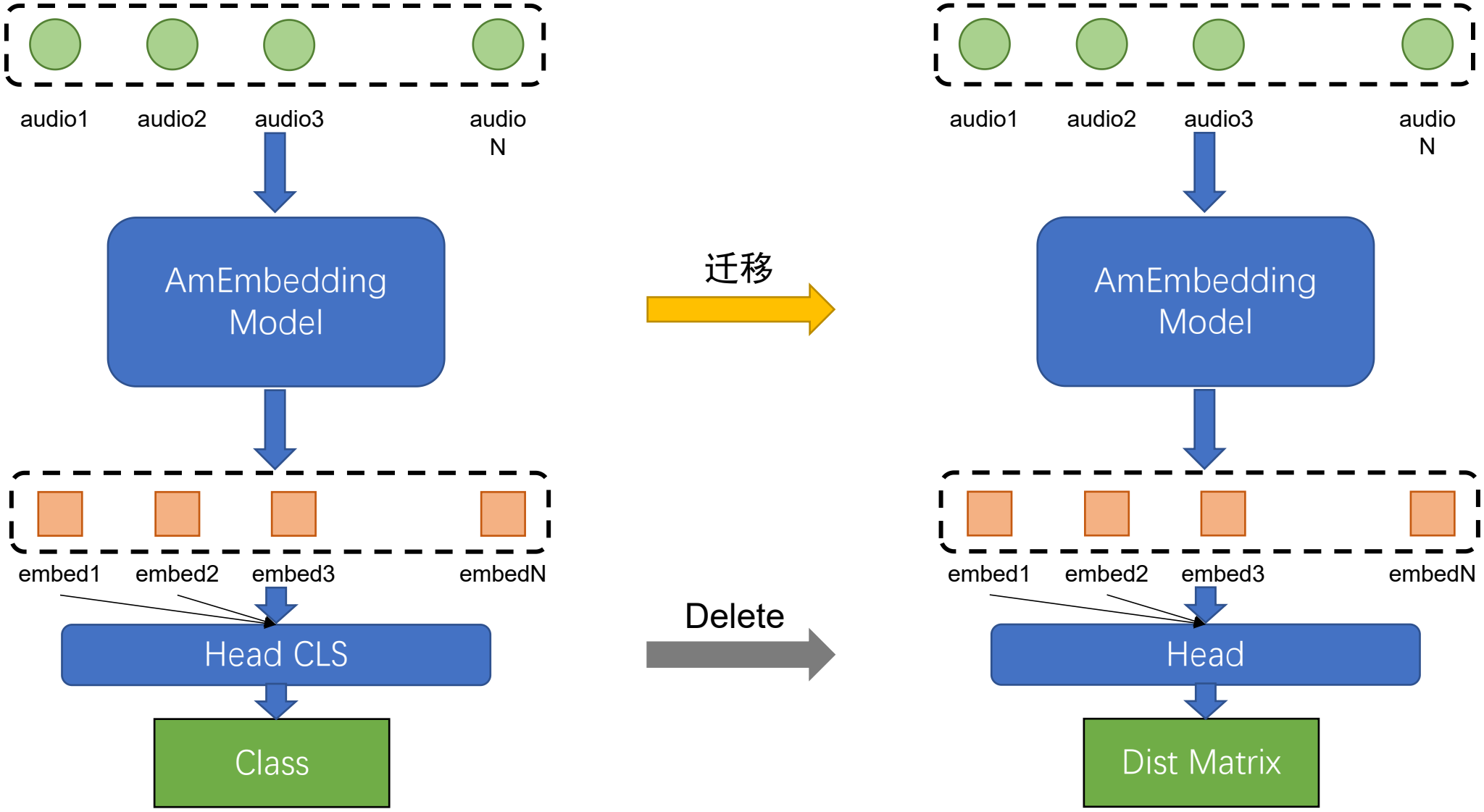
$$audio_{mixup} = \alpha * audio_i + (1 - \alpha) * audio_j$$

