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

OPTIC-CZUR

吕光涛 李灏为

目录

CONTENTS

1. 团队简介

吕光涛

西安电子科技大学 OPTIC实验室研一

导师：邓成教授

研究方向：Co-Speech Gesture Generation

2022 微信大数据挑战赛 季军

李灏为

大连成者科技　大连理工大学硕士期间加盟

任首席算法专家、AI 部门总监

研究方向：计算机视觉、 视听多模态

1. 团队简介

团队（OPTIC-CZUR）成绩

复赛 rank2　初赛 rank1

复赛 MAE: 10.21　　-0.05 vs rank1
推理时间: 43 – 49 min

初赛　复赛

* 排名以最优成绩排列，实时更新		排名变化	分数：MAE(平均绝对误差)
1	OnTheWay	0	10.163494
2	OPTIC—CZUR	0	10.211669
3	yuanren	0	10.522829

初赛　复赛

* 排名以最优成绩排列，实时更新		排名变化	分数：MAE(平均绝对误差)
1	OPTIC—CZUR	0	5.553537
2	OnTheWay	0	5.956739
3	yuanren	0	5.965868

2. 数据构建

2.1 数据源

- 赛事数据

- 外部辅助数据 （+0.03~0.05 ）

  - MUSAN 噪声数据集 (OpenSLR 17)
  - RIR_NOISES 噪声和房间响应 (OpenSLR 28)
  - KeSpeech https://github.com/KeSpeech/KeSpeech

- 提升情况
  MUSAN+RIR_NOISES: +0.03~0.05
  KeSpeech 单模型时 +0.1~0.15 多模型时未见提升
  （后期发现单模实际提升原因为：预测的距离变大，而简单的clip小的距离为较大值，也能获得提升）

2. 数据构建

2.1 数据处理 （+0.08 ~ 0.15）

- > 10s 的音频， 拆分成多个 10s 的片段
- 初赛全部的训练数据， 利用分类模型过拟合的方式，剔除难例（噪声样本） acc: 0.997+
- 剔除掉音量过小的片段 < -60dbFS
- 平衡采样， 某类别样本数< 0.1 * 样本类别最大数时， 采样到 **min {** 0.1 * 样本类别最大数，自身数量20倍 **}**

2. 数据构建

2.2 数据增强 （+1.0 ~ 1.5）

- 人声增强
  - 若 > 4s， 随机 crop 4s (音量过小重新crop)
  - 若 < 4s，p=0.9 进行repeat，0.1 概率拼接同类别的音频
  - 随机（p =0.5）mix add同类别的音频
  - 进行以下多种增强

```
self.speech_augmentations = AugCompose([
    RandRemoveDC(prob=0.25, ),
    RandLFilt(prob=0.25, sr=sr),
    RandBiquadFilter(prob=0.1, sr=sr),
    RandResample(prob=0.1, sr=sr),
    RandSpeedPerturb(prob=0.1, sr=sr)
])
self.speech_distortions = AugCompose([
    RandClip(prob=0.1, c_range=(0.05, 0.9)) ])
```

2. 数据构建

2.2 数据增强

- 噪声混合

  - 有色噪声生成 NoiseGenerator(prob=1.0, f_decay_range=(-2, 2), sr=sr)
  - 多种噪声不同增益的混合，包括选择性导入MUSAN 中的噪声
  - SNR mix 0~35db
  - 噪声进一步增强

    RandLFilt(prob=0.25, sr=sr)
    RandBiquadFilter(prob=0.1, sr=sr)
    RandResample(prob=0.05, sr=sr)

- 针对混合后的音频

  选择性添加混响（FRA 生成方法或RIR外部数据）

  AddGaussianNoise(min_amplitude=0.001, max_amplitude=0.015, p=0.5),
  TimeStretch(min_rate=0.8, max_rate=1.25, p=0.5),
  PitchShift(min_semitones=-4, max_semitones=4, p=0.5)
  Normalize By Mean Std

