专有云定制声学模型训练用户手册

1. 综述

声学模型定制是指，利用用户真实场景下的得到的训练数据，以基础的声学模型为起点进行训练，从而使训练得到的声学模型在用户场景下比基础模型所有提升。

声学模型定制的流程如下：

1. 数据标注(本文档假设标注部分已经完成)
2. 数据格式标准化(如果标注格式不标准，需要转换成标准格式，需要用户自行写代码做转换)
3. 数据解析(统一数据格式并生成训练和测试数据集)
4. 数据处理(生成特征和标签，以及筛选数据)
5. 声学模型训练
6. 声学模型测试

2. 软硬件系统要求

CPU： 物理核>=16 (建议Intel(R) Xeon(R) CPU E5)
内存：>= 128G
硬盘: >=1T
GPU(可选)：NVidia TESLA 较新系列GPU(推荐P100), 卡数>=2；驱动必须安装NVIDIA-Linux-x86_64-384.81(训练包中提供)
Centos 7.4
Python 2.7
有系统sudo权限
训练代码授权许可

3. 训练包结构

```
dcs_model_train_package/
├── am/                                                 -- 声学模型训练相关
|   └── code/asrp_dcs_am.tar.gz                         -- 训练代码压缩包
|   └── gpu_driver/NVIDIA-Linux-x86_64-384.81.run       -- GPU卡驱动
|   └── data_label_format_sample/                       -- 标注格式参考
└── test/                                               -- 测试相关
|   └── code/asr-tools-new.tar.gz                       -- 测试工具
└── doc/                                                -- 文档目录
```

4. 代码授权许可

要使用专有云声学模型训练的代码需要进行使用鉴权，根据项目实际情况，我们会提供硬件鉴权方式（特殊U盘插在服务器上）或者服务器鉴权方式（专有云里面搭建鉴权服务器）。具体事宜请联系ASR服务部署人员。

5. 依赖安装

5.1 GPU驱动安装

如果使用GPU进行训练，需要GPU驱动。首先用nvidia-smi命令检查驱动是否存在，并且版本是否是384.81。如果不符合，按照如下步骤进行GPU驱动安装（注意，需要使用root账号或者有root的权限的账号sudo运行下面的命令）：

- 用以下命令查看系统的kernel版本，确认为3.10.0-693.*.*.el7.x86_64

```
uname -r
```

- 用以下命令查看系统的kernel devel版本

```
rpm -q kernel-devel
```

- 如果不存在或者版本低于3.10.0-693，用dcs_model_train_package/am/gpu_driver/kernel-devel-3.10.0-693.11.1.el7.x86_64.rpm 进行升级

```
rpm -ivh kernel-devel-3.10.0-693.11.1.el7.x86_64.rpm
```

- 重启系统(reboot),并再次确认kernel和kernel devel版本为3.10.0-693
- 屏蔽Nouveau driver，把下面的代码加入 /etc/default/grub 中的GRUB_CMDLINE_LINUX项里面

```
rd.driver.blacklist=nouveau nouveau.modeset=0
```

- 生成新的grub 配置

```
grub2-mkconfig -o /boot/grub2/grub.cfg
```

- 把下面的代码加入 /etc/modprobe.d/blacklist.conf 中（若不存在请新建）

```
blacklist nouveau
```

- 分别运行如下两行代码

```
mv /boot/initramfs-$(uname -r).img /boot/initramfs-$(uname -r)-nouveau.img
dracut /boot/initramfs-$(uname -r).img $(uname -r)
```

- 重启系统(reboot)
- 运行驱动安装程序dcs_model_train_package/am/gpu_driver/NVIDIA-Linux-x86_64-384.81.run (中间问询的问题都选择yes即可, 中间出现warning一般可以不予理会)

```
sh NVIDIA-Linux-x86_64-384.81.run --kernel-source-path="/usr/src/kernels/3.10.0-693.11.1.el7.x86_64"
```

