

Kaldi

1.整体：注意三个脚本:cmd.sh path.sh run.sh,其中 cmd.sh 做如下修改，修改为本地运行

```
export train_cmd=run.pl
export decode_cmd="run.pl --mem 4G"
export mkgraph_cmd="run.pl --mem 8G"
export cuda_cmd="run.pl --gpu 1"
```

其中 cmd.sh 设置环境变量(相当于全局)，分了 a、b、c 三类。run.sh 用于运行的主脚本，会调用很多其他的 shell 文件。path.sh 对程序中需要用到的文件路径进行设置

2.数据准备：local/thchs30_data_prep.sh，utils/prepare_lang.sh

3.文件：word.txt（词序列），phone.txt（音素序列），text（与 word.txt 相同），wav.scp（语音），utt2pk（句子与说话人的映射），spk2utt（说话人与句子的映射）

4. produce MFCC features 是提取 MFCC 特征，分为两步，先通过 steps/make_mfcc.sh 提取 MFCC 特征，再通过 steps/compute_cmvn_stats.sh 计算倒谱均值和方差归一化。

5.prepare language stuff 是构建一个包含训练和解码用到的词的词典。而语言模型已经由王东老师处理好了，如果不打算改语言模型，这段代码也不需要修改。

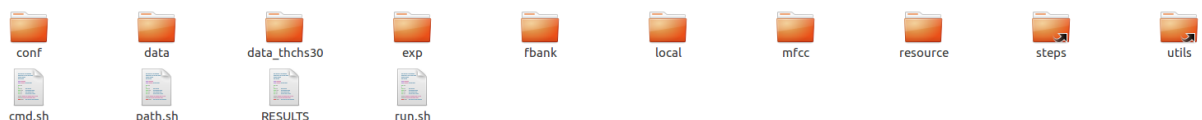
6.过程：数据准备，monophone 单音素训练，tri1 三因素训练，trib2 进行 lda_mllt 特征变换，trib3 进行 sat 自然语言适应，trib4 做 quick，最后 dnn。

7. 用标准的 13 维 MFCC 加上一阶和二阶导数训练单音素 GMM 系统，采用倒谱均值归一化（CMN）来降低通道效应。然后基于具有由 LDA 和 MLLT 变换的特征的单音系统构造三音 GMM 系统，最后的 GMM 系统用于为随后的 DNN 训练生成状态对齐。

8. 基于 GMM 系统提供的对齐来训练 DNN 系统，特征是 40 维 FBank。

9. train_mono.sh 用来训练单音子隐马尔科夫模型，一共进行 40 次迭代，每两次迭代进行一次对齐操作，train_deltas.sh 用来训练与上下文相关的三音子模型，train_lda_mllt.sh 用来进行线性判别分析和最大似然线性转换，train_sat.sh 用来训练发音人自适应，基于特征空间最大似然线性回归，train_quick.sh 用来在现有特征上训练模型，run_dnn.sh 用来训练 DNN。

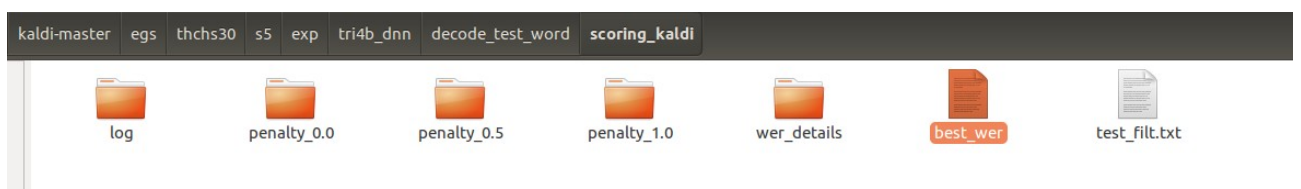
10.



11.



12.



13.%WER 23.72 [19244 / 81139, 464 ins, 602 del, 18178 sub] exp/tri4b_dnn/decode_test_word/wer_7_0.0

总结：kaldi

1.学会清洗数据，统一数据格式

2.字典设计，根据自己语种，设计相应字典

3.conf/下为提取特征参数脚本，需仔细看

4.decode 解码需仔细看，后面需要脚本工程化的化，这是首先要解决的

5.模型的范化能力，一般实验室针对特定数据集即可，若在工程需加噪音，分场景训练.

6.kaldi 工程很庞大，需慢慢学习，可以结合 pytorch 等结合训练.