2. 数据构建

2.2 数据增强

- 类别扩展 （类似mixup）

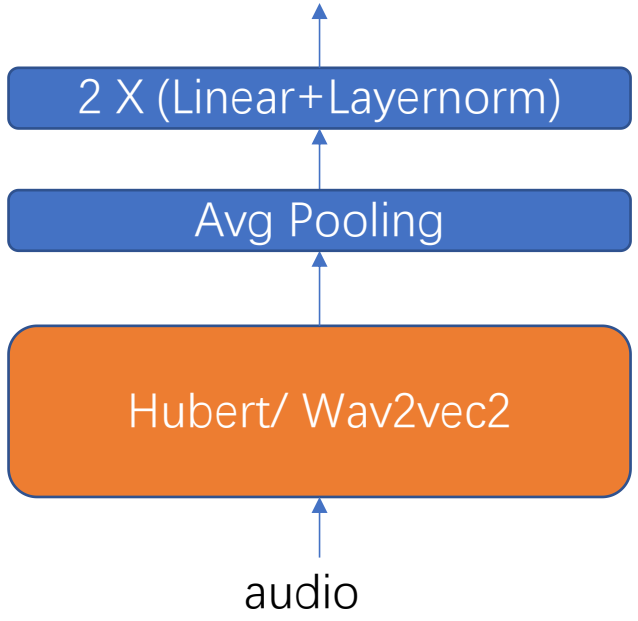
  类别不同的两段音频（i，j）进行叠加或者拼接生成一个新的类别音频，如原始9类，可生成45类

  距离标签“线性分解”，(i1，j1) 与 （i2，j2）直接的距离计算

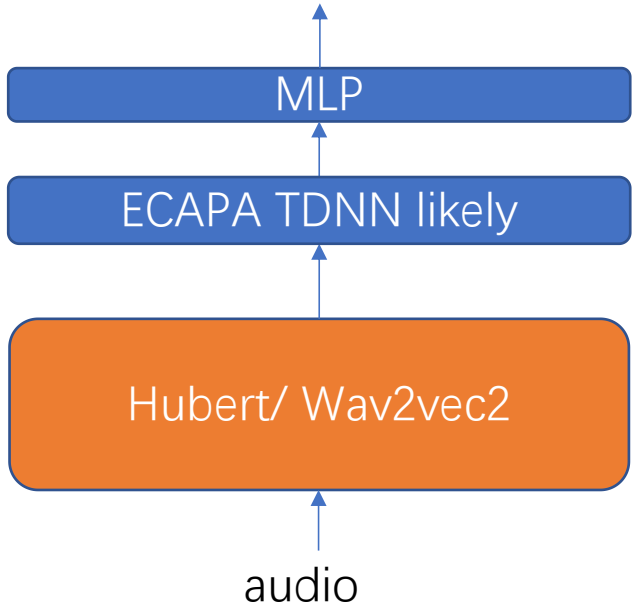
  d1 = DISTANCE_MATRIX_STAGE2[i1, i2]
  d2 = DISTANCE_MATRIX_STAGE2[i1, j2]
  d3 = DISTANCE_MATRIX_STAGE2[j1, i2]
  d4 = DISTANCE_MATRIX_STAGE2[j1, j2]
  d = (d1 + d2 + d3 + d4) / 4