- 用nvidia-smi命令验证驱动是否已经安装成功
- 参考文献： https://www.dedoimedo.com/computers/centos-7-nvidia.html <https://www.dedoimedo.com/computers/centos-7-nvidia.html> 和 http://www.advancedclustering.com/act_kb/installing-nvidia-drivers-rhel-centos-7/ <http://www.advancedclustering.com/act_kb/installing-nvidia-drivers-rhel-centos-7/>
- 训练包已经集成了其他相关依赖（包括cuda），不需要单独安装cuda。

5.2 其他注意

1. 确保以下命令在系统中可以运行： lspci， lscpu; 如果没有，请安装。
2. 训练请不要使用root账户进行

6. 训练流程简介

6.1 数据格式标准化：

目前支持的标注规范格式为长音频标准格式(STANDARD_LONG),短音频标准格式(STANDARD_SHORT)和纯文本格式(PLAIN)。标注格式的示例可以参考： dcs_model_train_package/am/data_label_format_sample 。如果原始标注数据的格式并不是上面提到的标准格式，则需要将训练数据转换成标准格式。因为原始标注数据没有统一格式，这里需要自行写代码做转换。 具体参见《声学模型训练标注数据格式规范及数据格式转换原则》。

6.2 数据解析：

数据解析的目的是把wav文件和标注文本进行格式规整后，生成训练和测试可直接调用的数据集。

6.3 数据处理

这个过程中，我们对数据进行处理，生成训练需要的特征和标签。同时拿到的训练数据可能存在一些质量问题，比如听不清楚，噪音过大，音量过大，音频太长或太短，标注不准确等问题，这里提供三种依次进行的筛选模块。

1. 基本筛选（BasicFilter）:过滤掉太长或太短的音频，以及出现截幅的比例过大音频；
2. 强对齐结果筛选(ForceAlignFilter）：过滤掉强对齐失败的音频；
3. Lattice oracle wer结果筛选(LatticeWerFilter）：首先对每句话生成lattice，然后在lattice中找到一条最接近label的一条路径，并且用这条路径计算lattice oracle WER。过滤掉lattice oracle WER大于特定阈值的音频。

6.4 模型训练

模型训练包括Cross Entropy准则训练和sMBR准则序列训练两个训练步骤。

7. 如何使用

7.1 训练代码

使用前请确保相关软硬件系统要求已经满足！

1. 拷贝训练代码 dcs_model_train_package/am/code/asrp_dcs_am.tar.gz 到本地机器，并解压。

```
tar -xvf asrp_dcs_am.tar.gz
```

2. 拷贝代码中的 /asrp_dcs_am/exampleinput/am_training_pipeline/local_update/ 到单独的本地目录。下面以该目录中的代码为例，对配置更改后进行各训练步骤。

3. 注意：请不要使用root账号运行训练脚本。

7.2 入口脚本配置

在目录中执行sh run_data_and_training.sh 可运行流程，但是首先需要更改 local_src中的路径到asrp_dcs_src/src所在的本地路径。 数据解析(调用 data_parse.xml),数据处理和筛选(调用data_process_filter.xml)和模型训练(调用dcs_am_pipeline.xml) 三个步骤在该脚本中被运行。注意：建议三个步骤要通过三个boolean（run_dp, run_dpf, run_train）控制来单独运行。

7.3 数据解析配置 data_parse.xml

1. 需要更改的配置：

corpus/type	必须是STANDARD_SHORT，STANDARD_LONG和PLAIN中间的一个
corpus/name	解析后的数据子目录名称
corpus/osstranscriptionpath	标注文件的本地目录，目录中可以含有多个标注文件
corpus/osswavpath	wav文件目录
corpus/ossuploadrootpath	数据解析output的root地址
corpus/samplerate	根据实际情况填写16000或者8000

2. 可选配置：

corpus/process/wordlist	用于分词的词典文件（目前训练中文，留空即可，代码自动查找）
trainpercentage和devtestpercentage	若只生成训练数据，trainpercentage=1，devtestpercentage=0。若只生成测试数据，则反之。

3. 解析后的结果：

- 训练数据保存在 $ossuploadrootpath/train/$name/ (后续使用只需要提供目录中的wavelist.txt和transcriptionlist.txt)
- 测试数据保存在 $ossuploadrootpath/devtest/$name/
- 我们还可以查看解析后实际得到的小时数，例如 $ossuploadrootpath/train/$name/001/wavetimeinfo.txt。如果有001,002,总小时候需要在不同目录里面累加出来。