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

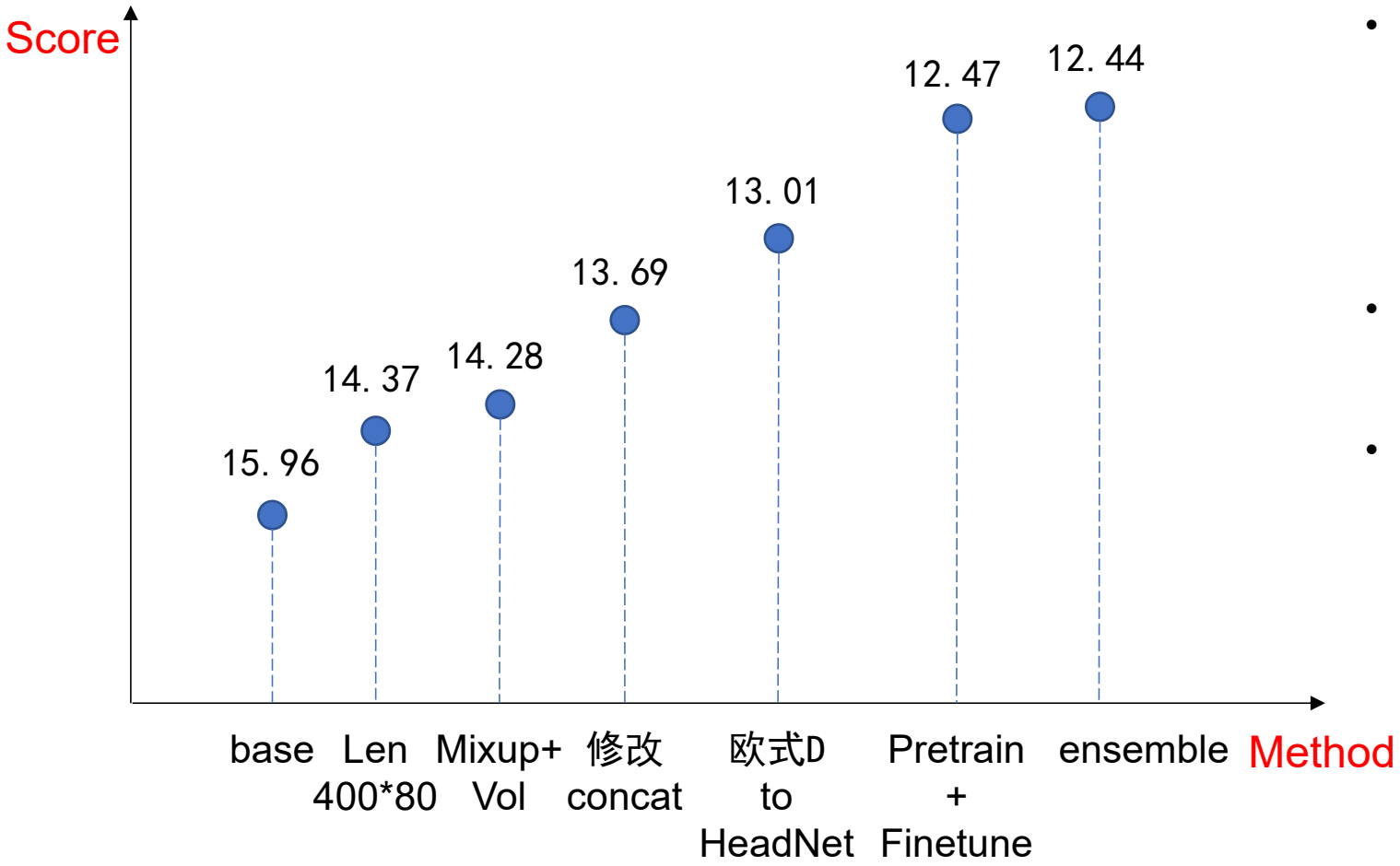
Pretrain与Finetune



EMA

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

信也科技杯 FINVOLUTION BIG DATA COMPETITION | 信也 FINVOLUTION

THANK YOU