3. 模型与训练

3.1 模型架构



Model Arc1

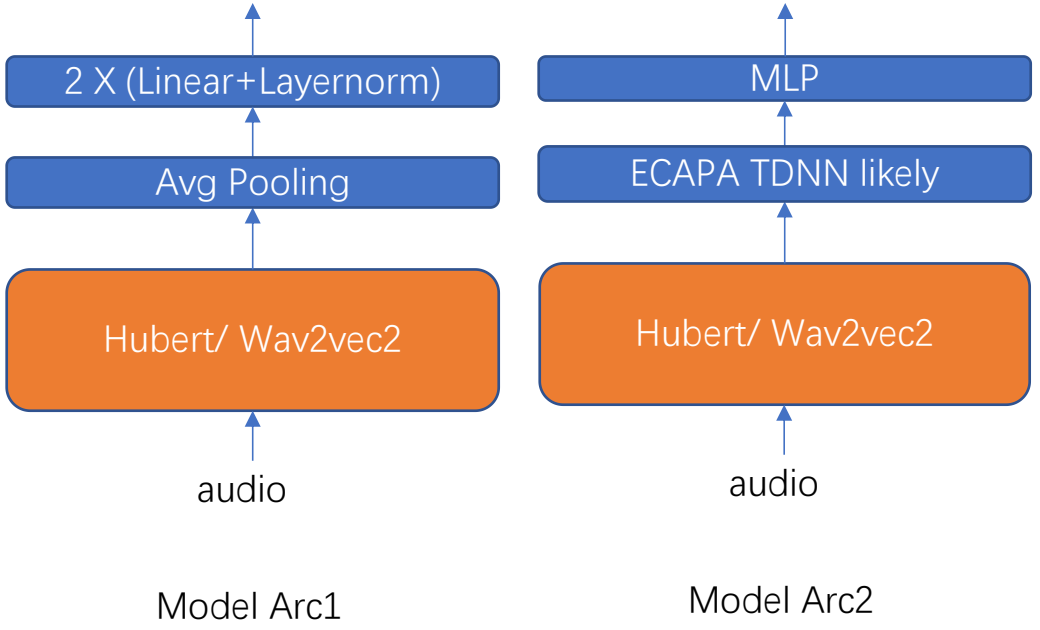


Model Arc2

3. 模型与训练

3.1 模型架构

- Model Arc1 取全局平均特征，收敛慢
- Model Arc2 取多尺度的Attention融合特征，收敛快
- Model Arc2 效果提升 +0.03 ~ 0.05
- 两种模型集成



3. 模型与训练

3.2 预训练模型

1. MMS-facebook-300M https://huggingface.co/facebook/mms-300m
2. chinese-hubert-base https://huggingface.co/TencentGameMate/chinese-hubert-base
3. chinese-hubert-large https://huggingface.co/TencentGameMate/chinese-hubert-large
4. chinese-wav2vec2-large https://huggingface.co/TencentGameMate/chinese-wav2vec2-large

模型表现 （初赛测试集测定）

- MMS-facebook-300M Best Single: 6.32
- chinese-hubert-large Best Single: 5.63
- chinese-wav2vec2-large Best Single: 5.9
- **4 ensemble (our stage1 final)** **5.55**

3. 模型与训练

3.3 损失函数

基于baseline 的 PairDistanceLoss 进行以下改善

- + 计算加速
- + 类别间最大距离不超过 82， 最小距离不小于8，超出区间计算L1 损失 （rule1）
- + 监督同类别样本距离为0 （rule2）

Loss 的提升情况

以 0至13为类内训练数据， 14至17 为类外训练数据（即这部地分距离矩阵设置缺失，但loss 有 rule1 和 rule2）， 评测 14至17 的验证数据。

原始PairDistanceLoss: 13.5 + 改善PairDistanceLoss: 10+

（更适用于知道少量方言间的距离矩阵， 同时还有大量已知类别但未知距离的方言数据，双向验证模型以及距离矩阵标定的合理性）

3. 模型与训练

3.4 模型训练

- 时长逐步增长训练，加快收敛 1s 、 2s、 4s
- 选择性在赛事数据上进行model pretrain
- 1s 2s 4s train epoch [80, 80, 20]
- early stop N: [10, 10, 5]
- weight decay random select from: [2e-5, 3e-5, 4e-5, 5e-5]
- learning rate: [0.00002, 0.00001, 0.00002]
- batch size: [80, 64, 32] or [64, 32, 24]
- AdamW with warmup cosine lr scheduler
- Model Arc2 backbone fix weight load from Model Arc1, and larger batch size
- 6 fold * (Hubert + Wav2vec2)
- amp fp16 O1 train
- EMA and only save fp16 model