7.4 数据处理 data_process_filter.xml

1. 需要修改的配置：

resourceRootPath	数据处理需要的resource的根目录(这个是一个训练资源包，和asr服务器上基础模型有对应关系，需联系相关发布人员)
outputDir	筛选后结果的保存目录
EvaluateData	每个该标签中指定一个要处理的数据，比如例子中，分别对训练数据和测试数据进行过滤
EvaluateData/Corpus/corpusname	用于在outputDir生成不同子目录的名称
EvaluateData/Corpus/waveList	数据解析得到的wave list文件: $ossuploadrootpath/train/$name/wavelist.txt

EvaluateData/Corpus/transcriptionList	数据解析得到的transcription list文件:$ossuploadrootpath/train/$name/transcriptionlist.txt
EvaluateData/Corpus/samplerate	采样率

注：
a. 因为解析好的数据会有train和devtest两个目录，所以数据处理和筛选一般同时对这两组数据进行处理，即指定训练数据的EvaluateData xml标签和测试数据的EvaluateData xml标签;

2. 可选配置:

EvaluateData/BasicFilter/MinSentenceLengthInMillsecond	音频长度下限
EvaluateData/BasicFilter/MaxSentenceLengthInMillsecond	音频长度上限 （注意，如果实际语音长度一般较长，需要在这里放宽配置）
EvaluateData/BasicFilter/PeakPointRatio	截幅出现频率的上限
EvaluateData/LatticeWerFilter/werThresholdDrop	lattice oracle wer大于该值的音频被过滤掉

3. 处理后的结果:

- 数据处理耗时较长（可比训练本身的时间长）,因为在这个过程中，我们进行的训练的特征和标签的生成。本步骤结束后，会在运行目录中，生成 feature_align_label_train 和 feature_align_lattice_label_train 两个目录供后续训练使用。
- 筛选结果保存在 $outputDir/$corpusname/final_list/中的wavelist_{new,drop}.txt 和 transcriptionlist_{new,drop}.txt, 其中new 和drop 分别对应于过滤之后的wav，被过滤掉的质量较差的wav。一般训练使用过滤后的wav，测试则需要测a.整个测试集，b.过滤后的，c.过滤掉的wav。后两种情况可以作为结果分析参考。
- 我们还可以看到筛选后和筛选掉的wave数和小时数：$outputDir/$corpusname/final_list/wavelist_new_statistics.txt 和 $outputDir/$corpusname/final_list/wavelist_drop_statistics.txt。
- 每个阶段的筛选结果在$outputDir/$corpusname/xxxFilter/

7.5 模型训练配置 dcs_am_pipeline.xml

1. 需要修改的配置：

outputDir	模型输出和log输出的本地目录
resourceRootPath	训练需要的resource的根目录(这个是一个训练资源包，和asr服务器上基础模型有对应关系，需联系相关发布人员)
licenseType	授权方式，服务器鉴权填写software, 硬件授权填写hardware
licenseServerList	如果是服务器鉴权，需要填写鉴权服务器IP地址
CorpusList/Corpus/CeTrainFeatureLabelPath	数据处理之后在本地生成的feature_align_label_train全路径
CorpusList/Corpus/SmbrTrainFeatureLabelPath	数据处理之后在本地生成的feature_align_lattice_label_train全路径
CorpusList/Corpus/copyNum	当指定CeTrainFeatureLabelPath和SmbrTrainFeatureLabelPath，同时需要指定这个参数，一般写1。当大于1的时候，该Corpus会被复制copyNum-1次用于训练

xxTrain/initModel	用于训练的起始模型的完整路径（一般从asr服务器模型目录下面am/am.net 基础模型下载到本地）
xxTrain/featureTrans	和上面的模型对应的特征变换文件的完整路径(从asr服务器模型目录下am/am.mvn下载到本地)

注：CorpusList里面可以指定多个Corpus，把数据合并起来训练,并且可以分别配置copyNum。即多个<Corpus CeTrainFeatureLabelPath=..., SmbrTrainFeatureLabelPath... copyNum=x /> 。

2. 可选配置：

CorpusList/Corpus/waveList	wave文件列表文件地址，由上一步数据处理自动生成：$outputDir/train/final_list/wavelist_new.txt, 其中$outputDir是数据处理输出目录。但如果指定了CeTrainFeatureLabelPath 和 SmbrTrainFeatureLabelPath 则不需要指定wavelist，因为代码会从上述目录中自动获取。如果指定，则会使用指定的wavelist。
CorpusList/Corpus/trascriptionList	transcription列表文件地址，由上一步数据处理自动生成：$outputDir/train/final_list/trascriptionList_new.txt, 其中$outputDir是数据处理输出目录。但如果指定了CeTrainFeatureLabelPath 和 SmbrTrainFeatureLabelPath 则不需要指定trascriptionList，因为代码会从上述目录中自动获取。如果指定，则会使用指定的wavelist。
xxTrain/epochNum	进行多少个epoch(跑多少次完整训练集)
xxTrain/maxValHour	最多用于validation的小时数。系统自动取10%的数据作为validation 数据，但是不能超过maxValHour
xxTrain/cpuProcessNum	根据实际情况调整，但不能多于本机cpu core数。如果设置超过，系统会进行自动调整
xxTrain/gpuProcessNum	根据实际情况调整，但不能多于本机gpu卡数。如果设置超过，系统会进行自动调整

3. 可选高级配置：

Corpus/segDictFile	分词词典本地目录
(xxFe/xxTrain)/ResourcePath	四个子步骤的资源文件目录。除了CeFe内的，其余3个都为lfr模型资源目录
xxTrain/useGpu	如果是true，系统自动检测如果有gpu，则用gpu;如果是false，强制不用gpu。

4. 训练后的结果:

- 训练最后生成的模型放在$outputDir/models/（nnet_smbr是最终输出，其他文件为中间模型。可以考虑都进行测试选最优）
- 训练的log也在$outputDir下面的各个子目录中。一般训练失败，如果屏幕输出看不到具体原因，可以去对应的$outputDir/xx_nnet/log/log文件看具体训练算法输出。
- 原始的训练结果（包括CE和smbr所有中间模型）在当前目录下的exp中。

8 训练建议

1. 强烈建议在训练之前，利用/asrp_dcs_am/exampleinput/am_training_resource_checker/中的脚本示例，对提供的训练资源resource包和被训练的基础模型的兼容性进行检查。如果没有问题，会在屏幕输出的最后显示AM Resource checking is finished。
2. 强烈建议在正式训练之前，先用小数据量走一遍流程（请不要用包里面提供的标注格式用例）；但是小数据也需要有500句以上，同时训练数据需要大于0.8小时。
3. 训练时间很长，请使用screen等工具让任务在背景运行；
4. 先只修改XML中的需要修改的配置，其他可选配置不要动；
5. CE和SMBR训练可以单独进行，只需要去掉对应的XXFe和XXTrain标签。例如dcs_am_pipeline_skip_ce.xml（如下图）配置了只进行smbr训练。initModel可以是之前训练出来的ce模型或者是base模型。

9 常见错误

1. 训练失败，$outputDir/xx_nnet/log/log里面报no CUDA-capable device is detected
一般情况是GPU驱动没有按照规定安装384.8这个版本。

2. 训练失败，$outputDir/xx_nnet/log/log里面报mpiexec has detected an attempt to run as root
训练不能使用root账户。如果已经报这个错误，需要改用非root权限，并且确保训练相关的文件（数据，训练工具，训练启动脚本等等）的owner转换成非root。

3. 数据处理的时候报错，The number of newpost.*.scp is not equal to 20
一般情况下，这个是因为数据太少导致。

4. 训练失败，$outputDir/xx_nnet/log/log里面报Dimensionality mismatch, class_frame_counts xxxx pdf_output_llk xxx
训练起始模型（即被优化的基础模型）和训练资源不匹配导致，请联系提供训练基础模型的同学。

5. 训练失败，$outputDir/xx_nnet/log/log里面报apes授权失败
训练代码授权不成功，联系部署apes授权的同学。