4.推理与集成

- 集内样本的embedding 采用该样本所属类别的中心嵌入（训练数据上求嵌入均值）预测时这部分样本不再重复推理
- 6 fold * (Hubert + Wav2vec2) 多组模型集成，求平均
- 剔除掉离集成平均结果最大的模型结果，将剩余的重新求平均
- 预测量化，如果距离 <5，直接认为是同一类别，量化距离为0, 如果大于距离矩阵中的最大值，则置为最大值

4.推理与集成

More Tricks （+1.0+）

Find1 初赛集内-集内 以及 类内-类内 的效果很好
Find2 预测集外且类外的样本时， 模型过度自信，将其“划分”到见过的类别，导致大部分的预测距离偏小
Find3 在 Find2 的基础上， 手动将预测比较小的距离clip 成更大的值，本地验证有提升

→ 1. 对应find1 计算预测的嵌入与18类嵌入中心的最近距离， 如果组内几个模型，有80%模型都是对应到同一个类别， 且平均距离 < 5 则将预测的嵌入直接置为对应的中心嵌入
→ 2. 对应find2 模型预测结果求max 而不是mean， 带来了明显的提升
→ 3. 在 2 的基础上， 进行online 量化 (先验修正)， 如下：

4.推理与集成

→ 3. 在 2 的基础上， 进行online 量化 (先验修正)， 如下：

预测多个模型的结果，求max， 当作fake gt
定义如下量化函数：

```
def regularize_dis_v1(dist, t):
    if dist < 5:
        dist = 0
    elif dist < t:
        dist = t
    elif dist > DIS_MAX_STAGE2:
        dist = DIS_MAX_STAGE2
    return dist
```

→

```
def regularize_dis_v2(dist, t0,   t):
    if dist < 5:
        dist = 0
    elif dist < t0:
        dist = t0
    elif dist < t:
        dist = t
    elif dist > DIS_MAX_STAGE2:
        dist = DIS_MAX_STAGE2
    return dist
```

→

例如以步长1, 最大跨越10搜索regularize_dis_v1函数的参数 t， 使得多个单个模型的预测结果尽可能接近 fake gt, 记为t0， 利用t0 和该量化函数可以得到新的 fake gt
然后再继续搜索regularize_dis_v2 的参数 t， 以此类推，可搜索多次， 直至err 不再降低。

求max 带来了提升，引导量化参数的搜索，但预测距离本就很大的样本对取max 不合理
-> 搜索多次后得到 ti, 对预测距离 > ti 的结果采用模型的平均， 而不是max

5. 总结展望

- 首先尤其感谢信也小伙伴在比赛过程的鼎力支持

- 数据部分：数据部分处理相对比较粗暴，直接截短，某些体现口音的词或许会被切断，换为动态长度更合理。 外部数据提升情况不佳，不符合直观感觉，赛后进一步探索。

- 模型部分：时间维度上多尺度信息的利用不够， 构建多尺度甚至多基础输入特征的预训练模型可能带来提升，Arxiv 最近已有多尺度Hubert 相关文章。

- 距离矩阵：模型的泛化整体一般， 探索已知少量方言的距离矩阵， 同时已知非常大量方言所属的类别（如语保平台有1000多种方言的数据）， 来探索少量标定距离矩阵下，方言的距离度量， 或许是有趣的。

- 训练方式：全监督方法的集外泛化能力非常一般，半监督的聚类方法的引入待进一步实验探索。

信也科技杯 FINVOLUTION BIG DATA COMPETITION | 信也 FINVOLUTION

THANK YOU